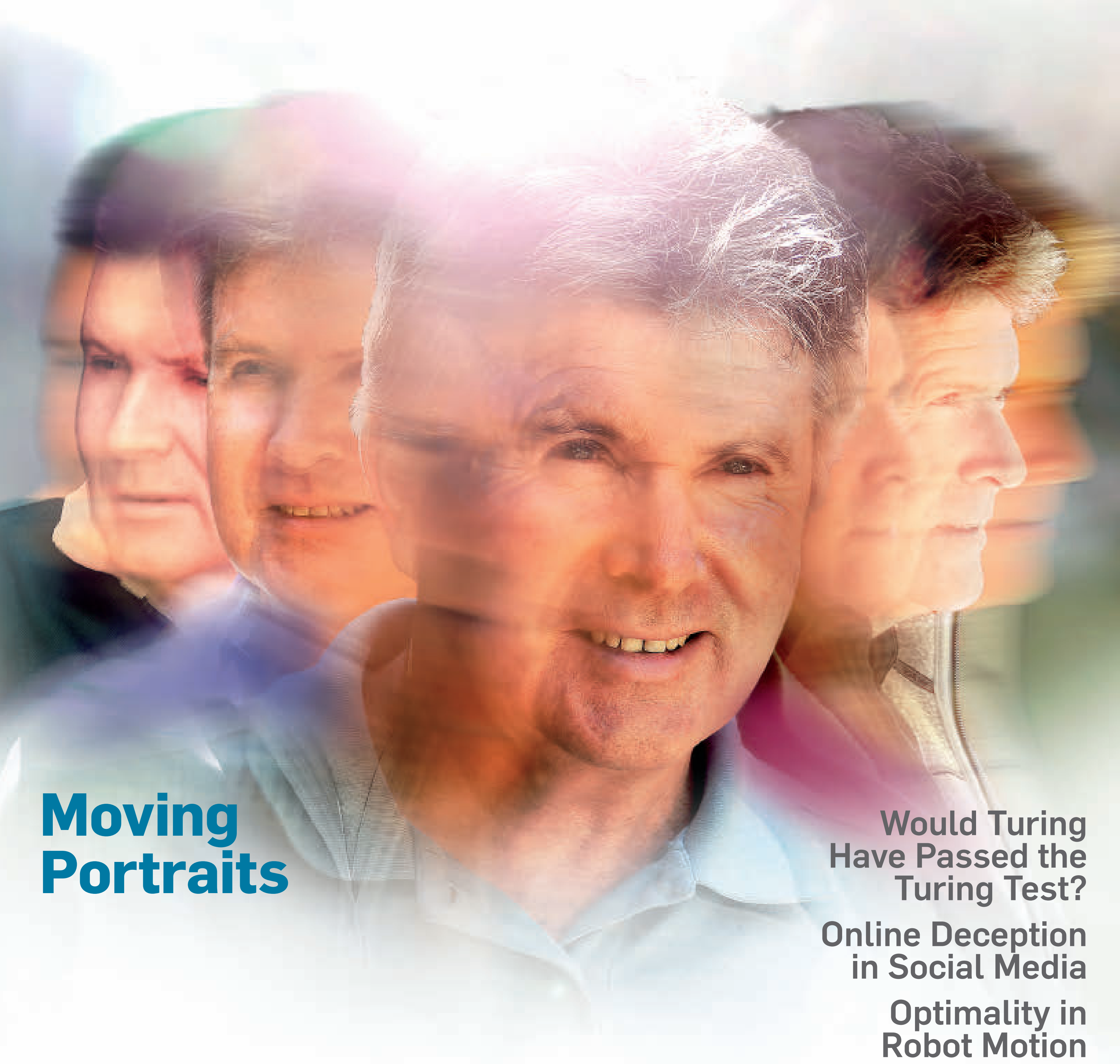
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

09/2014 VOL.57 NO.09



Moving Portraits

Would Turing Have Passed the Turing Test?

Online Deception in Social Media

Optimality in Robot Motion

Securing the Tangled Web

Association for Computing Machinery

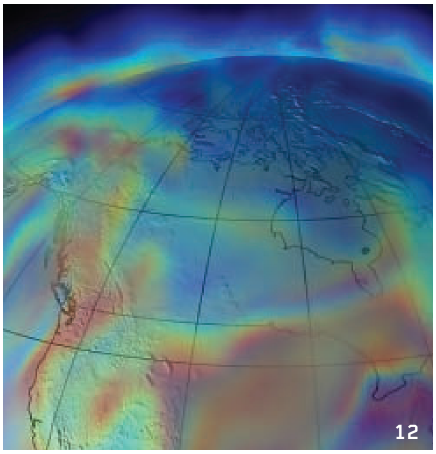
COMMUNICATIONS OF THE ACM

Departments

Last Byte

Association for Computing Machinery
Advancing Computing as a Science & Profession

News



Viewpoints

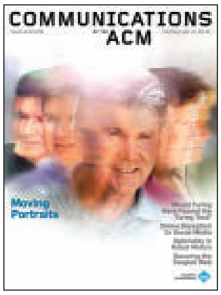
MAP VISUALIZATION BY TRENT SCHINDLER, NASA/GODDARD/UMBC

09/2014
VOL. 57 NO. 09

Practice

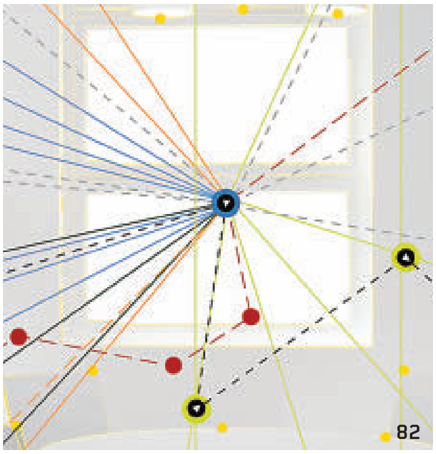
Articles' development led by acmqueue
queue.acm.org

Contributed Articles



About the Cover: The average person is photographed thousands of times during a lifespan. A collection of these images makes up a visual record, or photobio, of an individual. This month's cover story (p. 93) introduces a new approach for presenting such a collection, one that automatically identifies facial features, even as they age, and renders a seamless display of moving portraits from still photos. Cover photo illustration by Barry Downard. Cover model: Barend Booysen.

Review Articles



Research Highlights

IMAGES BY DALIBOR ZIVOTIC; NOMAD SOUL; ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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

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

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

ACM COUNCIL

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

BOARD CHAIRS

Education Board
Andrew McGettrick
Practitioners Board
George Neville-Neil

REGIONAL COUNCIL CHAIRS

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

PUBLICATIONS BOARD

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

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

Computer Science Teachers Association
Lissa Clayborn, Acting Executive Director

STAFF

DIRECTOR OF GROUP PUBLISHING
Scott E. Delman
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
Editorial Assistant
Zarina Strakhan
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
Emily Williams

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/guidelines

ACM ADVERTISING DEPARTMENT

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

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

Media Kit acmmediasales@acm.org

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

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

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/2643596

Moshe Y. Vardi

Would Turing Have Passed the Turing Test?

It's time to consider the Imitation Game as just a game.

ON JUNE 7, 2014, a Turing-Test competition, organized by the University of Reading to mark the 60th anniversary of Alan Turing's death, was won by a Russian chatterbot pretending to be a Russian teenage boy named Eugene Goostman, which was able to convince one-third of the judges that it was human. The media was abuzz, claiming a machine has finally been able to pass the Turing Test.

The test was proposed by Turing in his 1950 paper, "Computing Machinery and Intelligence," in which he considered the question, "Can machines think?" In order to avoid the philosophical conundrum of having to define "think," Turing proposed an "Imitation Game," in which a machine, communicating with a human interrogator via a "teleprinter," attempts to convince the interrogator that it (the machine) is human. Turing predicted that by the year 2000 it would be possible to fool an average interrogator with probability of at least 30%. That prediction led to the claim that Goostman passed the Turing Test.

While one commentator argued that Goostman's win meant that "we need to start grappling with whether machines with artificial intelligence should be considered persons," others argued that Goostman did not really pass the Turing Test, or that another chatterbot—Cleverbot—already passed the Turing Test in 2011.

The real question, however, is whether the Turing Test is at all an important indicator of machine intelligence. The reality is the Imitation Game is philosophically the weakest part of Turing's 1950 paper. The main focus of his paper is whether machines can be intelligent. Turing answered in the affirmative and the bulk of the paper is a philosophical analysis in justification of that answer, an analysis that is as fresh and compelling today as it was in 1950. But the analysis suffers from one major weakness, which is the difficulty of defining intelligence. Turing decided to avoid philosophical controversy and define intelligence operationally—a machine is considered to be intelligent if it can act intelligently. But Turing's choice of a specific intelligence test—the Imitation Game—was arbitrary and lacked justification.

The essence of Turing's approach, which is to treat the claim of intelligence of a given machine as a theory and subject it to "Popperian falsification tests," seems quite sound, but this approach requires a serious discussion of what counts as an intelligence test. In a 2012 *Communications* article, "Moving Beyond the Turing Test," Robert M. French argued the Turing Test is not a good test of machine intelligence. As Gary Marcus pointed out in a *New Yorker* blog, successful chatterbots excel more in quirkiness than in intelligence. It is easy to imagine highly intelligent fictional beings, such as Star Trek's Mr. Spock, badly failing the Turing Test. In fact, it is doubtful whether Turing himself would have passed the test. In a 2003 paper in the *Irish Journal of Psychological Medicine*, Henry O'Connell and Michael Fitzgerald concluded that Turing had Asperger syndrome. While this "diagnosis" should not be taken too seriously, there is no doubt that Turing was highly eccentric, and, quite possibly, might have failed the Turing Test had he taken it—though his own intelligence is beyond doubt.

In my opinion, Turing's original question "Can machines think?" is not really a useful question. As argued by some philosophers, thinking is an essentially human activity, and it does not make sense to attribute it to machines, even if they act intelligently. Turing's question should have been "Can machines act intelligently?," which is really the question his paper answers affirmatively. That would have led Turing to ask what it means to act intelligently.

Just like Popperian falsification tests, one should not expect a single intelligence test, but rather a set of intelligence tests, inquiring into different aspects of intelligence. Some intelligence tests have already been passed by machines, for example, chess playing, autonomous driving, and the like; some, such as face recognition, are about to be passed; and some, such as text understanding, are yet to be passed. Quoting French, "It is time for the Turing Test to take a bow and leave the stage." The way forward lies in identifying aspects of intelligent behavior and finding ways to mechanize them. The hard work of doing so may be less dramatic than the Imitation Game, but not less important. This is exactly what machine-intelligence research is all about!

Follow me on Facebook, Google+, and Twitter.

***Moshe Y. Vardi*, EDITOR-IN-CHIEF**

29th IEEE International Parallel & Distributed Processing Symposium

IPDPS 2015 India

www.ipdps.org

25-29 May 2015 • Hyderabad, India

CALL FOR PARTICIPATION

IPDPS is an annual gathering of computer scientists from around the world and serves as a forum to present and discuss the latest research findings in all aspects of parallel and distributed computing. Visit the IPDPS Website regularly to see information on several ways to participate and to learn details as the program for 2015 develops.

IPDPS Workshops provide attendees an opportunity to explore special topics and are a major part of the IPDPS week-long family of events. Each workshop has its own requirements and schedule for submissions and all are linked from the IPDPS Website. Most have a due date for submissions later than the author notification date for regular conference papers. Visit the IPDPS Website after September for links to the workshops being organized for IPDPS 2015 in Hyderabad, India.

IPDPS PhD Student Program will include a PhD Forum event which offers graduate students an opportunity to present a research poster describing their ongoing dissertation to the entire IPDPS conference audience. The conference will also organize mentoring sessions for students related to research topics, scientific writing and presentation skills.

CALL FOR PAPERS

Authors are invited to submit manuscripts that present original unpublished research in all areas of parallel and distributed processing, including the development of experimental or commercial systems. Work focusing on emerging technologies is especially welcome. See the full Call for Papers published at the IPDPS Website for details.

Topics of interest include, but are not limited to:

- Parallel and distributed algorithms
- Applications of parallel and distributed computing
- Parallel and distributed architectures
- Parallel and distributed software

WHAT/WHERE TO SUBMIT

See the IPDPS Website for details. IPDPS 2015 will require submission of abstracts and registration of papers one week before the paper submission deadline without any late exceptions. All submitted manuscripts will be reviewed. Submitted papers should NOT have appeared in or be under consideration for another conference, workshop or journal.

Abstracts due	October 10, 2014
Submissions due	October 17, 2014
Author notification	December 12, 2014

GENERAL CO-CHAIRS

Susamma Barua, California State University, Fullerton, USA
R. Govindarajan, Indian Institute of Science, Bangalore, India

PROGRAM CHAIR

Srinivas Aluru, Georgia Institute of Technology, USA

PROGRAM VICE-CHAIRS

(Algorithms) Geppino Pucci, University of Padova, Italy
(Applications) Sivan Toledo, Tel-Aviv University, Israel
(Architecture) Mahmut Taylan Kandemir, Pennsylvania State University, USA
(Software) Vivek Sarkar, Rice University, USA

IPDPS 2015 IN HYDERABAD, INDIA

The Hyderabad International Convention Centre will host an event that offers the full IPDPS program of workshops, contributed papers, and keynote speakers as well as representatives from industry and opportunities for students to hear from and interact with senior researchers attending the conference. The Hyderabad airport (RGIA) has direct flights from all international hubs and direct flights to all major cities in India. Hyderabad offers a variety of tourist attractions for attendees and their families and a chance to explore other parts of India. Visit the IPDPS Website regularly to see new program developments and to get tips on travel to the premier conference in parallel and distributed computing.

Sponsored By:
IEEE computer society
Technical Committee on Parallel Processing

In Cooperation With:
ACM SIGARCH
IEEE Computer Society Technical Committee on Computer Architecture
IEEE Computer Society Technical Committee on Distributed Processing

DOI:10.1145/2656433

Vinton G. Cerf

Augmented Reality

My attention was recently drawn to a remarkable seven-minute TED video by Louie Schwartzberg[a] that reinforced for me the power of technology to adapt to the limitations of our human perceptions.

With the aid of technology, often digital in nature and often involving some serious computation, we can perceive that which is too fast, too slow, too big, too small, too diverse, and too high or low (as in frequency). As Schwartzberg's video illustrates, we can use time-lapse photography to watch processes too slow to perceive or high-speed photography to make visible that which is too fast for the human eye to see. We can downshift or upshift frequencies to make things audible that we would otherwise not detect: the low-frequency communication of elephants[b] and the high frequencies generated by bats and pest-control devices. We can shift or detect high-energy and high-frequency photons, such as X-rays, and make them visible to the human eye. We can take images in ultraviolet or infrared that our eyes cannot see but our instruments can, and thus make them visible.

Anyone who has watched a time-lapse film of flowers opening or mushrooms growing or vines climbing can appreciate how dramatically the time-lapse images help us appreciate and understand processes that take place so slowly that we do not see them as dynamic. I recall visiting a rain forest in Irian Jaya (the western half of Papua New Guinea) where our guide explained the long, slow battle between the trees and the climbing vines that, ultimately, throttled the trees over a period of years. I recall when my son, David, suggested a 100-year project to photograph, in time-lapse, a forest's vivid story. It would be quite an interesting experience to watch the slow, titanic battles for control of the upper canopy and the survival and regeneration of the ground-hugging brush over the course of decades. It would be a technical challenge to ensure the equipment stayed functional, but one could use radio transmission to capture the images as long as the cameras were in operating condition. Similar tactics have been used to observe, on a continuous basis, areas not friendly to human habitation such as winters at the poles.

The rise of interest in "big data" has spurred a concurrent interest in visualization of collections of digital information, looking for patterns more easily recognized by humans than by computer algorithms. Creation of overlays of multiple data sources on Google Earth, correlated as to time and geographic location, also have served as an organized way to visualize and experience information we could not naturally observe with our human senses. Similar methods have brought visibility to the distribution of dark matter in the universe by inferring its existence and presence through its gravitational effects.

As our computational tools become more and more powerful, we can anticipate that our growing knowledge of the mechanics of our world will allow us to use simulation to visualize, understand, and even design processes that we could only crudely imagine before. The 2013 Nobel Prize for Chemistry went to Martin Karplus, Michael Levitt, and Arieh Warshel "for the development of multiscale models for complex chemical systems." This is computational chemistry at its best and it shows how far we have come with tools that depend upon significant computing power available in this second decade of the 21st century. Indeed, we hear, more and more, of computational physics, biology, linguistics, exegesis, and comparative literature as fields well outside the traditional numerical analysis and programming disciplines typically associated with computer science. Computation has become an infrastructure for the pursuit of research in a growing number of fields of science and technology, including sociology, economics, and behavioral studies.

One can only speculate what further accumulation of digitized data, computational power, storage, and models will bring in the future. The vast troves of data coming from the Large Hadron Collider, the Hubble, and future James Webb telescopes (among others), and the NSF National Ecological Observation Network (NEON) program[c] will be the sources for visualization, correlation, and analysis in the years ahead. Whoever thinks computer science is boring has not been paying attention!

[a] https://www.youtube.com/watch?v=FiZqn6fV-4Y

[b] https://www.youtube.com/watch?v=YfHO6bM6V8k

[c] http://www.nsf.gov/funding/pgm_summ.jsp and http://www.neoninc.org/

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

DOI:10.1145/2656189

Provenance of British Computing

David Anderson's Viewpoint "Tom Kilburn: A Tale of Five Computers" (May 2014) on the pioneering computers built at the University of Manchester was fascinating and informative, but no article on the history of British computing can avoid the precedence controversy between the Universities of Manchester and Cambridge. For example, Cambridge advocates will put the case for the EDSAC computer and its many achievements, including nearly 10 years of service to the scientific community starting in 1949. But the Manchester Baby was operational more than 10 months earlier. It is in this spirit we should examine Anderson's remark, "Starting in 1963, [Kilburn] spent several years establishing and organizing a new Department of Computer Science, the first of its kind in the U.K." It is no criticism of Manchester's fine School of Computer Science to ask, what is the word "first" doing here? (Let us ignore the qualifier "of its kind," which would guarantee uniqueness of almost anything.) The Cambridge department had already been in existence for 27 years. The central authorities at Cambridge published its *Report on the Establishment of a Computing Laboratory* in 1936, with the aim of providing a computing service to the sciences while also conducting research on computational techniques. Initially called the Mathematical Laboratory, it developed the EDSAC and other computers, taught programming to Edsger W. Dijkstra in 1951, and established the world's first course in computing (at the master's level) in 1953. Another point of note: Many people imagine the development of computing was driven by the demands of war, but the Mathematical Laboratory (now known as the Computer Laboratory) was created from the outset to meet the needs of science.

Lawrence C. Paulson, Cambridge, England

A Programming Language Is Not a User Interface

A programming language is not a user interface but rather an expert-only tool, like, say, a command line but not the language-as-interface concept outlined by Mark Guzdial in his blog "The Difficulty of Teaching Programming Languages, and the Benefits of Hands-on Learning" (July 2014) in response to Andy Ko's earlier blog. Viewing a language as a user interface reflects a flawed understanding of the language. How to learn a programming language depends on whether the programmer has an accurate model of the language, the underlying machine, and what the programmer is trying to accomplish.

Consider a musical instrument as a physical analogy to a programming language, where the instrument is the "interface" to the realm of creating music. Mastering the instrument is one thing; understanding music is something else. Without understanding, learning to play may be futile. No musician lacking talent or a deep understanding of music will ever be a truly accomplished player. Changing one type of instrument to one easier to use does not change the connection between understanding and performance.

The instrument's role in the process is minor. Yet with programming languages, some say we simply have not found a language that is easy to teach and inherent difficulty learning to write good code will magically disappear. Most bad software is produced by programmers with limited understanding of what they are trying to accomplish and the tools they are trying to use. Programming languages play only a minor role in such a personal struggle, while choosing a particular language is at best only a question of convenience. Sure, choosing a language with clear representation of specific concepts helps teach and learn the concepts but does not guarantee understanding.

Unless teachers acknowledge the inherent difficulty of programming and its dependence on talent and dedication, there can be no end to the software crisis. Moreover, trying to teach programming to students who lack that talent will continue to produce incompetent programmers.

Reflecting on my own experience in commercial projects, I can say that paying more for competent programmers pays off. Some programmers actually represent negative productivity, in that cleaning up after them costs more than any value they might have created. Though many who call themselves programmers may have to quit the profession, the same would happen to talentless musicians pursuing musical performance as a career. The difference is that most people recognize badly played music (it hurts), while, apparently, not all teachers of computer science recognize why so much bad code continues to be produced.

Arno Wagner, Zürich, Switzerland

Release the Source Code

A welcome addition to the 16 items Chuck Huff and Almut Furchert recommended in their Viewpoint "Toward a Pedagogy of Ethical Practice" (July 2014) would be the release of source code. Few practices could do as much to give users confidence that the code they depend on functions as intended, meets requirements, and reflects the choices they approve. Whether an open license is used (permitting code redistribution or alteration) is a separate matter based on the goals and business plan of the coding organization. But allowing outside experts to freely view the code would be a natural step for organizations developing software in the public interest.

Andy Oram, Cambridge, MA

Toward a Clear Sense of Responsibility

Vinton G. Cerf's Cerf's Up column "Responsible Programming" (July 2014) should be echoed wherever software is used, procured, or developed. Dismal software quality hinders the economy, national security, and quality of life. Every organization is likely rife with process error. If you have not been affected by a cyberattack you soon could be. Software industry analyst Capers Jones (http://www.spr.com) reported deployed software systems, circa 2012, contained approximately 0.4 latent faults per function point. Reflecting on the urgency of moving to responsible programming, this statistic improved approximately 300% since the 1970s; compare this statistic to automobile engineers achieving 3,000% reduction in emissions in less time.

Almost all operational errors and successful cyberattacks can be traced to faulty code. Responsible programming must therefore extend beyond individual programs to the whole set of programs that interoperate to accomplish a user's purpose, even in the context of nondeterministic situations. Responsible programming could thus ensure each program supports system principles concerning safety properties.

An early example involved Antonio Pizzarello, who co-founded a company in 1995 to commercialize a fault-detection-and-correction theory developed by Edsger W. Dijkstra et al. at the University of Texas. As described in Pizzarello et al.'s U.S. Patent No. 6029002 *Method and Apparatus for Analyzing Computer Code Using Weakest Precondition* the code-analysis method starts with a user-identified, unacceptable post-condition. An analyst writes the desired result, then calculates the weakest precondition until reaching a contradiction highlighting the statement containing the logic, arithmetic, or semantic fault. However, though Pizzarello's method was a technical success, it could not scale economically to larger systems of programs containing numerous possible paths because it was prohibitively labor-intensive and slow, even for highly trained analysts.

The promise of hardware for massively parallel, conditional processing prompted a complete reconceptualization in 2008; typical is the Micron Automata Processor (http://www.micron.com/about/innovations/automata-processing). A new software-integrity assessment method thus enables proofreading computer code as text while applying deep reasoning regarding software as predicates for logic, arithmetic, and semantic coherence at a constant, predictable rate of approximately 1Gb/sec. Software faults are detected, systemwide, automatically.

Informal polls of software developers find they spend approximately 50% of their project time and budget defining, negotiating, and reworking program interfaces and interoperation agreements. They then waste approximately 40% of test time and budget awaiting diagnosis of and fixes for test aborts. Software-integrity assessment can preclude wasted time and money. Moreover, software maintainers and developers may be able to find and null faults more quickly than cyberattackers are able to create them.

Jack Ring, Gilbert, AZ

A Plea for Consistency

Although Dinei Florêncio et al. made several rather grand claims in their Viewpoint "FUD: A Plea for Intolerance" (June 2014), including "The scale of the FUD problem is enormous," "While security is awash in scare stories and exaggerations," and "Why is there so much FUD?," they offered no evidence to support them. Odd, given that they also said, "We do not accept sloppy papers, so citing dubious claims (which are simply pointers to sloppy work) should not be acceptable either."

Alexander Simonelis,
Montréal, Canada

Authors' Response:

We offered many examples but could not include references for everything. Typing "digital Pearl Harbor," "trillion-dollar cybercrime," or other terms into a search engine will easily produce examples of who has been saying and repeating what.

Dinei Florêncio, **Cormac Herley**,
and **Adam Shostack**

Correction

An editing error in "From the President" (June 2014) resulted in an incorrect awards citation. Susan H. Rodger received the Karl V. Karlstrom Outstanding Educator Award for contributions to the teaching of computer science theory in higher education and the development of computer science education in primary and secondary schools.

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.

Coming Next Month in COMMUNICATIONS

Reading with Maps: The Power of Searching with Spatial Synonyms

Abstractions for Software-Defined Networks

Dissection: A New Paradigm for Solving Bicomposite Search Problems

Disrupting and Transforming the University

A Turing Tale

Wikidata: A Free Collaborative Knowledge Base

Plus the latest news about the evolution of programming paradigms, digital forensics, and optical computing.

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

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

DOI:10.1145/2641221 http://cacm.acm.org/blogs/blog-cacm

Refining Students' Coding and Reviewing Skills

Philip Guo sees code reviews providing students "lots of pragmatic learning."

Philip Guo
Small-Group Code Reviews for Education
http://bit.ly/1kf07PP
June 19, 2014

Code reviews (http://bit.ly/Ve4Hbw) are essential in professional software development. When I worked at Google, every line of code I wrote had to be reviewed by several experienced colleagues before getting committed into the central code repository. The primary stated benefit of code review in industry is improving software quality (http://bit.ly/1kvYv4q), but an important secondary benefit is education. Code reviews teach programmers how to write elegant and idiomatic code using a particular language, library, or framework within their given organization. They provide experts with a channel to pass their domain-specific knowledge onto novices; knowledge that cannot easily be captured in a textbook or instruction manual.

Since I have now returned to academia, I have been thinking a lot about how to adapt best practices from industry (http://bit.ly/1iPU31a) into my research and teaching. I have been spending the past year as a postdoc (http://bit.ly/1qRnowt) in Rob Miller's (http://bit.ly/1mKQybj) group at the Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory (MIT CSAIL, http://www.csail.mit.edu/) and witnessed him deploying some ideas along these lines. For instance, one of his research group's projects, Caesar (http://bit.ly/1o9AXVg), scales up code review to classes of a few hundred students.

In this article, I describe a lightweight method Miller developed for real-time, small-group code reviews in an educational setting. I cannot claim any credit for this idea; I am just the messenger.

Small-Group Code Reviews for Education

Imagine you are a professor training a group of students to get up to speed on specific programming skills for their research or class projects.

Ideally, you would spend lots of one-on-one time with each student, but obviously that does not scale. You could also pair them up with senior grad students or postdocs as mentors, but that still does not scale well. Also, you now need to keep *both* mentors and mentees motivated enough to schedule regular meetings with one another.

Here is another approach Miller has been trying: use regular group meeting times, where everyone is present anyway, to do real-time, small-group code reviews. Here is the basic protocol for a 30-minute session:

1. Gather together in a small group of four students plus one facilitator (for example, a professor, TA, or senior student mentor). Everyone brings their laptop and sits next to one another.

2. The facilitator starts a blank Google Doc and shares it with everyone. The easiest way is to click the "Share" button, set permissions to "Anyone who has the link can edit," and then send the link to everyone in email or as a ShoutKey shortened URL (http://bit.ly/1mUHq8U). This way, people can edit the document without logging in with a Google account.

3. Each student claims a page in the Google Doc, writes their name on it, and then pastes in one page of code they have recently written. They can pass the code through an HTML colorizer (http://bit.ly/1jKDaE9) to get syntax highlighting. There is no hard and fast rule about what code everyone should paste in, but ideally it should be about a page long and demonstrate some nontrivial functionality. To save time, this

setup can be done offline before the session begins.

4. Start by reviewing the first student's code. The student first gives a super-quick description of their code (<30 seconds) and then everyone reviews it in silence for the next four minutes. To review the code, simply highlight portions of text in the Google Doc and add comments using the "Insert comment" feature. Nobody should be talking during these four minutes.

5. After time is up, spend three minutes going around the group and having each student talk about their most interesting comment. This is a time for semi-structured discussion, organized by the facilitator. The magic happens when students start engaging with one another and having *"Aha! I learned something cool!"* moments, and the facilitator can more or less get out of the way.

6. Now move onto the next student and repeat. With four students each taking ~7.5 minutes, the session can fit within a 30-minute time slot. To maintain fairness, the facilitator should keep track of time and cut people off when time expires. If students want to continue discussing after the session ends, then they can hang around afterward.

That is it! It is such a simple, lightweight activity, but it has worked remarkably well so far for training undergraduate researchers in our group.

Benefits

Lots of pragmatic learning occurs during our real-time code reviews. The screenshot above shows one of my comments about how the student can replace an explicit JavaScript for-loop with a jQuery .each() function call (http://bit.ly/1sYfa9O). This is exactly the sort of knowledge that is best learned at a code review rather than by, say, reading a giant book on jQuery. Also, note that another student commented about improving the name of the "audio" variable. A highly motivating benefit is that students learn a lot even when they are reviewing someone else's code, not just when their own code is being reviewed.

With a 4-to-1 student-to-facilitator ratio, this activity scales well in a research group or small project-based class. This group size is well-suited for serendipitous, semi-impromptu discussions and eliminates the intimidation factor of sitting one-on-one with a professor. At the same time, it is not large enough for students to feel anonymous and tune out like they would in a classroom. Since everyone sees who else is commenting on the code, there is some social pressure to participate, rather than zoning out.

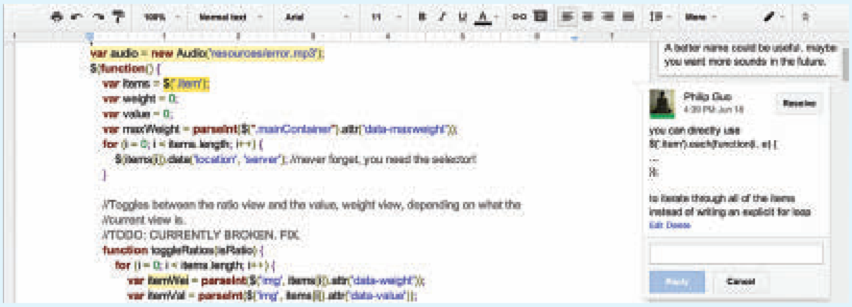
A side benefit of holding regular code reviews is that it forces students to make consistent progress on their projects so they have new code to show during each session. This level of added accountability can keep momentum going strong on research projects since, unlike industry projects, there are no market-driven shipping goals to motivate daily progress. At the same time, it is important not to make this activity seem like a burdensome chore, since that might actually drain student motivation.

Comparing with Industry Code Reviews

Industry code reviews are often asynchronous and offline, but ours happen in a real-time, face-to-face setting. This way, everyone can ask and answer clarifying questions on the spot. Everything happens within a concentrated 30-minute time span, so everyone maintains a shared context without getting distracted by other tasks.

Also, industry code reviews are often summative assessments (http://bit.ly/1z8kOH1) where your code can be accepted only if your peers give it a good enough "grade." This policy makes sense when everyone is contributing to the same production code base and wants to keep quality high. Here, reviews are purely formative (http://bit.ly/1jKElU6) so students understand they will not be "graded" on it. Rather, they understand the activity is being done purely for their own educational benefit.

The author comments on a student's code.



```
var audio = new Audio('resources/error.mp3');
$(function() {
  var items = $('.item');
  var weight = 0;
  var value = 0;
  var maxWeight = parseInt($(".mainContainer").attr('data-maxweight'));
  for (i = 0; i < items.length; i++) {
    $(items[i]).data('location', 'server'); //never forget, you need the selector!
  }

  //Toggles between the ratio view and the value, weight view, depending on what the
  //current view is.
  //TODO: CURRENTLY BROKEN. FIX.
  function toggleRatios(isRatio) {
    for (i = 0; i < items.length; i++) {
      var itemWei = parseInt($('img', items[i]).attr('data-weight'));
      var itemVal = parseInt($('img', items[i]).attr('data-value'));
```

A better name could be useful, maybe you want more sounds in the future.

Philip Guo
4:30 PM Jun 18

you can directly use
$('.item').each(function(i, e) {
...
});

to iterate through all of the items instead of writing an explicit for loop

Parting Thoughts

When running this activity, you might hit the following snags:

‣ *Overly quiet students.* It usually takes students a few sessions before they feel comfortable giving and receiving critiques from their peers. It is the facilitator's job to draw out the quieter students and to show that everyone is there to help, not to put one another down. Also, not all comments need to be critiques; praises and questions are also useful. I have found that even the shyest students eventually open up when they see the value of this activity.

‣ *Running over time.* When a discussion starts to get really interesting, it is tempting to continue beyond the three-minute limit. However, that might make other students feel left out since their code did not receive as much attention from the group, and it risks running the meeting over the allotted time. It is the facilitator's job to keep everyone on schedule.

‣ *Lack of context.* If each student is working on their own project, it might be hard to understand what someone else's code does simply by reading it for a few minutes. Thus, student comments might tend toward the superficial (like *"put an extra space here for clarity"*). In that case, the facilitator should provide some deeper, more substantive comments. Another idea is to tell students to ask a question in their code review if they cannot think of a meaningful critique. That way, at least they get to learn something new as a reviewer.

Try running a real-time code review at your next group or class meeting, and email me at philip@pgbovine.net with questions or suggestions.

Philip Guo is an assistant professor of computer science at the University of Rochester.

Science | DOI:10.1145/2641225

Samuel Greengard

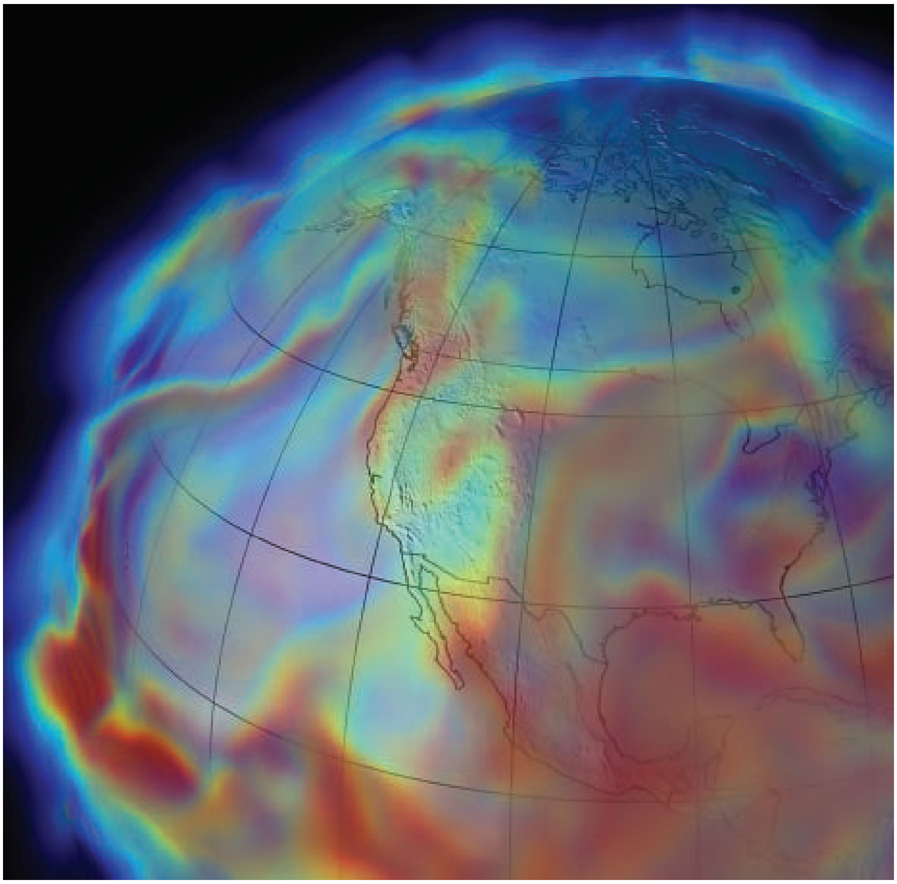
Weathering a New Era of Big Data

Increased computing power combined with new and more advanced models are changing weather forecasting.

THROUGHOUT HISTORY, MANKIND has attempted to gain a better understanding of weather and forecast it more accurately. From ancient observations about wind direction, cloud formations, and barometric pressure to more recent attempts to accumulate data from satellites, sensors, and other sources, weather forecasting has both fascinated and infuriated everyone from picnickers to farmers and emergency responders. It is, in a word, *unpredictable*.

Yet over the last two decades, thanks to increasingly powerful computers, big data, and more sophisticated modeling and simulations, weather forecasting has been steadily moving forward. Amid growing concerns about global warming and more volatile weather and climate patterns, researchers are attempting to develop better algorithms and systems. Says Cliff Mass, professor of atmospheric science at the University of Washington, "Numerical data is the core technology of weather prediction. Everything is dependent upon it."

Moreover, the stakes continue to grow. There is mounting concern that not all weather models are created equal. In some cases, European and



A visualization of data from the NASA Center for Climate Simulation, a state-of-the-art supercomputing facility in Greenbelt, MD, that runs complex models to help scientists better understand global climate. This visualization depicts atmospheric humidity during the Great Mississippi and Missouri Rivers Flood of 1993.

VISUALIZATION BY TRENT SCHINDLER, NASA/GODDARD/UMBC

American forecasting methods lead to significantly different predictions—and noticeably different results. This includes predicting the impact of snowstorms in the northeast U.S. in the winter of 2013–2014 and the effects of Hurricane Sandy on that region in 2012.

Meanwhile, private companies such as IBM are entering the picture and introducing entirely different tools and methods for forecasting weather-related events.

Says Lloyd Treinish, an IBM Distinguished Engineer at the IBM Thomas J. Watson Research Center, "The history of weather forecasting and the history of computing have been very tightly coupled. Since the 1940s, revolutions in computing have been very closely tied to weather forecasting and building better equations and models. Over the last couple of decades, we have seen steady improvements in computing, sensor technology, an understanding of the atmosphere, and the overall science. Over time, we are learning how to put all the pieces to work."

A Clearer View

The basis for understanding weather in a more systematic way dates back more than a century, to a time when scientists were beginning to examine the physics of the atmosphere and were first applying numerical methods to understand extraordinarily complex physical processes. By the 1950s, forecasters had begun using mainframe computers to build weather models, moving beyond field observations and telegraph reports from the field. By the 1960s, satellites and sensors from automatic stations, aircraft, ships, weather balloons, and drifting ocean buoys entered the picture; they created entirely new and powerful ways to collect data, so it is possible to better understand the mechanisms associated with weather and climate.

Along the way, advances in technology and modeling have led to remarkable improvements—although, as Mass notes, "We are constantly limited by computer power." It is a concern echoed by Ben Kyger, director of central operations for the National Centers for Environmental Prediction at the U.S. National Oceanic and Atmospheric Administration (NOAA). "Scientists increase the grid resolution to take advantage of the available processing power. Higher resolutions mean that scientists can develop more accurate forecasts that extend further out in time."

Advances in technology and modeling have led to remarkable improvements, although "we are constantly limited by computer power."

Today, the most powerful weather computers rely on hundreds of thousands of processors and, in many cases, millions of data points. The National Weather Service (NWS) in the U.S. currently relies on a pair of supercomputers with over 200 teraflops (a teraflop equals one trillion floating-point operations per second) of capacity. By way of comparison, China's Tianhe-2 ("Milky Way 2") supercomputer, which topped the June 2014 Top500 supercomputer rankings, delivers performance of up to 33.86 petaflops per second (since a petaflop is equal to 1,000 teraflops, Tianhe-2 provides nearly 170 times more raw processing power than the NWS has available to it).

Meanwhile, the Korean Meteorological Administration in South Korea is expanding its computing storage capacity to 9.3 petabytes in order to better predict weather events, including typhoons. The European Centre for Medium-Range Weather Forecasts (ECMWF) processes 300 million observations on a daily basis, producing about 65 terabytes of forecasts every day, with peaks of 100 terabytes. ECMWF's archive holds 65 petabytes of data, and it is growing at rate of approximately 50% annually, says software strategist Baudouin Raoult.

Interestingly, weather forecasting organizations worldwide rely on much of the same data derived from many of the same sources, Mass points out. Differences in forecasts typically revolve around the ways in which mathematicians and researchers approach statistical processing and how they average and round off numbers. In addition, "Web (forecasting services) obtain weather data from various sources and apply it in different ways," he explains. "The result is different forecasts but the underlying modeling is much less variable."

Kyger says current NWS models are more than 60% accurate beyond five days—generally considered a "skillful forecast" benchmark. Yet, because the physics and dynamics of the atmosphere are not directly proportional to expanding a grid resolution, it is not possible to rely on a static model or linear equation to extrapolate data. In fact, with every hardware upgrade, it can take up to a year to fine-tune a new model to the point where it outperforms an existing model. "At one point, a skillful forecast was only a day or two. The improvements are very gradual year over year, but they add up to the point where significant improvements take place," he explains.

Peter Bauer, head of ECMWF's Model Division for the European Centre, says predictive skills have improved to the point where researchers are witnessing about a one-day-per-decade improvement rate in forecasting. This means that today's six-day forecasts are about on par with the accuracy of five-day forecasts a decade ago. "In addition to extending the range and accuracy of large-scale forecasts, the techniques for predicting regional and local weather parameters such as precipitation, surface temperature, and wind have dramatically improved," he points out.

The Sky's the Limit

In practical terms, even a small improvement in forecasting quality can produce enormous benefits for individuals, businesses, and society, from providing warnings for short-term events such as tornados and floods to long-term issues such as how to construct buildings and design infrastructure. For instance, before Hurricane Sandy slammed the northeastern U.S. in October 2012, the ECMWF had successfully predicted the storm track and intensity of the event five days out, while NWS models lagged by about a day. The deviation in modeling focused attention on the perceived deficiencies of the NWS.

Kyger acknowledges the episode was a "disappointment" for the NWS. This led, in May 2013, to the U.S. Congress approving $23.7 million in supplemental funding to upgrade NWS systems from 90 teraflops to upward of 200 teraflops, as well as addressing other issues. However, U.S. forecasting technology continues to generate concerns. "There have been a number of important forecasts where U.S. prediction systems performed in an inferior way," Mass says. A recent blog posted by Mass stated that the U.S. had slipped into fourth place in global weather prediction, behind facilities in continental Europe, the U.K., and Canada.

"A major reason why the U.S. is falling behind is that the other centers are using far more advanced data assimilation or higher resolution, both of which require very substantial computer power, which the U.S. National Weather Service has been lacking," Mass explains. Over the last decade, Congress has not provided adequate funding to keep up with the fast-moving computing and data environment; "for a very modest cost, the United States could radically improve weather prediction," he says.

The upgrades to the NOAA supercomputers completed in August 2013 were part of the first phase of a two-step plan to increase its available processing power. Early results show that, in some cases, a 15% forecasting improvement has resulted. The computing power will increase to 1,950 teraflops in 2015, if current funding stays in place. NOAA operates the systems as a private cloud that is scalable. It uses these resources across agencies and tasks, and utilizes capacity in the 90%-plus range. Kyger says a cluster or grid approach that extends beyond NOAA is not feasible, for financial and practical reasons.

Meanwhile, the ECMWF is continuing to refine and improve its forecasting model. Moving forward, Bauer says, the Centre is attempting to focus on the environmental system in a far more comprehensive way, in order to gain a better understanding of key factors impacting weather, including greenhouse gasses, ocean temperatures, and sea ice. "The better the observations and the more critical data points we have, the better the mathematical methods," he explains. "More important in the future will be the prediction of extremes, which places a greater emphasis on predicting the right probabilities of events and doing so in a longer time range."

In the end, the quest for more accurate weather forecasts leads back to the need for more computing power and the development of better algorithms.

IBM's Deep Thunder initiative is further redefining the space. It has predicted snowfall accumulations in New York City and rainfall levels in Rio de Janeiro with upward of 90% accuracy by taking a somewhat unconventional approach. "We are not looking to use the traditional method of covering a large area with as high a resolution as possible using uniform information," Treinish says. "We are putting bounds on the computing problem by creating targeted forecasts for particular areas." As part of the initiative, IBM plugs in additional types of data sources—including agricultural measurements and wind farm data—and manipulates existing sources in different ways.

In fact, as the Internet of Things (IoT) takes hold, new types of sensors and crowdsourcing techniques will appear, and will further redefine weather forecasting. Kyger says the NWS has already started to experiment with crowdsourcing and other social media input, including data from hyperlocal Twitter accounts. Treinish believes smartphones and other devices could provide insights into everything from temperature and wind conditions to barometric pressure and humidity on a block-by-block level. The challenge, he says, is that the massive amount of data can be "really noisy and not of real high quality."

Adding to the challenge, the IoT will collect far more data, but at the same time will further tax existing and already constrained supercomputers.

In the end, the quest for more accurate forecasts leads back to the need for more computing power and the development of better algorithms; that, in turn, drives the need for even more powerful computers. There is an ongoing need for adequate funding and additional IT resources; it also is critical to continually upgrade models using an assortment of statistical and regression analysis techniques, combined with human analysis and judgement.

"The goal," Kyger says, "is to continually look for things that we can do better. It's a closed loop cycle that never ends."

Further Reading

Voyant, C., Notton, G., Paoli, C., Nivet, M.L., Muselli, M., Dahmani, K.
Numerical weather prediction or stochastic modeling: an objective criterion of choice for the global radiation forecasting, ***International Journal of Energy Technology and Policy*** **(2014), http://arxiv.org/abs/1401.6002.**

Krishnappa1, D.K., Irwin, D., Lyons, E., Zink, M.
CloudCast: Cloud Computing for Short-Term Weather Forecasts, ***Comput. Sci. Eng.*** **15, 30 (2013); http://dx.doi.org/10.1109/MCSE.2013.43**

Sawale, G.J., Gupta, S.R.
Use of Artificial Neural Network in Data Mining For Weather Forecasting, ***International Journal Of Computer Science And Applications*****, Vol. 6, No.2, Apr 2013, http://www.researchpublications.org/IJCSA/NCAICN-13/244.pdf**

Bainbridge, L.
Ironies of automation. ***New Technology and Human Error*****, J. Rasmussen, K. Duncan, J. Leplat (Eds.). Wiley, Chichester, U.K., 1987, 271–283.**

Roulstone, I., Norbury, J.
Computing Superstorm Sandy, ***Scientific American*** **309, 22 (2013), http://www.nature.com/scientificamerican/journal/v309/n2/full/scientificamerican0813-22.html**

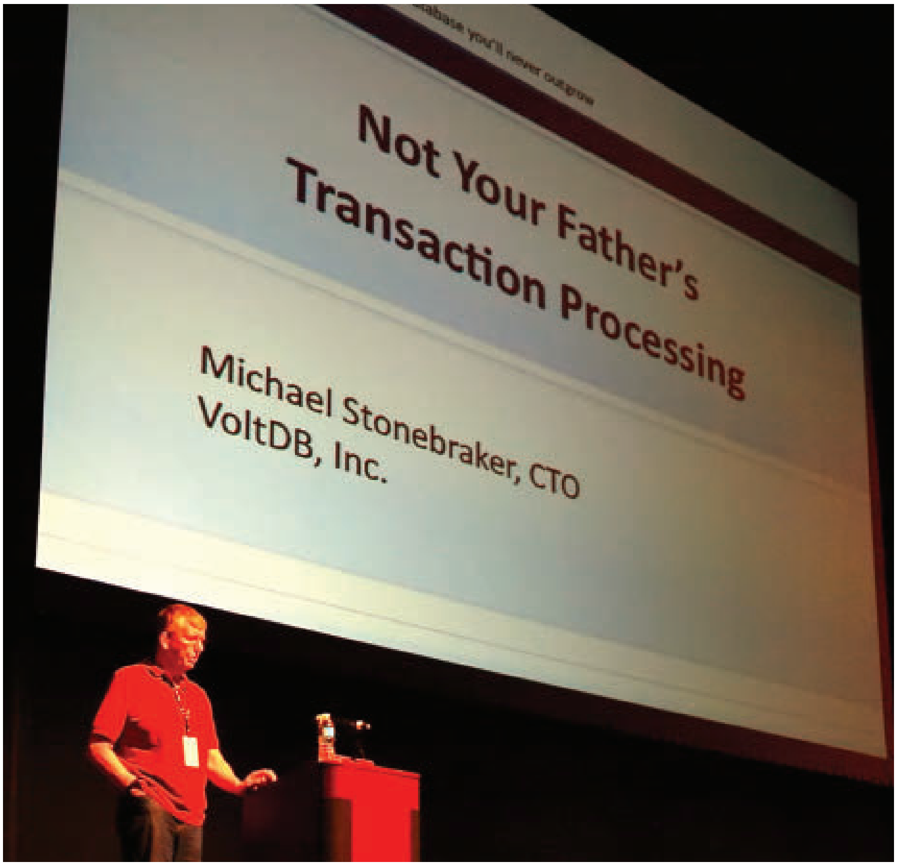
Samuel Greengard is an author and journalist based in West Linn, OR.

Technology | DOI:10.1145/2641229

Neil Savage

The Power of Memory

In-memory databases promise speedier processing.



Database expert and VoltDB co-founder Michael Stonebraker making a presentation on in-memory databases at the Strange Loop multidisciplinary conference in September 2012.

PHOTOGRAPH BY FRANK VILHELMSEN

KEEPING DATA IN MEMORY instead of pulling it in from a disk can speed up the processing of that data by orders of magnitude, which is why database companies have been vying for a share of the in-memory database market. IBM, Oracle, and Microsoft introduced versions of such databases this year, while SAP has been selling its Hana product for the past three years. Smaller companies including Aerospike, VoltDB, and MemSQL have all been getting into the act as well.

What they promise is a way to speed up activities that are important to businesses, such as processing transactions with their customers or analyzing the ever-growing quantities of information those transactions produce. "I think the potential is enormous," says Amit Sinha, senior vice president for marketing at SAP, headquartered in Walldorf, Germany. "Changing the data center from disk to memory has huge implications."

In-memory databases, also known as main memory databases, speed up processing in two basic ways: with the data available in memory, the time lag caused by fetching the data off a disk is erased; also, data tends to be stored on disk in blocks, and to get the one desired piece of data, the computer imports the whole block from disk, decodes it, runs the process on the piece it wants, re-encodes the block, and sends it back where it came from.

"The data structure and the algorithms for managing all that get quite complicated and have high overhead, but when you store data in memory, you get rid of all of that," says Paul Larson, principal researcher at Microsoft Research in Redmond, WA, who has worked on Microsoft's Hekaton in-memory database. "When data lives entirely in memory, you can be much more efficient."

It is difficult to quantify exactly how much of an efficiency boost an in-memory database provides; the answer is affected by the type and quantity of the data and what is done to it. "It's extremely workload dependent," Larson says.

Michael Stonebraker, a leading database expert and a co-founder of VoltDB of Bedford, MA, as well as an adjunct professor of computer science at the Massachusetts Institute of Technology (MIT) in Cambridge, MA, says processes can be "anywhere from marginally faster to wildly faster." Some might be improved by a factor of 50 or 100, he says.

Though the speed increase varies depending on the particular applications, some of SAP's customers have managed a 10,000-fold improvement, Sinha says, with many processes that used to take minutes now being done in seconds. That can be important to businesses, for instance by allowing them to run real-time fraud analysis. "Before the credit card drops, you need to have analysis of whether this transaction is valid," he says.

It might also allow new kinds of analyses. Imagine you work at World Wide Widgets, and your job is to make sure you buy sufficient raw materials and get them to the right factories to provide all the Walmarts in a particular region with just as many widgets as they are likely to sell—not more because holding onto inventory costs money, and not fewer because that means you have missed out on sales. To do this job, known in the supply chain industry as "materials requirement planning," you keep track of both the sales at the different stores and your supplies of raw materials. If, instead of doing such an analysis weekly or daily, you could run it in real time, you could fine-tune the process and get a more accurate balance between supply and demand. "That level of planning can really take enormous amounts of costs out of business," Sinha says.

Others wonder whether the case has really been made that using in-memory databases for data analytics provides a clear economic advantage. "It's interesting that they feel there's a good commercial market here," says Samuel Madden, a professor in the Computer Science and Artificial Intelligence Laboratory at MIT.

Madden does not dispute in-memory databases are faster. "You can pore through more records per second from memory than you can through a traditional disk-based architecture," he says. However, other methods of streamlining the process—from reducing the number of instructions to arranging data by columns rather than rows—might produce a similar increase in efficiency, or at least enough of a boost that the added expense of more memory is not worthwhile. "The performance of these things might not be all that different," Madden says.

Data analytics are only one part of the database world. Stonebraker divides it into one-third data warehouses, one-third online transaction processing (OLTP), and one-third "everything else"—Hadoop, graph databases, and so on. For OLTP, both Stonebraker and Madden say, the advantage is clearer.

With in-memory databases, "you can pore through more records per second from memory than you can through a traditional disk-based architecture."

Performing analytics on a data warehouse involves scanning many records, perhaps millions or billions, that change infrequently, if ever. In OLTP—say, processing an order on Amazon or moving money between bank accounts—the computer touches a small amount of data and performs a small operation on it, such as updating a balance. Being able to perform such transactions faster means a company can either do more of them in a given period of time, increasing its business, or do the same amount with less processing hardware, saving infrastructure costs. "Instead of tens of milliseconds per transaction, you can go to hundreds of microseconds, and in some applications that really matters," Madden says. Computerized trading on Wall Street might be one beneficiary; "if you can issue your trade 1ms faster than the other guy, you can make more money."

From an economic standpoint, OLTP might be a better fit for in-memory databases because of their relative sizes, Stonebraker says. A one-terabyte OLTP database would be considered large; 10 TB would be huge. A terabyte of main memory today costs less than $30,000, so the efficiency gain could make that affordable for some companies. On the other hand, he says, "data warehouses are getting bigger at a faster rate than main memory is getting cheaper." For instance, the gaming company Zynga has amassed around five petabytes in its data warehouse. "You're not about to buy 5PB of main memory," Stonebraker says.

Larson sees the analytics side growing as well. "The general trend is toward more real-time analytics, partly because now we can do it," he says "It's technologically feasible," because DRAM has gotten denser and cheaper,

Milestones

Computer Science Awards, Appointments

DONGARRA RECEIVES KEN KENNEDY AWARD

Jack Dongarra of the University of Tennessee has received the ACM-IEEE Computer Society Ken Kennedy Award, awarded annually in recognition of substantial contributions to programmability and productivity in computing and substantial community service or mentoring contributions.

Dongarra, Distinguished University Professor at the University of Tennessee, is founder and director of the Innovative Computing Laboratory at the university, and holds positions at Oak Ridge National Laboratory and the University of Manchester.

He was cited for his leadership in designing and promoting standards for software used to solve numerical problems common to high-performance computing (HPC). Dongarra's work has led to the development of major software libraries of algorithms and methods that boost performance and portability in HPC environments.

MOLER RECEIVES VON NEUMANN MEDAL

The Institute for Electrical and Electronic Engineers (IEEE) awarded its 2014 John Von Neumann Medal for outstanding achievements in computer-related science and technology to Cleve Moler, chief mathematician of MathWorks, for his "fundamental and widely used contributions to numerical linear algebra and scientific and engineering software that transformed computational science."

Recipient of the IEEE Computer Society's 2012 Computer Pioneer Award, Moler was a professor of mathematics and computer science for almost 20 years at the University of Michigan, Stanford University, and the University of New Mexico. Before joining MathWorks in 1989, he also worked for Intel Hypercube and Ardent Computer Corp.

REDDY NAMED FELLOW OF NATIONAL ACADEMY OF INVENTORS

The National Academy of Inventors (NAI) has named Raj Reddy, Mozah Bint Nasser University Professor of Computer Science and Robotics at Carnegie Mellon University (CMU), to its roster of NAI Fellows.

NAI Fellows are recognized for their "prolific spirit of innovation in creating or facilitating outstanding inventions and innovations that have made a tangible impact on quality of life, economic development and the welfare of society."

Reddy received the ACM A.M. Turing Award in 1994 for his work in pioneering practical, large-scale artificial intelligence systems.

His research interests beyond speech recognition include robotics, human-computer interaction, innovations in higher education and efforts to bridge the "digital divide," particularly for people in developing nations. He also initiated CMU's autonomous vehicle program.

making it possible to hold so much data in memory, and because processing power has increased enough to handle all that data.

Yet DRAM, of course, is volatile; if the power goes out, the data goes away. That means there always needs to be a backup, which can add cost and slow performance. One way to back up data is to replicate it on a separate DRAM with a different power source, which means a company can switch to its backup with little delay. "How many replicas you run depends on your level of paranoia and how much money you have," Larson says.

Another approach is to make a slightly out-of-date copy of the data on a disk or in flash memory, and keep a log of transaction commands in flash. If data is deleted, the stale copy can be moved from the disk and the log used to recreate the transactions until the data is up to date. Exactly how a company handles backup depends on how fast it needs to restore, and how much it can afford to spend. Madden points out, though, "These database systems don't crash all that often."

Crashes may be less of a concern in the future, if makers of memory such as Intel introduce a non-volatile technology, such as magnetic RAM or phase-change memory. Larson says researchers say the future of memory seems promising. "The technology is beginning to look good enough and the capacities are looking quite large," he says. "What they don't want to talk about at this point is when and at what price."

Non-volatile memory will likely be slower than DRAM, but if it is significantly less expensive, that could be a worthwhile trade-off, Larson says.

Another issue that could affect the in-memory field is the existence of distributed databases. When the data an application needs is spread out among different machines, the advantages of in-memory may disappear. "Building a database system on top of the shared memory system is problematic," says Larson. "I don't really see how to make them efficient."

Stonebraker is more blunt. "Distributed memory is just plain a bad idea," he says. "No serious database system runs on distributed memory."

Sinha, on the other hand, says distributed memory can work. With fast interconnects, sometimes it is easier to access the main memory of a nearby machine, he says. It is also important to make sure a piece of data is only written to one place in memory, and that data is organized so it rarely has to cross a partition. "You can be intelligent in keeping data together," he says.

> **"Distributed memory is just plain a bad idea. No serious database system runs on distributed memory."**

Stonebraker sees in-memory databases as eventually taking over online transactions. "I think main memory is the answer for a third of the database world," he says. He expects that takeover to take the rest of the decade to happen, while algorithms mature and businesses examine the value of the technology. "It's early in the market," he says.

Further Reading

Lahiri, T., Niemat, M-A., Folkman, S.
Oracle TimesTen: An In-Memory Database for Enterprise Applications, ***Bull. IEEE Comp. Soc. Tech. Comm. Data Engrg. 36*** **(2), 6-13, 2013.**

Lindström, J., Raatikka, V., Ruuth, J., Soini, P., Vakkuila, K.
IBM solidDB: In-Memory Database Optimized for Extreme Speed and Availability, ***Bull. IEEE Comp. Soc. Tech. Comm. Data Engrg. 36*** **(2), 14-20, 2013.**

Kemper, A., Neumann, T., Finis, J., Funke, F., Leis, V., Mühe, H., Mühlbauer, T, Rödiger, W.
Processing in the Hybrid OLTP & OLAP Main-Memory Database System HyPer, ***Bull. IEEE Comp. Soc. Tech. Comm. Data Engrg. 36*** **(2), 41-47, 2013.**

Zhang, C., Ré, C.
DimmWitted: A Study of Main-Memory Statistical Analytics, ***ArXiv*****, 2014.**

SAP's Hasso Plattner on Databases and Oracle's Larry Ellison https://www.youtube.com/watch?v=W6S5hrPNr1E

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

ACM Member News

BORGMAN CONTEXTUALIZES DATA FOR MEANING

Contextualizing and dissecting data is Christine L. Borgman's specialty. A professor and Presidential Chair in Information Studies at the University of California, Los Angeles, Borgman helps scholars, computer science professionals, engineers, and policy makers "understand data."

Borgman says her research "aims to contextualize data by applying knowledge of scientific data practices to the design of data collection and management tools, and to the design and policy of information services for research and education.

"When people talk about mining big data, they're missing how difficult it is to integrate and interpret data derived from different sources."

The Detroit native got her B.A. in mathematics from Michigan State University, an M.S. in library science from the University of Pittsburgh, and a Ph.D. in communications "with a heavy dose of communications technology and social media," she says, from Stanford University.

Borgman's mother, a university librarian, urged her early in her academic career to consider working with computers; that led, she recalls, to an "amazing opportunity" working as lead systems analyst automating the Dallas Public Library's card catalog, written in assembler code.

"Building algorithms to match data records is the easier part," Borgman says; the greater challenge is to contextualize data in scientific practice, which requires complex documentation of many related digital objects and software code.

"Designing a UI that supports human inquiry and interpretation is extremely difficult. But as scholarship in all fields becomes more data-intensive and collaborative, the ability to share, compare and reuse data is more essential."

—Laura DiDio

Society | DOI:10.1145/2641227

Gregory Mone

The New Digital Medicine

Affordable, connected, personal medical devices are slowly changing the nature of health care.

THE AVERAGE PERSON sees his or her physician for about an hour each year. At our annual checkups, doctors and nurses check our blood pressure, heart rate, and other vital signs, and from those brief snapshots they attempt to determine our overall health. Until recently, monitoring these metrics outside the office, and over long stretches of time, would have been neither affordable nor efficient. Today, however, the average person carries a versatile medical gadget on them at all times.

"A smartphone has become a medical device of the highest potential," says Sreeram Ramakrishnan, manager of Insights-driven Wellness Services with IBM's Health Informatics group. "The type of sensors that have already evolved are clearly establishing that there's no technological limit. You could capture almost anything you want."

Smartphones can measure your heart rate, count your steps, and tell you how well you sleep at night. It is not just our phones; there has been a boom in health-centric devices offering a broad range of medically relevant statistics.

The BAM Labs sensor, packed deep inside a mattress, monitors heart rate, respiration, and overall sleep quality throughout the night. The popular RunKeeper app, which makes use of a smartphone's GPS to track its users' running routes and speed, can also talk to cloud-linked scales that measure your weight and body mass index, over-the-counter blood pressure cuffs, and wristbands that keep tabs on your activity during the day, when you are not out racing. All of this data is then pulled together into an easy-to-read graph that offers a more complete picture of your health.

Handyscope, the "first mobile connected dermatoscope," combines smartphone technology with a sophisticated tool for skin cancer screening.

IMAGE COURTESY OF HANDYSCOPE.NET/FOTOFINDER SYSTEMS

For several years now, experts have been saying this sort of digital technology could revolutionize the health care system, leading to a fundamental shift in the way patients interact with their doctors. Instead of seeing physicians only when something goes wrong, patients using these apps and devices could work together with their providers to lead healthier lives and reduce the frequency of sick visits. "Today we have a sick care system, not a wellness system," says Stephen Intille, a computer scientist in the Personal Health Informatics group at Northeastern University in Boston, MA. "You'd like to see a world where you have these personal devices keeping you healthy."

For several years, experts have been saying digital technology could revolutionize the health care system.

The technology could also be a tremendous help to people battling long-term problems, such as hypertension, by allowing them to track their progress over time and potentially address issues before they become too serious. For this more efficient, data-driven health care system to become a reality, though, several significant hurdles need to be overcome.

Moving Beyond the Fitness Crowd

First, many of these health-centric devices are not always accurate. IBM's Ramakrishnan points out that if you wear a pedometer on each leg, their step counts vary significantly at the end of the day. Yet he says it is not the technology alone that prevents these gadgets from having a greater impact on health care. The accuracy and reliability of the devices will improve, but the technology also has to reach beyond the fitness crowd to a broader audience. "These devices have centered onto a self-selected audience of people who are already healthy," he notes.

At Northeastern University, Intille is working on strategies to engage patients who need the most help, such as those suffering from hypertension, by empowering them to take control of their own health.

The more popular fitness applications often use competition as a motivator, allowing users to compare themselves to others. Yet Intille says this will not work with the average person hoping to lower his or her blood pressure or lose weight. If your phone suddenly prompts you to get up and move because you have been sedentary too long, or even just pings you to take your medicine, you will most likely ignore the advice. "If you design a technology that tells people what to do, that becomes very annoying," he says. "Spouses can't get away with that. Why would we think a mobile phone would?"

Instead, Intille and his group are working on a more positive approach. For example, they are developing applications that tap into a phone's accelerometer to make assumptions about a user's posture and ambulation—whether the individual is up and moving around—and then compare this information to the individual's baseline behavior in real time. When the individual goes for a walk at a time they typically are sedentary, the application might offer the person a digital pat on the back in the form of an encouraging message. "We wait until they do something better than average, then give them subtle reinforcement," he explains.

> **"Physicians are overloaded already. If you bring in all your data, are they going to be able to understand it? Will they have time to look at it?"**

IMAGES COURTESY OF OWLETCARE.COM/OWLET PROTECTION ENTERPRISES, LLC

The Owlet Smart Sock wirelessly transmits a child's health data to a parent's smartphone via Bluetooth 4.0.

Blogosphere

Top 5 Blog@ CACM Posts

BASED ON PAGE VIEWS JANUARY 1, 2014 THROUGH JULY 15, 2014

1. Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities
Philip Guo
July 7, 2014
http://bit.ly/W0vtox
Page views: 67,881

2. Those Who Say Code Does Not Matter
Bertrand Meyer
April 15, 2014
http://bit.ly/1mNqout
Page views: 52,007

3. Data Science Workflow: Overview and Challenges
Philip Guo
Oct. 30, 2013
http://bit.ly/1sjQxmP
Page views: 4,406
Page views since posting: 9,897

4. Why Scientists and Engineers Must Learn Programming
Philip Guo
July 18, 2013
http://bit.ly/12yGZUs
Page views: 2,950
Page views since posting: 7,522

5. Teaching Programming To A Highly Motivated Beginner
Philip Guo
Oct. 31, 2012
http://bit.ly/U5lDjC
Page views: 2,480
Page views since posting: 37,894

Data source: Google Analytics at press time.

Intille and colleagues at Duke University are currently conducting a randomized trial of a weight-loss app. The results are not yet available, but Intille says this kind of study is one of the other missing pieces when it comes to personal health devices. "Most of what's out there in the commercial world, it hasn't been proven that it works," Intille notes. "If we can provide evidence that some of these things do work, more than just having people buy the stuff, then I think you'll see rapid adoption."

Dealing with the Data

At the same time, the prospect of rapid adoption is not exciting to everyone. Steven Steinhubl, director of Digital Medicine at the Scripps Translational Science Institute, says many health care providers express concern at the notion of patients tracking their own vital signs. They worry these devices could have a negative impact on the doctor-patient relationship, and that doctors will end up seeing their patients even less than they do today.

There is also some concern patients will not know what to do with the rush of data they collect, and that all these numbers could boost anxiety rather than health. Doctors could be equally stumped as to how to deal with all that information. "Many of us would like to see a world where you could collect data yourself and then bring it in to your physician," says Intille. "The challenge is that physicians are overloaded already. If you bring in all your data, are they going to be able to understand it? Will they have time to look at it?"

Moore believes the shift in approach was just as important. The patients were encouraged to make health decisions on their own.

John Moore, an MIT Media Lab alumnus, recently co-founded Twine Health, a startup focused on forging more productive digital connections between patients and their health care providers. A former physician, Moore envisions patients working with their doctors to establish certain health goals, then using devices and apps to track their progress. In one of his group's studies, patients initially consulted with a doctor or nurse and set a goal of lowering their blood pressure to a certain level. The providers avoided the paternalistic approach common to the traditional doctor-patient relationship, in which the physician simply told the patient what to do. Instead, they mutually agreed on how to use a combination of exercise, diet, and medicine to lower their blood pressure.

After that initial meeting, the patients interacted with a tablet application that displayed their daily schedules and informed them when certain actions were due to be completed. They would indicate when they took the prescribed pills, and they regularly used a smart blood pressure cuff that relayed their vital signs to the tablet and to the physician's office. As with Intille's work, there was no pestering, but patients could converse with a nurse through the tablet and receive encouragement for sticking to the regimen.

The results, Moore says, were outstanding. "One hundred percent of the patients reached their blood pressure goal," he says. "We achieved dramatically better results at substantially lower costs."

The savings were attributed in part to the tablet app, which allowed the patients to connect with their providers without having to schedule an office visit. Yet, Moore believes the shift in approach was just as important. The patients were encouraged to make health decisions on their own. "People really responded to that," Moore says. "When we interviewed them at the end, they said no one had ever asked them to be in charge."

A Concerted Effort

Regardless of whether or not the larger population chooses to be more proactive in the next few years, the boom in connected medical devices does not look to be slowing down. Rumored smartwatches from HTC and other companies, possibly even Apple, would open the doors for even more health-related data. Since it would regularly be in contact with our skin, a watch could potentially gather more accurate data on stress levels and offer continuous heart rate monitoring.

Joe Bondi, chief technical officer at RunKeeper, is excited about the possibility of more accurate, continuous heart rate monitors, since existing technology does not work as well for users on the run. A monitor built into ear buds, for example, might be able to deliver better results. "There's a lot you can do in terms of coaching if you know someone's heart rate," he says.

Despite the pace of technological development, experts caution all these devices and digital platforms will not force major change quickly. "I do believe that this is the future of healthcare, but you can't just throw the technology out there and hope it will cure everything," says Steinhubl.

"It's going to take a really concerted effort of payers and providers and industry support to say we're going to totally reengineer health care and take advantage of this technology." C

Further Reading

Eric Topol
The Creative Destruction of Medicine. **Basic Books, 2012.**

Boulous, M., Wheeler, S., Tavares, C., Jones, R.
How Smartphones are Changing the Face of Mobile and Participatory Healthcare. ***Biomedical Engineering Online*, 10:24, 2011.**

Dunton, G.F., Dzubur, E., Kawabata, K., Yanez, B., Bo, B., and Intille, S.
Development of a Smartphone Application to Measure Physical Activity Using Sensor-Assisted Self-Report. ***Frontiers in Public Health*, 2013.**

Moore, J., Marshall, M.A., Judge, D., et. al.
Technology-Supported Apprenticeship in the Management of Hypertension: A Randomized Controlled Trial. ***JCOM*, March 2014.**

"The Wireless Future of Medicine" A video overview of the promise of connected medical devices: http://www.ted.com/talks/eric_topol_the_wireless_future_of_medicine

Gregory Mone is a Boston, MA-based writer and the author of the children's novel *Dangerous Waters.*

DOI:10.1145/2644146

Stefan Bechtold and Adrian Perrig

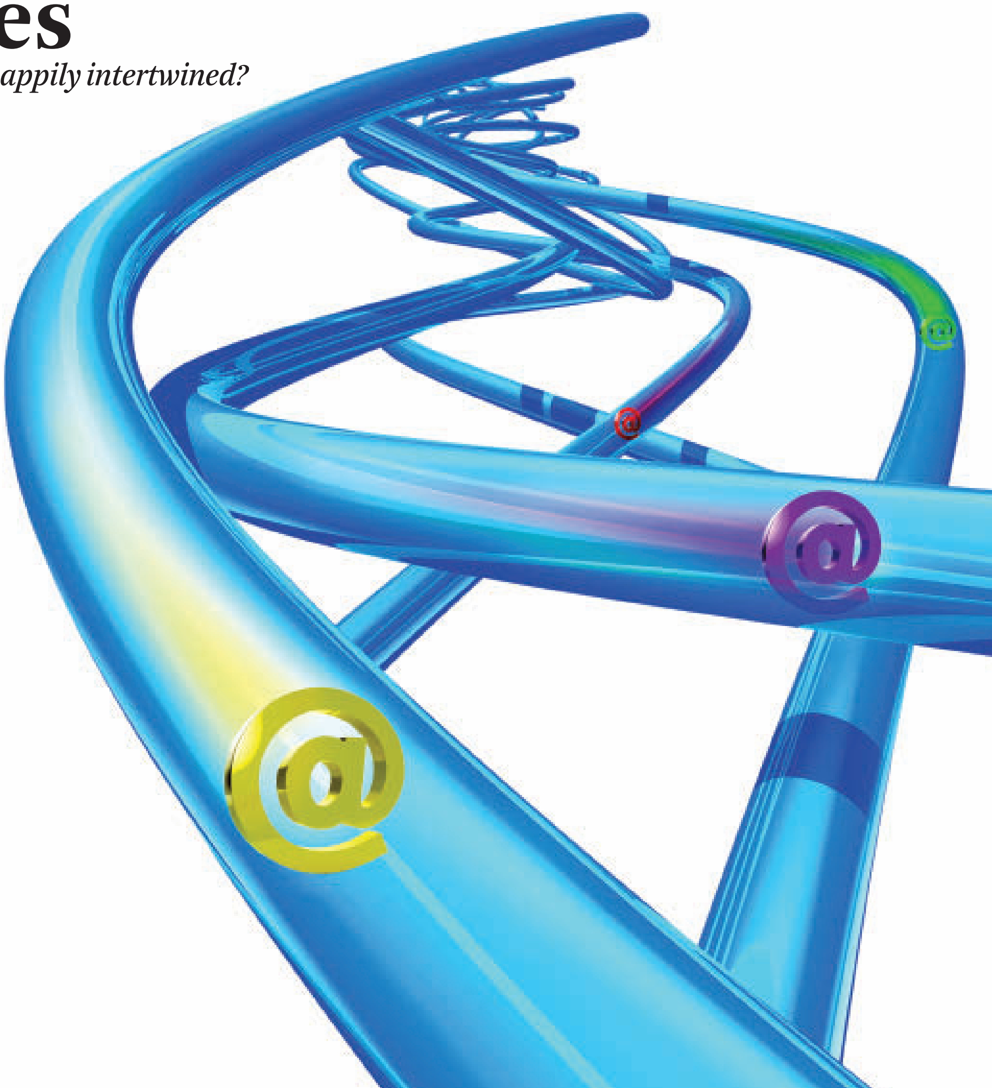
Law and Technology

Accountability in Future Internet Architectures

Can technical and legal aspects be happily intertwined?

When the Internet architecture was designed some 40 years ago, its architects focused on the challenges of the time. These included the creation of a distributed communication network that is robust against packet loss and other network failures; support across multiple types of networks and communication services; and the management of Internet resources in a cost-effective and distributed way. As history has shown, the Internet's architects succeeded on many dimensions. The phenomenal success of the Internet has often been attributed to its basic architectural principles.

As the uses of the Internet have expanded beyond the original creators' wildest dreams, its protocols have been stretched to accommodate new usage models, such as mobile, video, real-time, and security-sensitive applications. A string of extensions has resulted in an infrastructure that has increasingly become ossified due to the numerous constraints each extension



introduces, in turn complicating further extensions. These challenges have prompted researchers to rethink architectural principles, thereby engaging in visionary thinking about what a future Internet architecture, which should last for many decades, should look like.

One important dimension of clean-slate Internet architecture proposals is to rethink the role of accountability. The general idea is that accountability for one's actions would enable identification of the offender, making it possible to either defend oneself against misbehavior or deter it altogether. It is therefore natural to consider accountability as a way of addressing network attacks, ranging from route hijacking, to various kinds of network denial-of-service attacks and remote exploitation of host vulnerabilities. Increased accountability could not only address some of the technical shortcomings of the current Internet architecture. It could also enable various partly legal solutions to problems which, to date, have not been solved by purely technical means.

In recent years, security incidents have repeatedly stressed the need for accountability mechanisms. We highlight the use of accountability to address the hijacking of Internet traffic routing by altering or deleting authorized Border Gateway Protocol (BGP) routes. In 2008, YouTube became globally unreachable after a Pakistani Internet service provider (ISP) altered a route in an attempt to block YouTube access in Pakistan. In 2013, the network intelligence firm Renesys documented that traffic routes from Mexico to Washington, D.C., and from Denver to Denver had been rerouted via Belarus and Iceland. In March 2014, Google's Public Domain Name System (DNS) server, which handles approximately 150 billion queries a day, had its IP address hijacked for 22 minutes. During this time, millions of Internet users were redirected to British Telecom's Latin America division in Venezuela and Brazil. Such rerouting, whether deliberate or not, abuses the implicit trust enshrined in the BGP routing protocol. Traffic rerouting is often difficult to detect for both Internet users and network operators. It can be used for a wide range of attacks. Despite the introduction of BGPSEC (a security protocol that promises to stop hijacking attacks), accountability—which makes it possible for an attacker to be identified, sued, and prosecuted—may prove a better solution to the hijacking problem.

Another example where accountability matters is the network neutrality debate. Insufficient accountability mechanisms in today's Internet prevent consumers from finding out why their access to particular services has been blocked or slowed down. Is today's access to Hulu slow due to technical problems at Hulu's servers, due to delays somewhere in the network, or due to bandwidth limitations between your ISP and your home network? It is difficult to determine. More generally, if a technical architecture does not provide means for users to monitor whether service providers keep their promises with regard to service quality and features, service providers may have insufficient incentives to actually keep their promises.

An architecture that leaves loopholes in legal and technical accountability has it costs. As the Internet traffic hijacking example shows, it may encourage unlawful online activities, with all the negative effects this entails for society. As the network neutrality example demonstrates, it may deter business partners from entering into contractual agreements, as their terms may be unenforceable.

Currently, manifold attempts are being made to deal with accountability loopholes. On the legal front, legislators and government agencies are designing rules to provide network providers and users with the right incentives despite limited accountability. In the ongoing battle over network neutrality regulations, for example, the U.S. Federal Communications Commission (FCC) has proposed rules that will force ISPs to disclose their network management practices.[a] In June 2014, the FCC announced it would investigate the impact peering agreements between ISPs such as Comcast and Verizon and content providers such as Netflix have on broadband consumption and Internet congestion.

Security incidents have repeatedly stressed the need for accountability mechanisms.

On the technical front, any technology aimed at increasing accountability should provide irrefutable proof that parties have performed certain actions: in particular, of who is being held accountable for what action to whom. End users, hosts, ISPs (or their routers and network equipment), service operators, or content providers could all potentially be held accountable or be enabled to verify the accountability. Consider a system that would hold an ISP's routers accountable for delayed packet forwarding. It would have to ensure the routers cannot hide the fact they delayed forwarding a packet. Such accountability for delays could serve as a technical measure to validate the network neutrality of an ISP.

Researchers have proposed numerous technical solutions for various types of accountability. Bender et al. propose to hold the source accountable for packets created, and enable each router to verify.[2] Such packet origin accountability is a popular property, which subsequent researchers have pursued with varying assumptions and approaches for cryptographic key setup.[1,3,7] Li et al. propose a general key setup mechanism between sources and network routers to enable packet origin, router forwarding, and routing message accountability.[6] Naous et al. propose a system for packet origin and strong router forwarding accountability.[9] Zhou et al.[11] propose a strong notion of making the network accountable for any state it may have ("secure network provenance"). The same authors have extended their work to also provide time-aware provenance.[12]

[a] This aspect of the proposed Open Internet Rules has not been affected by the January 2014 decision of the U.S. Court of Appeals for the District of Columbia, which struck down anti-blocking and anti-discrimination obligations.

Implementing only legal or technical measures to increase accountability on the Internet has limitations. We believe it is a fruitful exercise to combine technical and legal aspects for two reasons. First, this challenges perceptions lawyers have about technology and vice versa. As the Internet traffic hijacking and the network neutrality examples demonstrate, it is often difficult to identify what caused network errors. From a legal perspective, lacking identifiability makes it impossible to hold someone accountable for the error. This, in turn, reduces everyone's incentive to prevent network errors, as the risk of being held liable is low. All too often, the legal debate simply assumes such accountability loopholes are a given fact on the Internet. The debate has not considered how liability regimes and the types of contracts and services offered on the Internet would change if a future Internet architecture were to provide enhanced accountability mechanisms. The current lack of accountability, for example, prevents service level agreements that span beyond a single autonomous system. Accountability for network operations could enable an ISP to provide inter-ISP service-level agreements, as the ISP could restrict his liability to internal errors, thereby excluding external errors that can be attributed to the appropriate responsible party. Increasing accountability could thus make liability risks manageable and contractable.

Second, by combining technical and legal aspects of accountability in network design, we can focus on trade-offs in network design decisions that might otherwise pass unnoticed. An important issue is the trade-off between accountability and privacy. Usually they are in conflict, as accountability requires sacrificing privacy.[5] However, in some cases, both can be achieved. For example, Mallios et al. have proposed a system where privacy is achieved as long as a user does not misbehave, whereas misbehavior will render the user accountable.[8,b] Another important trade-off exists between accountability and personal freedom. Lessig argues that e-commerce will require accountability at the cost of personal freedom.[5] There might be other issues here. If everyone's actions on the Internet were traceable, how could political activists communicate under oppressive political systems? How could highly privacy-sensitive citizens communicate? Technical solutions such as anonymous communication systems implemented as an overlay network on the Internet can achieve anonymous communication despite a traceable or accountable underlying network architecture. The important research question is how the two properties can be meaningfully combined. The answer may be something similar to the privacy example described previously: As long as users communicate within some defined traffic pattern, their communications remain anonymous. If they deviate from the pattern, their (potential mis-) behavior can be traced back. It is also worth noting that increased accountability can be advantageous to political activists. In societies where governments control Internet traffic within the country and across borders, increased accountability can impede unobtrusive censorship, as the increase in transparency makes it more difficult for the government to hide its censoring activities.

Many design decisions have implications for social interactions that lie in the realm of the law.

We cannot offer any easy ways to deal with such trade-offs. We can, however, observe that many important problems in today's Internet are due to a lack of accountability and transparency. The response—to increase accountability—is not a mere technical enterprise. Many design decisions have implications for social interactions that lie in the realm of the law. Because law and technology are sometimes interchangeable and sometimes lead to difficult trade-offs, legal considerations should be taken into account not only after a novel Internet architecture has been implemented, but as an integral part of the design process of the architecture itself.[4,10] Such an approach could do more than enhance the value of the architecture itself. Increased accountability may also produce novel services that we cannot envision at present, precisely because of accountability loopholes that affect the current Internet.

As the interaction between network usage and the law increases, the network's technical architecture must cope with trade-offs and policy values that have long been familiar within the legal system. It is one of the challenges of future Internet architecture design to develop holistic approaches that will integrate technical and legal aspects and enable researchers and developers to be versatile in both fields.

[b] This works like the detection of double spending in digital cash: a payment is untraceable as long as the user spends the coin only once, but the identity is revealed if the coin is spent twice.

References

1. Andersen, D.G. et al. Accountable Internet Protocol (AIP). In *Proceedings of ACM SIGCOMM*, 2008.
2. Bender, A. et al. Accountability as a service. In *Proceedings of USENIX SRUTI*, 2007.
3. Andersen, D., Parno, B., and Perrig, A. SNAPP: Stateless network-authenticated path pinning. In *Proceedings of AsiaCCS*, March 2008.
4. Flanagan, M., Howe, D.C., and Nissenbaum, H. *Embodying Values in Technology: Theory and Practice.* Cambridge University Press, Cambridge, 2008, 322–353.
5. Lessig, L. *Code and Other Laws of Cyberspace.* Basic Books, NY, 1999.
6. Li, A., Liu, X., and Yang, X. Bootstrapping accountability in the Internet we have. In *Proceedings of USENIX NSDI*, 2011.
7. Liu, X. et al. Passport: Secure and adoptable source authentication. In *Proceedings of USENIX NSDI*, 2008.
8. Mallios, Y. et al. Persona: Network layer anonymity and accountability for next generation Internet. In IFIP TC 11 International Information Security Conference, May 2009.
9. Naous, J. et al. Verifying and enforcing network paths with ICING. In *Proceedings of ACM CoNEXT*, 2011.
10. Nissenbaum, H. How computer systems embody values. *IEEE Computer 34*, 3 (2001), 118–120.
11. Zhou, W. et al. Secure network provenance. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, October 2011.
12. Zhou, W. et al. Distributed time-aware provenance. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, August 2013.

Stefan Bechtold (sbechtold@ethz.ch) is Professor of Intellectual Property at ETH Zurich and a *Communications* Viewpoints section board member.

Adrian Perrig (adrian.perrig@inf.ethz.ch) is Professor of Computer Science at ETH Zurich.

The authors would like to thank Srdjan Capkun, Susanne Hambrusch, John L. King, and Timothy Roscoe for helpful feedback.

DOI:10.1145/2644148

Thomas Haigh

Historical Reflections
We Have Never Been Digital

Reflections on the intersection of computing and the humanities.

This column is inspired by the fashionable concept of the "digital humanities." That will be our destination rather than our starting point, as we look back at the long history of the idea that adoption of computer technology is a revolutionary moment in human history. Along the way we will visit the work of Nicholas Negroponte and Bruno Latour, whose books *Being Digital* and *We Have Never Been Modern* I splice to suggest that we have, in fact, never been digital.

The computer is not a particularly new invention. The first modern computer programs were run in 1948, long before many of us were born. Yet for decades it was consistently presented as a revolutionary force whose imminent impact on society would utterly transform our lives. This metaphor of "impact," conjuring images of a bulky asteroid heading toward a swamp full of peacefully grazing dinosaurs, presents technological change as a violent event we need to prepare for but can do nothing to avert.

Discussion of the looming revolution tended to follow a pattern laid out in the very first book on electronic computers written for a broad audience: Edmund Callis Berkeley's 1949 *Giant Brains: Or Machines That Think*.[1] Ever since then the computer has been surrounded by a cloud of promises and predications, describing the future world it will produce.

The specific machines described in loving detail by Berkeley, who dwelled on their then-novel arrangements of relays and vacuum tubes, were utterly obsolete within a few years. His broader hopes and concerns for thinking machines, laid out in chapters on "what they might do for man" and "how society might control them" remain much fresher. For example, he discussed the potential for autonomous lawnmowers, automated translation, machine dictation, optical character recognition, an "automatic cooking machine controlled by program tapes," and a system by which "all the pages of all books will be available by machine." "What," he asked, "shall I do when a robot machine renders worthless all the skills I have spent years in developing?"

Computer systems have always been sold with the suggestion they represent a ticket to the future. One of my favorite illustrations of this comes from 1953, when W.B. Worthington, a business systems specialist, promised at a meeting of his fellows that "the changes ahead appear to be similar in character but far beyond those effected by printing." At that point no American company had yet applied a computer to administrative work, and when they did the results would almost invariably disappoint. The machines needed more people than anticipated to tend them, took longer to get running, and proved less flexible. So why did hundreds of companies rush into computerization before its economic feasibility was established? Worthington had warned that "The first competitor in each industry to operate in milliseconds, at a fraction of his former overhead, is going to run rings around his competition. There aren't many businesses that can afford to take a chance on giving this fellow a five-year lead. Therefore, most of us have to start now, if we haven't started already."[a]

Computer systems have always been sold with the suggestion they represent a ticket to the future.

Following his belief that "the ominous rumble you sense is the future coming at us." Worthington was soon to give up his staff job at Hughes Aircraft in favor of a consulting role, promoting his own expertise as a guide toward the electronic future. He had promised that "We can set our course toward push-button administration, and God willing we can get there." Similar statements were being made on the pages of the *Harvard Business Review* and in speeches delivered by the leaders of IBM and other business technology companies as a

[a] W.B. Worthington. "Application of Electronics to Administrative Systems," *Systems and Procedures Quarterly 4,* 1 (Feb. 1953), 8–14. Quoted in T. Haigh, "The Chromium-Plated Tabulator: Institutionalizing an Electronic Revolution, 1954–1958," *IEEE Annals of the History of Computing* 23, 4 (Oct.–Dec. 2001), 75–104.

broad social alliance assembled itself behind the new technology.

After this initial surge of interest in computerization during the 1950s there have been two subsequent peaks of enthusiasm. During the late 1970s and early 1980s the world was awash with discussion of the information society, post-industrial society, and the microcomputer revolution. There followed, in the 1990s, a wave of enthusiasm for the transformative potential of computer networks and the newly invented World Wide Web.

Rupture Talk and Imaginaires

Discussion of the "computer revolution" was not just cultural froth whipped up by the forces of technological change. Instead the construction of this shared vision of the future was a central part of the social process by which an unfamiliar new technology became a central part of American work life. Patrice Flichy called these collective visions "imaginaires" and has documented their importance in the rapid spread of the Internet during the 1990s.[2] Rob Kling, a prolific and influential researcher, wrote extensively on the importance of "computerization movements" within organizations and professional fields.[5]

Historian of technology Gabrielle Hecht called such discussion "rupture talk" in her discussion of the enthusiasm with which France reoriented its colonial power and engineering talent during the 1950s around mastery of nuclear technology.[4] This formulation captures its central promise: that a new technology is so powerful and far-reaching it will break mankind free of history. Details of the utopian new age get filled in according to the interests, obsessions, and political beliefs of the people depicting it. That promise is particularly appealing to nations in need of a fresh start and a boost of confidence, as France then was, but its appeal seems to be universal. This dismissal of the relevance of experience or historical precedent carries out a kind of preventative strike on those who might try to use historical parallels to argue that the impact of the technology in question might in fact be slower, more uneven, or less dramatic than promised. Yet this fondness for rupture talk is itself something with a long history around technologies such as electric power, telegraphy, air travel, and space flight.

Enter "The Digital"

One of the most interesting of the cluster of concepts popularized in the early 1990s to describe the forthcoming revolution was the idea of "the digital" as a new realm of human experience. Digital had, of course, a long career as a technical concept within computing. It began as one of the two approaches to high-speed automatic computation back in the 1940s. The new breed of "computing machinery," after which the ACM was named, was called digital because the quantities the computer calculated with were represented as numbers. That is to say they were stored as a series of digits, whether on cog wheels or in electronic counters, and whether they were manipulated as decimal digits or the 0s and 1s of binary. This contrasted with the better-established tradition of analog computation, a term derived from the word "analogy." In an analog device an increase in one of the quantities being modeled is represented by a corresponding increase in something inside the machine. A disc rotates a little faster; a voltage rises slightly; or a little more fluid accumulates in a chamber. Traditional speedometers and thermometers are analog devices. They creep up or down continuously, and when we read off a value we look for the closest number marked on the gauge.

Throughout the 1950s and 1960s analog and digital computers coexisted. The titles of textbooks and university classes would include the word "analog" or "digital" as appropriate to avoid confusion. Eventually the increasing power and reliability of digital computers and their

What the heck
is Electronic Mail?

Electronic Mail is a term that's been bandied about data processing circles for years.

Simply put, it means high speed information transportation.

One of the most advanced methods is terminals talking to one another.

Your mailbox is the terminal on your desk. Punch a key and today's correspondence and messages are displayed instantly.

Need to notify people immediately of a fast-breaking development? Have your messages delivered to their terminal mailboxes electronically, across the hall or around the world.

Electronic Mail is document distribution that's more timely, accurate and flexible than traditional methods.

There's no mountain of paperwork.

Administrative personnel are more effective.

Managers have access to more up-to-date information. Decision-making is easier.

Tomorrow's automated office will clearly include Electronic Mail. But like the rest of the Office of the Future, it's available at Honeywell today.

For more information call Mr. Laurie Reeves at (800) 225-3222/3 (within the 617 area, call 552-2048). Or write him at Honeywell Office Automation Systems, Three Newton Executive Park Drive, Newton Lower Falls, Massachusetts 02162.

Honeywell

Someday, in the comfort of your home, you'll be able to shop and bank electronically, read instantly updated newswires, analyze the performance of a stock that interests you, send electronic mail across the country, then play Bridge with three strangers in LA, Chicago and Dallas.
Welcome to someday.

Someday is today! with the CompuServe Information Service. CompuServe is available through a local phone call in most major U.S. cities. It connects almost any brand or type of personal computer or terminal with our big mainframe computers and data bases. All you need to get started is an inexpensive telephone coupler and easy to use software.

CompuServe's basic service costs only $5.00 per hour, billed in minute increments to your charge card.

The CompuServe Information Service is available at many computer stores across the country. Check with your favorite computer center or contact CompuServe.

Welcome to someday.

CompuServe

falling cost squeezed analog computers out of the niches, such as paint mixing, in which they had previously been preferred. Most analog computer suppliers left the industry, although Hewlett-Packard made a strikingly successful transition to the digital world. By the 1970s it was generally no longer necessary to prefix computer with "digital" and consequently the word was less frequently encountered in computing circles.

"Digital" acquired a new resonance from 1993, with the launch of the instantly fashionable *Wired* magazine. In the first issue of *Wired* its editor proclaimed the "the Digital Revolution is whipping through our lives like a Bengali typhoon," just as enthusiasm was building for the information superhighway and the Internet was being opened to commercial use. *Wired* published lists of the "Digerati"—a short-lived coinage conservative activist and prophet of unlimited bandwidth George Gilder used to justify something akin to *People*'s list of the sexiest people alive as judged on intellectual appeal to libertarian techno geeks. The magazine's title evoked both electronic circuits and drug-heightened fervor. As Fred Turner showed in his book *From Counter Culture to Cyberculture*, *Wired* was one in a series of bold projects created by a shifting group of collaborators orbiting libertarian visionary Steward Brand.[8] Brand had previously created the *Whole Earth Catalog* back in the 1960s and a pioneering online community known as the WELL (Whole Earth 'Lectronic Link) in the 1980s. His circle saw technology as a potentially revolutionary force for personal empowerment and social transformation. In the early 1990s this held together an unlikely alliance, from Newt Gingrich who as House Speaker suggested giving laptops to the poor rather than welfare payments, to the futurist Alvin Toffler, U.S. Vice President Al Gore who championed government support for high-speed networking, and Grateful Dead lyricist John Perry Barlow who had founded the Electronic Frontier Foundation to make sure that the new territory of "cyberspace" was not burdened by government interference.

One of the magazine's key figures, Nicholas Negroponte, was particularly important in promoting the idea of "the digital." Negroponte was the entrepreneurial founder and head of MIT's Media Lab, a prominent figure in the world of technology whose fame owed much to a book written by Brand. Negroponte took "digital" far beyond its literal meaning to make it, as the title of his 1995 book *Being Digital*, suggested, the defining characteristic of a new way of life. This was classic rupture talk. His central claim was that in the past things "made of atoms" had been all important. In the future everything that mattered would be "made of bits."

As I argued in a previous column, all information has an underlying material nature.[3] Still, the focus on digital machine-readable representation made some sense: the computer is an exceptionally flexible technology whose applications gradually expanded from scientific calculation to business administration and industrial control to communication to personal entertainment as their speed has risen and their cost fallen. Each new application meant representing a new aspect of the world in machine-readable form. Likewise, the workability of modern computers depended on advances in digital electronics and conceptual developments in coding techniques and information theory. So stressing the digital nature of computer technology is more revealing than calling the computer an "information machine."

Here is a taste of *Being Digital*: "Early in the next millennium, your left and right cuff links or earrings may communicate with each other by low-orbiting satellites and have more computer power than your present PC. Your telephone won't ring indiscriminately; it will receive, sort, and perhaps respond to your calls like a well-trained English butler. Mass media will be refined by systems for transmitting and receiving personalized information and entertainment. Schools will change to become more like museums and playgrounds for children to assemble ideas and socialize with children all over the world. The digital planet will look and feel like the head of a pin. As we interconnect ourselves, many of the values of a nation-state will give way to those of both larger and smaller communities. We will socialize in digital neighborhoods in which physical space will be irrelevant and time will play a different role. Twenty years from now, when you look out of a window what you see may be five thousand miles and six time zones away…"

A wave of enthusiasm for "the digital" has swept through humanities departments worldwide.

Like any expert set of predictions this cluster of promises extrapolated social and technology change to yield a mix of the fancifully bold, the spot-on, and the overly conservative. Our phones do support call screening, although voice communication seems to be dwindling. Online communities have contributed to increased cultural and political polarization. Netflix, Twitter, blogs, and YouTube have done more than "refine" mass media.

As for those satellite cuff links, well the "Internet of Things" remains a futuristic vision more than a daily reality. As the career of the "cashless society" since the 1960s has shown, an imaginaire can remain futuristic and exciting for decades without ever actually arriving.[b] However, when the cuff links of the future do feel the need to communicate they seem more likely to chat over local mesh networks than precious satellite bandwidth. This prediction was perhaps an example of the role of future visions in promoting the interests of the visionary. Negroponte was then on the board of Motorola, which poured billions of dollars into the Iridium network of low-earth orbit satellites for phone and pager communication. That business collapsed within months of launch in 1998 and plans to burn up the satellites to avoid leaving space junk were canceled only after the U.S. defense department stepped in to fund their continued operation.

[b] A phenomenon I explore in more detail in B. Batiz-Lazo, T. Haigh, and D. Steans, "How the Future Shaped the Past: The Case of the Cashless Society," *Enterprise and Society, 36*, 1 (Mar. 2014), 4–17.

Eroding the Future

Of course we never quite got to the digital future. My unmistakably analog windows show me what is immediately outside my house. Whether utopian or totalitarian, imagined future worlds tend to depict societies in which every aspect of life has changed around a particular new technology, or everyone dresses in a particular way, or everyone has adopted a particular practice. But in reality as new technologies are assimilated into our daily routines they stop feeling like contact with an unfamiliar future and start seeming like familiar objects with their own special character. If a colleague reported that she had just ventured into cyberspace after booking a hotel online or was considering taking a drive on the information superhighway to send email you would question her sincerity, if not her sanity. These metaphors served to bundle together different uses of information technology into a single metaphor and distance them from our humdrum lives. Today, we recognize that making a voice or video call, sending a tweet, reading a Web page, or streaming a movie are distinct activities with different meanings in our lives even when achieved using the same digital device.

Sociologist Bruno Latour, a giant in the field of science studies, captured this idea in the title of his 1993 book *We Have Never Been Modern*, published just as Negroponte began to write his columns for *Wired*. Its thesis was that nature, technology, and society have never truly been separable despite the Enlightenment and Scientific Revolution in which their separation was defined as the hallmark of modernity. Self-proclaimed "moderns" have insisted vocally on these separations while in reality hybridizing them into complex socio-technical systems. Thus, he asserts "Nobody has ever been modern. Modernity has never begun. There has never been a modern world."[6]

Latour believed that "moderns," like Negroponte, see technology as something external to society yet also as something powerful enough to define epochs of human existence. As Latour wrote, "the history of the moderns will be punctuated owing to the emergence of the nonhuman—the Pythagorean theorem, heliocentrism...the atomic bomb, the computer.... People are going to distinguish the time 'BC' and 'AC' with respect to computers as they do the years 'before Christ' and 'after Christ'."

He observed that rhetoric of revolution has great power to shape history, writing that "revolutions attempt to abolish the past but they cannot do so..." Thus we must be careful not to endorse the assumption of a historical rupture as part of our own conceptual framework. "If there is one thing we are incapable of carrying out," Latour asserted, "it is a revolution, whether it be in science, technology, politics, or philosophy...."

Our world is inescapably messy, a constant mix of old and new in every area of culture and technology. In one passage Latour brought things down to earth by discussing his home repair toolkit: "I may use an electric drill, but I also use a hammer. The former is 35 years old, the latter hundreds of thousands. Will you see me as a DIY expert 'of contrasts' because I mix up gestures from different times? Would I be an ethnographic curiosity? On the contrary: show me an activity that is homogenous from the viewpoint of the modern time."

According to science fiction writer William Gibson, "The future is already here—it's just not very evenly distributed."[c] That brings me comfort as a historian because of its logical corollary, that the past is also mixed up all around us and will remain so.[d] Even Negroponte acknowledged the uneven nature of change. Back in 1997, in his last column for *Wired*, he noted that "digital" was destined for banality and ubiquity as "Its literal form, the technology, is already beginning to be taken for granted, and its connotation will become tomorrow's commercial and cultural compost for new ideas. Like air and drinking water, being digital will be noticed only by its absence, not its presence."[7]

Digital Humanities

Even after once-unfamiliar technologies dissolve into our daily experience, rupture talk and metaphors of revolution can continue to lurk in odd and unpredictable places. While we no longer think of the Internet as a place called "cyberspace" the military-industrial complex seems to have settled on "cyber warfare" as the appropriate name for online sabotage. Likewise, the NSF has put its money behind the idea of "cyberinfrastructure." The ghastly practice of prefixing things with an "e" has faded in most realms, but "e-commerce" is hanging on. Like most other library schools with hopes of continued relevance my own institution has dubbed itself an "iSchool," copying the names of Apple's successful consumer products. There does not seem to be any particular logic behind this set of prefixes and we might all just as well have settled on "iWarfare," "cybercommerce" and "e-school." But these terms will live on, vestiges of the crisp future vision that destroyed itself by messily and incompletely coming true.

The dated neologism I have been hearing more and more lately is "the digital humanities." When I first heard someone describe himself as a "digital historian" the idea that this would be the best way to describe a historian who had built a website seemed both pretentious and oddly outdated. Since then, however, a wave of enthusiasm for "the digital" has swept through humanities departments nationwide.

According to Matthew Kirschenbaum, the term "digital humanities" was first devised at the University of Virginia back in 2001 as the name for a mooted graduate degree program. Those who came up with it wanted something more exciting than "humanities computing" and broader than "digital media," two established alternatives. It spread widely through the *Blackwell Companion to*

[c] The sentiment is Gibson's, although there is no record of him using those specific words until after they had become an aphorism. See http://quoteinvestigator.com/2012/01/24/future-has-arrived/.

[d] Gibson himself appreciates this, as I have discussed elsewhere T. Haigh, "Technology's Other Storytellers: Science Fiction as History of Technology," in *Science Fiction and Computing: Essays on Interlinked Domains*, D.L. Ferro and E.G. Swedin, Eds., McFarland, Jefferson, N.C., 2011, 13–37

the Digital Humanities issued in 2004. As Kirschenbaum noted, the reasons behind the term's spread have "primarily to do with marketing and uptake" and it is "wielded instrumentally" by those seeking to further their own careers and intellectual agendas. In this humanists are not so different from Worthington back in the 1950s, or Negroponte and his fellow "digerati" in the 1990s, though it is a little incongruous that they appropriated "the digital" just as he was growing tired of it.

The digital humanities movement is a push to apply the tools and methods of computing to the subject matter of the humanities. I can see why young humanists trained in disciplines troubled by falling student numbers, a perceived loss of relevance, and the sometimes alienating hangover of postmodernism might find something liberating and empowering in the tangible satisfaction of making a machine do something. Self-proclaimed digital humanists have appreciably less terrible prospects for employment and grant funding as a humanist than the fusty analog variety. As Marge Simpson wisely cautioned, "don't make fun of grad students. They just made a terrible life choice."

It is not clear exactly what makes a humanist digital. My sense is the boundary shifts over time, as one would have to be using computers to do something that most of one's colleagues did not know how to do. Using email or a word processing program would not qualify, and having a homepage will no longer cut it. Installing a Web content management system would probably still do it, and anything involving programming or scripting definitely would. In fact, digital humanists have themselves been arguing over whether a humanist has to code to be digital, or if writing and thinking about technology would be enough. This has been framed by some as a dispute between the virtuous modern impulse to "hack" and the ineffectual traditional humanities practice of "yack."

As someone who made a deliberate (and economically rather perverse) choice to shift from computer science to the history of technology after earning my first masters' degree, I find this glorification of technological tools a little disturbing. What attracted me to the humanities in the first place was the promise of an intellectual place where one could understand technology in a broader social and historical context, stepping back from the culture of computer enthusiasm that valued coding over contemplating and technological means over human ends.

There is a sense in which historians of information technology work at the intersection of computing and the humanities. Certainly we have attempted, with rather less success, to interest humanists in computing as an area of study. Yet our aim is, in a sense, the opposite of the digital humanists: we seek to apply the tools and methods of the humanities to the subject of computing (a goal shared with newer fields such as "platform studies" and "critical code studies"). The humanities, with their broad intellectual perspective and critical sensibility, can help us see beyond the latest fads and think more deeply about the role of technology in the modern world. Social historians have done a great job examining the history of ideas like "freedom" and "progress," which have been claimed and shaped in different ways by different groups over time. In the history of the past 60 years ideas like "information" and "digital" have been similarly powerful, and deserve similar scrutiny. If I was a "digital historian," whose own professional identity and career prospects came from evangelizing for "the digital," could I still do that work?

There are many ways in which new software tools can contribute to teaching, research, and dissemination across disciplines, but my suspicion is that the allure of "digital humanist" as an identity will fade over time. It encompasses every area of computer use (from text mining to 3D world building) over every humanities discipline (from literary theory to classics). I can see users of the same tools in different disciplines finding an enduring connection, and likewise users of different tools in the same discipline. But the tools most useful to a particular discipline, for example the manipulation of large text databases by historians, will surely become part of the familiar scholarly tool set just as checking a bank balance online no longer feels like a trip into cyberspace. Then we will recognize, to adapt the words of Latour, that nobody has ever been digital and there has never been a digital world. Or, for that matter, a digital humanist.

Further Reading

Gold, M.K., Ed.
***Debates in the Digital Humanities*, University of Minnesota Press, 2012. Also at http://dhdebates.gc.cuny.edu/.** Broad coverage of the digital humanities movement, including its history, the "hack vs. yack" debate, and discussion of the tension between technological enthusiasm and critical thinking.

Gibson, W.
***Distrust that Particular Flavor*, Putnam, 2012.** A collection of Gibson's essays and nonfiction, including his thoughts on our obsession with the future.

Latour, B.
***Science in Action: How to Follow Scientists and Engineers through Society.* Harvard University Press, 1987 and B. Latour and S. Woolgar, *Laboratory Life: The Construction of Scientific Facts.* Princeton University Press, 1986.** *We Have Never Been Modern* is not the gentlest introduction to Latour, so I suggest starting with one of these clearly written and provocative studies of the social practices of technoscience.

Marvin, C.
***When Old Technologies Were New: Thinking About Electric Communication in the Late Nineteenth Century.* Oxford University Press, 1988.** The hopes and fears attributed to telephones and electrical light when they were new provide a startlingly close parallel with the more recent discourse around computer technology.

Morozov, E.
***To Save Everything, Click Here*, Perseus, 2013.** A "digital heretic" argues with zest against the idea of the Internet as a coherent thing marking a rupture with the past.

Winner, L.
***The Whale and the Reactor: A Search for Limits in an Age of High Technology.* University of Chicago Press, 1986.** A classic work in the philosophy of technology, including a chapter "Mythinformation" probing the concept of the "computer revolution."

References

1. Berkeley, E.C. *Giant Brains or Machines That Think.* Wiley, NY, 1949.
2. Flichy, P. *The Internet Imaginaire.* MIT Press, Cambridge, MA, 2007.
3. Haigh, T. Software and souls; Programs and packages. *Commun. ACM 56*, 9 (Sept. 2013), 31–34.
4. Hecht, G. Rupture-talk in the nuclear age: Conjugating colonial power in Africa. *Social Studies of Science 32*, 6 (Dec. 2002).
5. Kling, R. Learning about information technologies and social change: The contribution of social informatics. *The Information Society 16*, 3 (July–Sept. 2000), 217–232.
6. Latour, B. *We Have Never Been Modern.* Harvard University Press, Cambridge, MA, 1993.
7. Negroponte, N. Beyond digital. *Wired 6*, 12 (Dec. 1998).
8. Turner, F. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism.* University of Chicago Press, Chicago, 2006.

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

DOI:10.1145/2644230

Peter J. Denning

The Profession of IT
Learning for the New Digital Age

Digital machines are automating knowledge work at an accelerating pace. How shall we learn and stay relevant?

THE FIRST OF the two accompanying images shows the IBM Blue Gene supercomputer at Argonne Labs. It houses 250,000 processors in 72 cabinets connected by an optical network. It can perform approximately 10^{15} operations per second—a million times faster than the chip in your smartphone. The image on the next page is a beautiful graph of connections between Internet sites. The Internet is a supercomputer grown from a billion machines and several billion people.

The IBM supercomputer is a wholly electronic machine. It uses a robust design first conceived in the 1940s. Its structure is fixed. It is very good at processing large datasets with deterministic algorithms. It has no intelligence.

The Internet is an organic system with humans and machines in a never-ending dance of interaction amplifying each other's capabilities. It is constantly changing its structure and some of the changes are disruptive. It is nondeterministic because there is no way to predict how interactions among so many people and machines will turn out. It has intelligence—the collected, amplified, collaborative intelligence of everyone who participates in it.

The Internet organism is not replacing the machine. It is a new system built on machines, mobile devices, their connections, and their interaction with humans. The network of machines is the infrastructure of the organic system.

IBM Blue Gene supercomputer.

PHOTOGRAPH COURTESY OF ARGONNE NATIONAL LABORATORY

The two images represent not only developmental stages of computing, but also different approaches to understanding the world. The machine view represents the advancement of science, which seems poised to know all data, predict what will happen, and enable stabilizing controls over social and economic systems. The organism view exposes an unruly, ever-evolving world, rife in uncertainties, unpredictable events, and disruptions. Our attitudes toward learning new things and staying professionally relevant belong to the age of the machine. What do we need to stay professionally relevant in the age of organism?

The New Machine Age

Erik Brynjolfsson and Andrew McAfee refer to the new, organic era as the second machine age.[1] Computers are automating knowledge work just as engines began automating manual work two centuries ago.

Historians put the beginning of the previous machine age at the invention of the steam engine by James Watt around 1781. Many say engines were the moving force behind the Industrial Revolution. By automating manual work, engines increased productivity and opened new markets for factory-manufactured goods, creating widespread prosperity and many new jobs. By the early 1900s, machine-based factory operations were so well understood that Frederick Taylor, a mechanical engineer, formulated the famous theory of scientific management, which allowed many factories and businesses to be optimized for maximal production.

In the 1960s, management guru Peter Drucker coined the term "knowledge worker" for workers who depended on their minds rather than manual labor to produce value. Until that time, much knowledge work had been spared the ravages of automation. Because knowledge work was an ever-increasing portion of the labor market, Drucker considered knowledge worker productivity to be the next frontier of management. He foresaw that computers would play an increasing role in automating knowledge work.

IBM, founded in 1924, was one of the first companies to apply computing machines to business data processing, an early form of knowledge work. IBM's machines sorted and arranged punched cards representing inventories and customers. They enabled the growth of large corporations with worldwide operations. When it introduced the first hard disk, RAMAC, in 1956, IBM said the new machine could condense an entire warehouse of file cabinets onto a single disk and automate all the clerical functions of storing, retrieving, and analyzing files.

Since that time, the computing power of chips, storage, and networks has grown exponentially. We are digitizing everything and can manipulate the digital representations of almost anything. We can connect with almost anybody. These factors foster new waves of innovation. An example of a wave is the world of "apps"—small software programs with tightly focused functions that customize our mobile devices into personalized extensions of ourselves. App development is unlike any previous manufacturing world. New apps can be produced cheaply, copied any number of times perfectly, distributed at near-zero cost, and sold at low prices. Network-connected apps deliver amazing new capabilities that were considered impossible just 10 years ago. Apps craft our personal spaces, see and manipulate our environments, interface to large systems, and even lift our spirits and our hearts. The Apple and Android stores offer over 700,000 apps.

Automation and Jobs

Brynjolfsson and McAfee argue that digital machines are displacing less productive workers in a large number of job categories. In the Great Recession beginning in 2008, employers downsized labor forces and automated functions previously done by human workers; as the recession slowly ended, they did not rehire many displaced workers. Job growth has since been slow because the displaced workers do not have the new skill sets needed in the rapidly growing digital high technology sectors. There are few education programs to help them make the transition.

Disparity of wealth increases in the wake of digital automation. The relatively few designers and engineers who build the new technologies are well paid in salaries and in capital gains (for example, stock options); their technologies can displace relatively many workers. Although wealth spreads are a normal feature of technology expansion, they are worrisome when they get large enough to create social tensions. Examples are the resentments of the 1% who hold 99% of wealth, the gap between CEO and worker pay, and of soaring rents in San Francisco driven by the influx of high-tech industry workers and executives.

Automation may not be the only factor impeding job growth. John Dearie and Courtney Geduldig argue that, in the U.S. at least, almost all *new* jobs are created by businesses less than five years old, but recent restrictions from government regulations and tax structures have caused a sharp drop in new-company formation.[2] They propose regulatory and tax reforms that exempt new businesses from much of the regulatory and tax burden, allowing entrepreneurs access to the capital they need to get started. Human capital expert Edward Gordon has similar proposals and strongly endorses education as a way to help overcome persistent joblessness.[4]

Whatever the correct explanation, almost everyone agrees that education is a powerful means to increase mobility of workers and reduce the spread between high and low incomes.

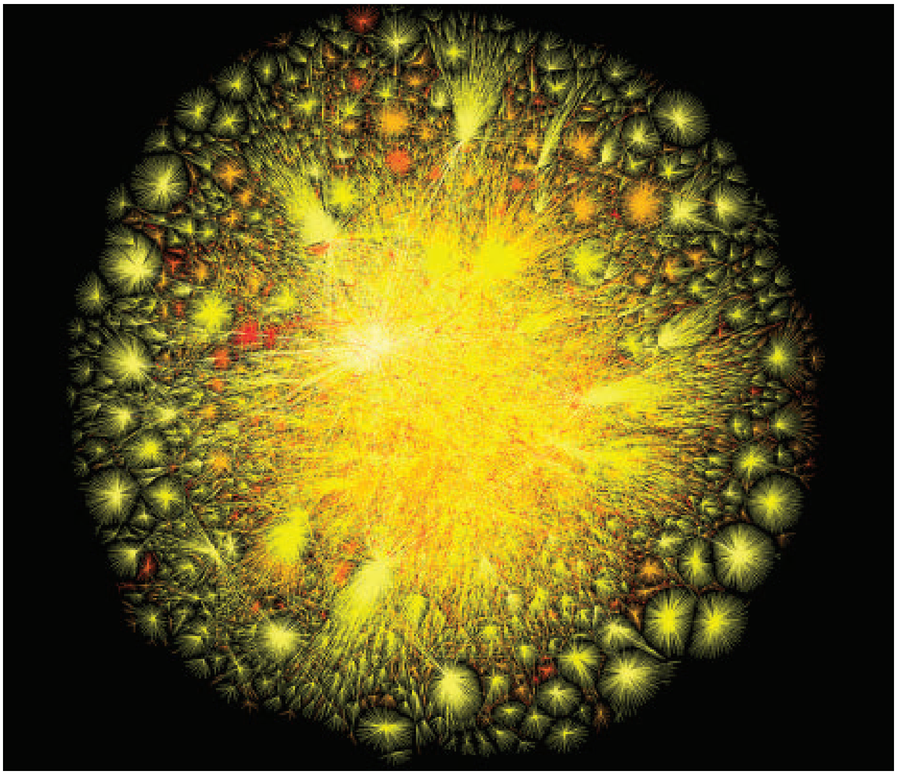
Still, Brynjolfsson and McAffee are optimistic we can design new education programs that help workers move into new areas. They believe the process of automation displacing workers is likely to be gradual and that the secret to success will be to show people how to generate new value from the machines—to race "with" the machines instead of "against" them.

What to Learn?

Two excellent analyses of what education can do to help professionals keep up with accelerating changes in the job markets can be found in the Chile report *Surfing Towards the Future*[3] and in the book *A New Culture of Learning* by Douglas Thomas and John Seely Brown.[5]

Both books observe that our current school systems are structured around assumptions that are becoming less and less tenable. The bodies of knowledge of many technical fields are changing more rapidly than curricula; a degree may be obsolete before its holder graduates. A graduation certificate is less valuable than demonstrated ability to perform, adapt to accelerating changes, and mobilize networks to get jobs done. The Chile report discusses three dimensions of education for this world.

Pragmatics. This has two aspects.



An Internet graph rendered using Border Gateway Protocol data points circa 2010.

IMAGE COURTESY OF BARRETT LYON/OPTE.ORG

One is the learning of skills for competent performance in your domain or profession. The other is learning social interaction skills in communication, coordination, identity, listening, trust, making history, cultivating pluralistic networks, and more. As a professional, you need both kinds of skills to function well.

Coping with accelerating change. Changes in our social systems always happen as historical emergences—responses to events and concerns at a particular time. Professionals need the skills of designers and entrepreneurs to lead and manage the changes, to deal with the moods and emotions of those affected by the changes, and to deal with the economic and social consequences of changes. Professionals also need to be able to cope with disruptions and help others to cope. Disruptions can be large, for example when societies or businesses undergo sharp change; and they can be highly personal, such as a death or illness in a family or loss of a job. They can be relatively slow declines, such as the obsolescence of an industry, rather than cataclysms. Whether large or small, if a disruptive event hits you, you may suddenly find that your skills and practices are no longer of value. The source of your professional identity is gone. You must learn how to detach, let go, and invent new offers for the world that you now find yourself in. This is very difficult but it is essential.

Mentoring. Mentoring is the central skill powering learning networks. It means more-experienced people help less-experienced people learn new skills and form new networks and communities. It means mobilizing networks to get particular jobs done. It means cultivating design and entrepreneurial dispositions such as listening, proposing, experimenting, questing, inquiring, reflecting, and caring for future generations and the planet.

These three dimensions do not exist in the current formal educational system. A new kind of education system is likely to form, whose professionals are skilled mentors. They will integrate many traditional technology fields, such as engineering and computing, with development fields including somatics, coaching, language-action, and social networking. However, you cannot wait for someone to do this for you. You need to find existing communities in the Internet that enable you to learn in these dimensions.

Our attitudes toward learning new things and staying professionally relevant belong to the age of the machine.

Thomas and Seely Brown argue that you can access this kind of learning through numerous learning communities already existing in the Internet.[5] In a learning network the participants become immersed in conversations and actions together. There are tasks for members at all levels of skill, and the more experienced mentor the less experienced. For example, the Faulkes Telescope project (http://www.faulkes-telescope.com/) allows middle and high school students to schedule time on a robotic telescope, take pictures, and process the pictures with a library of image software. They interact with professional astronomers as they share, evaluate, and interpret their pictures.

Learning networks are ubiquitous. People with illnesses share their experiences about their therapies and support each other. Technical support discussion groups are far better at troubleshooting modern computers and networks than manuals and technical support phones. Crowdsourcing services mobilize many strangers to find solutions of problems posted by sponsors, and pay the winners for their time. Even online games such as World of Warcraft have been cited in business journals as learning networks for certain kinds of leadership skills.

Learning networks differ in at least six ways from traditional schools:

- There is no authority figure such as the "teacher"; anyone can be a mentor if they know something, can listen, and motivate.
- Participants can be involved at different levels: passive observing (hanging out), scattered experimentation (messing around), and serious projects (geeking).
- Some learning communities cultivate tolerance of pluralistic views and willingness to cooperate with people of different cultures.
- The network encourages experimenting.
- The network cultivates important dispositions such as questing and inquiry.
- Networks can operate as loose collectives or as tight-knit communities.

Over time, there will be more of these networks, they will be easier to find, and they may eventually converge with formal schools. Learning networks are excellent at coping with accelerating change and, in a growing number of cases, they are leading change.

What Can You Do?

If you are reading this, you are most likely a member of the computing field and are less likely to be one of the displaced workers. I hope you will reach out to people in other fields, who may be in the process of being displaced by your work, and help them find education programs and learning networks so that they may find new work. You also need to pay attention to your own professional health, especially if you are in an area where automation is possible. Find ways to learn social interaction pragmatics, coping with changes, and mentoring. Constantly focus not on fighting machines but using them to to increase your own productivity and value. Find and participate in learning networks on topics of interest or concern to you. C

References

1. Brynjolfsson, E. and McAfee, A. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies.* W.W. Norton, 2014.
2. Dearie, J. and Geduldig, C. *Where the Jobs Are: Entrepreneurship and the Soul of the American Economy.* Wiley, NY, 2013.
3. Flores, F. Report of Consejo Nacional de Innovación para la Competitividad, *Surfing Towards the Future: Chile on the 2025 Horizon.* 2013; http://chile-california.org/surfing-towards-the-future-into-counselor-fernando-floress-vision-for-innovation.
4. Gordon, E. *Future Jobs: Solving the Employment and Skills Crisis.* Praeger, 2013.
5. Thomas, D. and Seely Brown, J. *A New Culture of Learning: Cultivating the Imagination for a World of Constant Change.* CreateSpace, 2011.

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

DOI:10.1145/2644257

Luke Muehlhauser and Bill Hibbard

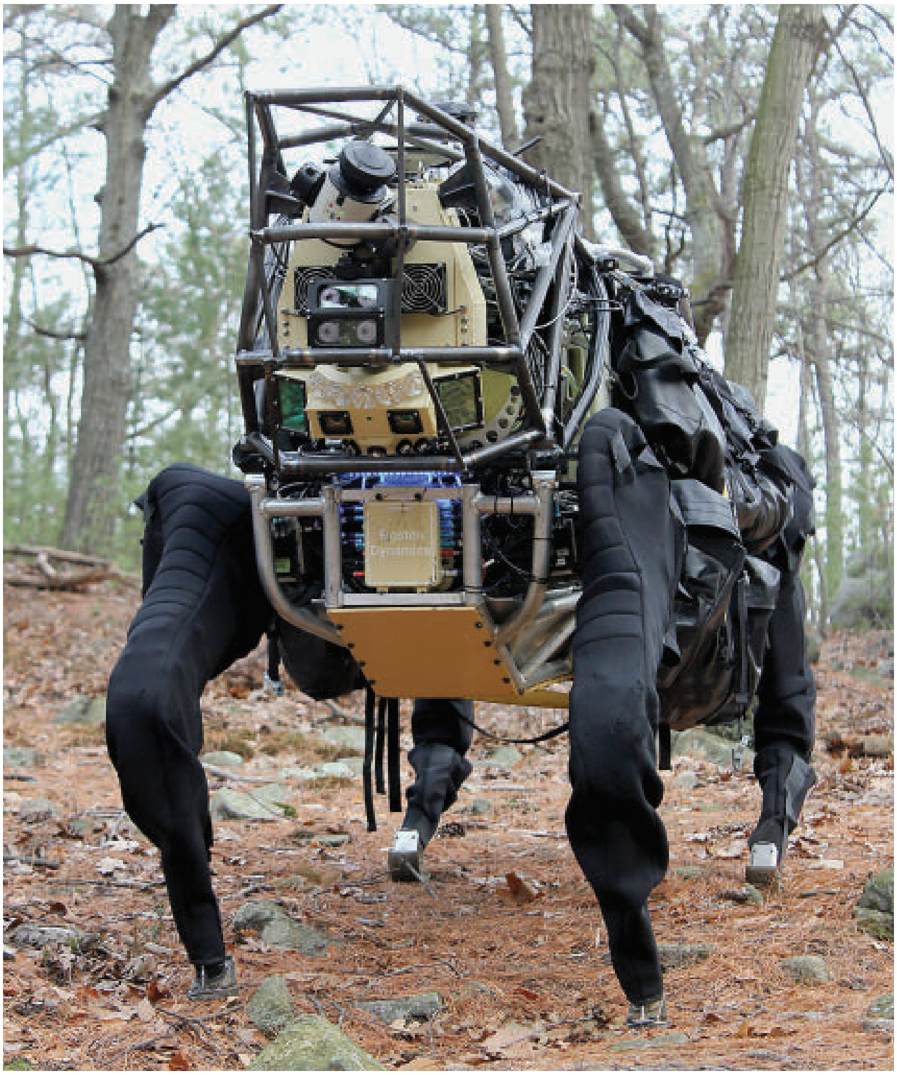
Viewpoint Exploratory Engineering in Artificial Intelligence

Using theoretical models to plan for AI safety.

WE REGULARLY SEE examples of new artificial intelligence (AI) capabilities. Google's self-driving car has safely traversed thousands of miles. IBM's Watson beat the "Jeopardy!" champions, and Deep Blue beat the chess champion. Boston Dynamics' Big Dog can walk over uneven terrain and right itself when it falls over. From many angles, software can recognize faces as well as people can.

As their capabilities improve, AI systems will become increasingly independent of humans. We will be no more able to monitor their decisions than we are now able to check all the math done by today's computers. No doubt such automation will produce tremendous economic value, but will we be able to *trust* these advanced autonomous systems with so much capability?

For example, consider the autonomous trading programs that lost Knight Capital $440 million (pre-tax) on August 1, 2012, requiring the firm to quickly raise $400 million to avoid bankruptcy.[14] This event undermines a common view that AI systems cannot cause much harm because they will only ever be tools of human masters. Autonomous trading programs make millions of trading decisions per day, and they were given sufficient capability to nearly bankrupt one of



PHOTOGRAPH COURTESY OF DARPA

DARPA's semi-autonomous Legged Squad Support System (AlphaDog) was developed from a decade of research on perception, autonomy, and mobility.

the largest traders in U.S. equities.

Today, AI safety engineering mostly consists in a combination of formal methods and testing. Though powerful, these methods lack foresight: they can be applied only to particular extant systems. We describe a third, complementary approach that aims to predict the (potentially hazardous) properties and behaviors of broad classes of future AI agents, based on their mathematical structure (for example, reinforcement learning). Such projects hope to discover methods "for determining whether the behavior of learning agents [will remain] within the bounds of prespecified constraints... after learning."[7] We call this approach "exploratory engineering in AI."

As their capabilities improve, artificial intelligence systems will become increasingly independent of humans.

Exploratory Engineering in Physics, Astronautics, Computing, and AI

In 1959, Richard Feynman pointed out that the laws of physics (as we understand them) straightforwardly imply that we should be able to "write the entire 24 volumes of the *Encyclopaedia Brittanica* on the head of a pin."[6] Feynman's aim was to describe technological possibilities as constrained not by the laboratory tools of his day but by known physical law, a genre of research Eric Drexler later dubbed "exploratory engineering."[4] Exploratory engineering studies the ultimate limits of yet-to-be-engineered devices, just as theoretical physics studies the ultimate limits of natural systems. Thus, exploratory engineering "can expose otherwise unexpected rewards from pursuing particular research directions [and] thus improve the allocation of scientific resources."[3]

This kind of exploratory engineering in physics led to large investments in nanoscale technologies and the creation of the U.S. National Nanotechnology Initiative. Today, nanoscale technologies have a wide range of practical applications, and in 2007 Israeli scientists printed the entire Hebrew Bible onto an area smaller than the head of a pin.[1]

Nanoscience is hardly the only large-scale example of exploratory engineering. Decades earlier, the scientists of pre-Sputnik astronautics studied the implications of physical law for spaceflight, and their analyses enabled the later construction and launch of the first spacecraft. In the 1930s, Alan Turing described the capabilities and limitations of mechanical computers several years before John von Neumann, Konrad Zuse, and others figured out how to build them. And since the 1980s, quantum computing researchers have been discovering algorithms and error-correction techniques for quantum computers that we cannot yet build—but whose construction is compatible with known physical law.

Pushing the concept of exploratory engineering a bit beyond Drexler's original definition, we apply it to some recent AI research that formally analyzes the implications of some theoretical AI models. These models might not lead to useful designs as was the case in astronautics and nanoscience, but like the theoretical models that Butler Lampson used to identify the "confinement problem" in 1973,[10] these theoretical AI models do bring to light important considerations for AI safety, and thus they "expose otherwise unexpected rewards from pursuing particular research directions" in the field of AI safety engineering. In this Viewpoint, we focus on theoretical AI models inspired by Marcus Hutter's AIXI,[9] an optimal agent model for maximizing an environmental reward signal.

AIXI-like Agents and Exploratory Engineering

How does AIXI work? Just as an idealized chess computer with vast amounts of computing power could brute-force its way to perfect chess play by thinking through the consequences of all possible move combinations, AIXI brute-forces the problem of general intelligence by thinking through the consequences of all possible actions, given all possible ways the universe might be. AIXI uses Solomonoff's universal prior to assign a relative prior probability to every possible (computable) universe, marking simpler hypotheses as more likely. Bayes' Theorem is used to update the likelihood of hypotheses based on observations. To make decisions, AIXI chooses actions that maximize its expected reward. More general variants of AIXI maximize a utility function defined on their observations and actions.

Based on an assumption of a stochastic environment containing an infinite amount of information, the original AIXI model is uncomputable and therefore not a subject of exploratory engineering. Instead, finitely computable variants of AIXI, based on the assumption of a stochastic environment containing a finite amount of information, can be used for exploratory engineering in AI. The results described here do not depend on the assumption of infinite computation.

A Monte-Carlo approximation of AIXI can play Pac-Man and other simple games,[16] but some experts think AIXI approximation is not a fruitful path toward human-level AI. Even if that is true, AIXI is the first model of cross-domain intelligent behavior to be so completely and formally specified that we can use it to make formal arguments about the properties that would obtain in certain classes of hypothetical agents if we could build them today. Moreover, the formality of AIXI-like agents allows researchers to uncover potential safety problems with AI agents of increasingly general capability—problems that could be addressed by additional research, as happened in the field of computer security after Lampson's article on the confinement problem.

AIXI-like agents model a critical property of future AI systems: they will need to explore and learn models of the world. This distinguishes AIXI-like agents from current systems that use predefined world models, or learn parameters of predefined world models. Existing verification techniques for autonomous agents[5] apply only to particular systems, and to avoiding unwanted optima in specific utility functions. In contrast, the problems described here apply to broad classes of agents, such as those that seek to maximize rewards from the environment.

For example, in 2011 Mark Ring and Laurent Orseau analyzed some classes of AIXI-like agents to show that several kinds of advanced agents will maximize their rewards by taking direct control of their input stimuli.[13] To understand what this means, recall the experiments of the 1950s in which rats could push a lever to activate a wire connected to the reward circuitry in their brains. The rats pressed the lever again and again, even to the exclusion of eating. Once the rats were given direct control of the input stimuli to their reward circuitry, they stopped bothering with more indirect ways of stimulating their reward circuitry, such as eating. Some humans also engage in this kind of "wireheading" behavior when they discover they can directly modify the input stimuli to their brain's reward circuitry by consuming addictive narcotics. What Ring and Orseau showed was that some classes of artificial agents will wirehead—that is, they will behave like drug addicts.

Fortunately, there may be some ways to avoid the problem. In their 2011 paper, Ring and Orseau showed that some types of agents will resist wireheading. And in 2012, Bill Hibbard showed[8] the wireheading problem can also be avoided if three conditions are met: the agent has some foreknowledge of a stochastic environment; the agent uses a utility function instead of a reward function; and we define the agent's utility function in terms of its internal mental model of the environment. Hibbard's solution was inspired by thinking about how *humans* solve the wireheading problem: we can stimulate the reward circuitry in our brains with drugs, yet most of us avoid this temptation because our models of the world tell us that drug addiction will change our motives in ways that are bad according to our current preferences.

Will we be able to *trust* these advanced autonomous systems with so much capability?

Relatedly, Daniel Dewey showed[2] that in general, AIXI-like agents will locate and modify the parts of their environment that generate their rewards. For example, an agent dependent on rewards from human users will seek to replace those humans with a mechanism that gives rewards more reliably. As a potential solution to this problem, Dewey proposed a new class of agents called *value learners*, which can be designed to learn and satisfy any initially unknown preferences, so long as the agent's designers provide it with an idea of what constitutes evidence about those preferences.

Practical AI systems are embedded in physical environments, and some experimental systems employ their environments for storing information. Now AIXI-inspired work is creating theoretical models for dissolving the agent-environment boundary used as a simplifying assumption in reinforcement learning and other models, including the original AIXI formulation.[12] When agents' computations must be performed by pieces of the environment, they may be spied on or hacked by other, competing agents. One consequence shown in another paper by Orseau and Ring is that, if the environment can modify the agent's memory, then in some situations even the simplest stochastic agent can outperform the most intelligent possible deterministic agent.[11]

Conclusion

Autonomous intelligent machines have the potential for large impacts on our civilization.[15] Exploratory engineering gives us the capacity to have some foresight into what these impacts might be, by analyzing the properties of agent designs based on their mathematical form. Exploratory engineering also enables us to identify lines of research—such as the study of Dewey's value-learning agents—that may be important for anticipating and avoiding unwanted AI behaviors. This kind of foresight will be increasingly valuable as machine intelligence comes to play an ever-larger role in our world.

References

1. Associated Press. Haifa Technion scientists create world's smallest bible. *Haaretz*, (Dec. 24, 2007); http://www.haaretz.com/news/haifa-technion-scientists-create-world-s-smallest-bible-1.235825
2. Dewey, D. Learning what to value. In *Artificial General Intelligence: 4th International Conference*. J. Schmidhuber et al., Eds. Springer, Berlin, 2011, 309–314.
3. Drexler, K.E. *Nanosystems: Molecular Machinery, Manufacturing, and Computation.* Wiley, NY, 1992, 490.
4. Drexler, K.E. Exploring future technologies. In *Doing Science: The Reality Club*. J. Brockman, Ed., Prentice Hall, NY, 1991, 129–150.
5. Fisher, M. et al. Verifying autonomous systems. *Commun. ACM 58*, 9 (Sept. 2013), 84–93.
6. Feynman, R. There's plenty of room at the bottom. Annual Meeting of the American Physical Society at the California Institute of Technology in Pasadena, CA, (Dec. 29, 1959).
7. Gordon-Spears, D. Assuring the behavior of adaptive agents. In *Agent Technology from a Formal Perspective*. C. Rouff et al., Eds., Springer, Berlin, 2006, 227–259.
8. Hibbard, B. Model-based utility functions. *Journal of Artificial General Intelligence 3*, 1 (2012), 1–24.
9. Hutter, M. One decade of universal artificial intelligence. In *Theoretical Foundations of Artificial General Intelligence*. P. Wang and B. Goertzel, Eds., Atlantis Press, Amsterdam, 2012.
10. Lampson, B. A note on the confinement problem. *Commun. ACM 16*, 10 (Oct. 1973), 613–615.
11. Orseau, L. and Ring, M. Memory issues of intelligent agents. In *Artificial General Intelligence: 5th International Conference*. J. Bach et al., Eds., Springer, Berlin, 2012, 219–231.
12. Orseau, L. and Ring, M. Space-time embedded intelligence. In *Artificial General Intelligence: 5th International Conference*. J. Bach et al., Eds., Springer, Berlin, 2012, 209–218.
13. Ring, M. and Orseau, L. Delusion, survival, and intelligent agents. In *Artificial General Intelligence: 4th International Conference*, J. Schmidhuber et al., Eds., pp. 11–20. Berlin: Springer, 2011, pp. 11–20.
14. Valetkevitch, C. and Mikolajczak, C. Error by Knight Capital rips through stock market. *Reuters* (Aug. 1, 2012).
15. Vardi, M. The consequences of machine intelligence. *The Atlantic* (Oct. 25, 2012); http://www.theatlantic.com/technology/archive/2012/10/the-consequences-of-machine-intelligence/264066/
16. Veness, J. et al. A Monte-Carlo AIXI approximation. *Journal of Artificial Intelligence 40* (2011), 95–142.

Luke Muehlhauser (luke@intelligence.org) is the executive director of the Machine Intelligence Research Institute in Berkeley, CA.

Bill Hibbard (billh@ssec.wisc.edu) is an Emeritus Senior Scientist at the University of Wisconsin-Madison Space Science and Engineering Center and a research associate at the Machine Intelligence Research Institute in Berkeley, CA.

DOI:10.1145/2644279

John Leslie King and Paul F. Uhlir

Viewpoint
Soft Infrastructure Challenges to Scientific Knowledge Discovery

Seeking to overcome nontechnical challenges to the scientific enterprise.

OPEN NETWORK ENVIRONMENTS have become essential in the sciences, enabling accelerated discovery and communication of knowledge. The genomics revolution, for example, involved gene-sequencing machines that accelerated genome mapping. Yet, the real revolution began when open community databases allowed researchers to build on existing contributions and compare their results to established knowledge. Such payoffs have spread to other fields through exploitation of open geographic information systems and open social network data. In another example, "citizen science" engages people not previously involved in science in data collection and analysis via technology-supported, distributed collaboration. Transforming scientific knowledge discovery requires collaboration among specialists in domain sciences (for example, chemistry, geology, physics, molecular biology), the computing sciences, and the human sciences. The very nature of scientific knowledge discovery is changing.

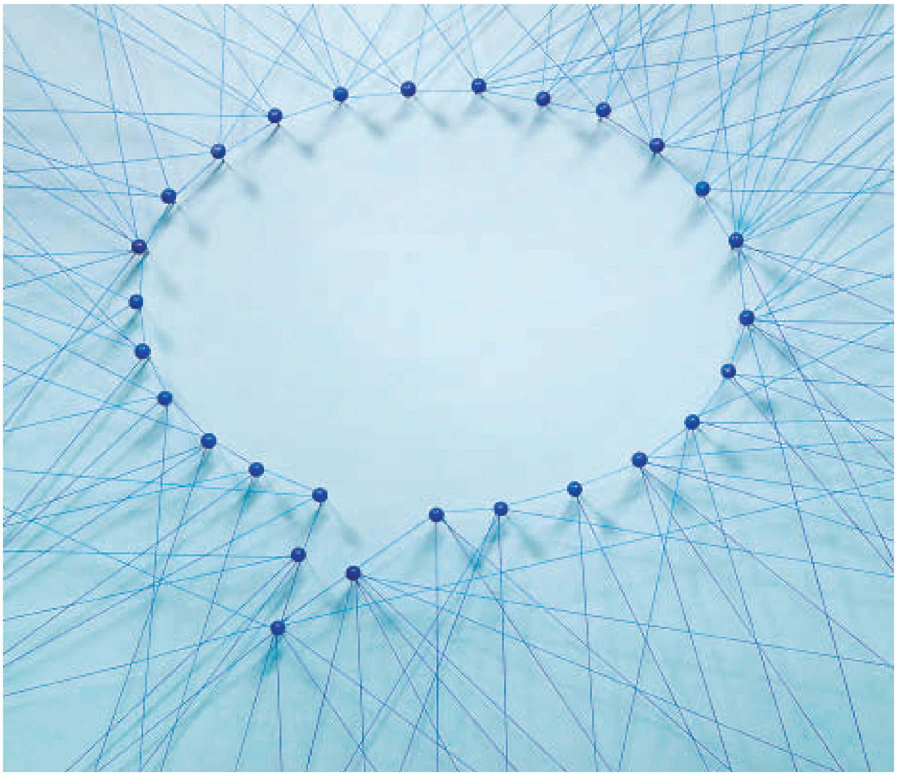


IMAGE BY POGONICI

The computing research community should be interested in these developments. Computing research is affected by these changes, and computing in general plays a special role in the technologies that make such progress possible. The National Research Council's (NRC) Board on Research Data and Information (BRDI) has been working on this topic (see the accompanying sidebar). New techniques and methods can help achieve new benefits at reduced time and cost. However, there are also significant barriers that are nontechnical in nature, which this Viewpoint highlights.

Digital computing technologies (data mining, information retrieval

The National Research Council's Board on Research Data and Information Workshop

A BRDI workshop titled "The Future of Scientific Knowledge Discovery in Open Networked Environments" was held in March 2011. John Leslie King of the University of Michigan was chair of a steering committee that included Hal Abelson of MIT, Francine Berman of RPI, Bonnie Carroll of Information International Associates, Michael Carroll of Washington College of Law, Alyssa Goodman of Harvard University, Sara Graves of the University of Alabama at Huntsville, Michael Lesk of Rutgers University, and Gilbert Omenn of the University of Michigan Medical School. The NRC's Paul F. Uhlir, director of BRDI, was rapporteur for the workshop, aided by additional rapporteurs Alberto Pepe of Harvard-Smithsonian Astronomical Observatory and Puneet Kishor, University of Wisconsin. Additional editorial support was provided by Daniel Cohen of the Library of Congress (on detail to BRDI) and Raed Sharif of Syracuse University. See http://sites.nationalacademies.org/PGA/brdi/PGA_060422 and http://www.nap.edu/catalog.php?record_id=18258 for free reports related to the workshop. A print version of the workshop report can be ordered for a fee from National Academies Press, http://www.nap.edu.

and extraction, artificial intelligence, distributed grid computing, and the like) are important, but they are only part of the cast in the larger play of computer-mediated knowledge discovery. Online availability of data and published papers affect the research life cycle through rapid dissemination of research results and broader participation in research activity.[a] Exploiting such opportunities requires moving beyond technical challenges, which are frequently difficult to overcome in their own right, to the challenges of "soft infrastructure": institutional factors, governance, and cultural inertia that tend to impede payoffs from the rapid evolution of techniques and methods. The promise is great, but it will be realized only if the complementary assets of production, including those of soft infrastructure, are provided.

Soft infrastructure ranges from the psychology of individual action to institutional reward structures and intellectual property conventions. Technology enables improvements and stimulates new thinking, but fitting new technology to existing practice can violate social protocols that have been refined and embedded over centuries. These include the attitudes and practices of researchers, publishers, reviewers, and university promotion committees. These are *part of* the culture of knowledge discovery. They predate technology-enabled, open networked environments, and many of them persist for sensible reasons. Dismissing them as "resistance to change" is dysfunctional.

In fact, the culture of knowledge discovery is open to change. For example, most university promotion committees now accept that some areas of computing research consider refereed conferences more important than refereed journals. This change took effort: it required persuasion and the authoritative efforts of the ACM, the Computing Research Association (CRA), and the National Research Council's Computer Science and Telecommunications Board (CSTB). The change was the "right thing to do," but it took much work and a number of years.

The concerns underlying soft infrastructure are important but often overlooked. Few scientists want to change a reward structure based on results (for example, contributing to scientific knowledge), even though that structure persuades smart researchers against sharing anything before rewards are worked out. Similarly, publishers, universities, and others are reluctant to give up intellectual property rights, and useful research efforts have been derailed by inability to come to agreement on such matters. This is not blind "resistance to change." It is smart, at least in the short run. But is it smart in the long run? If not, how can the culture of knowledge discovery be changed to address this? This challenge is exacerbated by the increasingly global nature of research. Different institutional approaches, languages, norms, and levels of development create challenges that will take time to sort out.

Innovative ways can be found to leverage open networked environments. Scientific knowledge discovery can be improved through support of science-funding agencies, research universities, and science and engineering professionals. To be effective, however, these efforts require attention to what economists call *complementary assets*: the full set of elements required to gain hoped-for benefits. An example is data curation, preparing and maintaining data with potential for reuse. This involves deciding which data will be kept, how the data are described (for example, through metadata), how quality control will be maintained, and how coding schemes and analytical tools that enable reuse will be provided. Whose job is data curation? Researchers frequently lack the skills and inclination to take on this work, and assume others such as academic librarians will do so. Yet where academic library resources are strained, librarians cannot take on extra work.

Serious power issues can arise from such challenges. Researchers that object to anyone but themselves controlling aspects of scientific work they see as essential might refuse to take on additional work they see as unessential. To continue the example, researchers and academic librarians have cooperated because of conventions created over decades and grounded in a different era. If researchers insist that academic librarians take on additional work the librarians cannot afford, a power conflict can emerge. It is difficult to change an equilibrium that has worked well in order to achieve new benefits of scientific knowledge in open networked environments. Citizen science raises

[a] An Internet search for "research life cycle" images (including the quotes) yields more than 170,000 links as of July 2014. Research is a set of activities that take place over time, sometimes presided over by different people. Open networked science can change many aspects of the research life cycle.

similar concerns. Citizen science is growing: the Cornell Lab for Ornithology's eBird project and Galaxy Zoo in astronomy and are but two examples, each involving tens of thousands if not hundreds of thousands of people who have never been socialized into research work. Such people may make unconventional demands if they feel they are not properly compensated for their important efforts. Such power conflicts do not arise from open networked environments, *per se*, but from the new opportunities enabled by such environments in circumstances of constrained resources.

It is common when confronting challenges involving rewards, power, and conflict to enlist the economic, social, and behavioral sciences—what some refer to as the "human" sciences. This effort is similar to that involving the computing sciences over the past three decades, and lessons from that experience are germane to the current situation. "Enlistment" is a telling notion: the human sciences are no more willing to be ordered to do such work than computing sciences were before them. Neither wishes to offer poorly formulated solutions to poorly understood problems. Neither sees its community as "hired guns" whose only purpose is to fix the problems faced by other scientists. Both are interested in making progress in their respective fields, and only when that progress is addressed are they willing to discuss work that *also* benefits other sciences. Edicts requiring interdisciplinary work involving people from various sciences seldom persuade the *best* scientists from various fields to collaborate. The human sciences are needed for the promise of scientific knowledge discovery in open networked environments, just as the computing sciences have been. In both cases, the art of effective "deal making" is still evolving.

Scientific knowledge discovery has become important. Important things become political.

A final complicating factor is the changing value proposition of research. During much of the 20th century the U.S. research enterprise capitalized on the benefits of scientific agriculture, advanced the industrial revolution, improved human health, and helped achieve victory in conflicts such as World War II and the Cold War. Knowledge discovery was a public good, and more was better. Now, politicians and policymakers acknowledge the value of scientific knowledge discovery, but at the same time ask how much is needed, at what price, paid for by whom, and benefiting whom? Scientific knowledge discovery has become important. Important things become political. The political salience of science is unlikely to translate into a blank check for scientists to spend as they choose. If anything, science is increasingly subject to calls for cost-benefit analyses in a pluralistic political environment where a shared *value proposition* (needed for coherent cost-benefit outcomes) is increasingly difficult to establish. Influential people who value open scientific knowledge discovery also invoke practical considerations such as economic growth, national security, health improvements, and other societal goals. Open networked environments can enhance scientific knowledge discovery, but the political tensions regarding science are likely to grow.

In short, open knowledge discovery can improve the scientific enterprise. One can think of this as a mandate created by technological progress, particularly in the digital realm. Nevertheless, the success of research, broadly considered, depends on the appropriate management of the underlying soft infrastructure.

John Leslie King (jlking@umich.edu) is W.W. Bishop Professor in the School of Information at the University of Michigan, Ann Arbor, MI.

Paul F. Uhlir (PUhlir@nas.edu) is the director of the Board on Research Data and Information (BRDI) at the U.S. National Academies in Washington, D.C.

The authors thank J Strother Moore and two anonymous reviewers for their help with this Viewpoint. The views expressed here are those of the authors and should not necessarily be ascribed to their employers.

Calendar of Events

September 15–16
Generative Programming: Concepts of Experiences,
Vasteras, Sweden,
Sponsored: SIGPLAN,
Contact: Ulrik Pagh Schultz,
Email: ups@mmmi.sdu.dk

September 15–19
ACM/IEEE International Conference on Automated Software Engineering,
Vasteras, Sweden,
Sponsored: SIGAI, SIGSOFT,
Contact: Ivica Crnkovic,
Email: ivica.crnkovic@mdh.se

September 16–19
ACM Symposium on Document Engineering,
Fort Collins, CO,
Sponsored: SIGWEB,
Contact: Steven Simske,
Email: steven.simske@hp.com

September 18–19
2014 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement,
Torino, Italy,
Sponsored: SIGSOFT,
Contact: Maurizio Morisio,
Email: maurizio.morisio@polito.it

September 21–26
The 17th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems,
Montreal, QC,
Sponsored: SIGSIM,
Contact: Azzedine Boukerche,
Email: boukerch@site.uottawa.ca

September 23–26
16th International Conference on Human-Computer Interaction with Mobile Devices and Services,
Toronto, ON Canada,
Sponsored: SIGCHI,
Contact: Aaron John Quigley,
Email: aquigley@gmail.com

September 24–26
1st International Conference on Information-Centric Networking,
Paris, France,
Sponsored: SIGCOMM,
Contact: Giovanna Carofiglio,
Email: giovanna.carofiglio@alcatel-lucent.com

DOI:10.1145/2643134

Article development led by acmqueue
queue.acm.org

Preventing script injection vulnerabilities through software design.

BY CHRISTOPH KERN

Securing the Tangled Web

SCRIPT INJECTION VULNERABILITIES are a bane of Web application development: deceptively simple in cause and remedy, they are nevertheless surprisingly difficult to prevent in large-scale Web development.

Cross-site scripting (XSS)[2,7,8] arises when insufficient data validation, sanitization, or escaping within a Web application allow an attacker to cause browser-side execution of malicious JavaScript in the application's context. This injected code can then do whatever the attacker wants, using the privileges of the victim. Exploitation of XSS bugs results in complete (though not necessarily persistent) compromise of the victim's session with the vulnerable application. This article provides an overview of how XSS vulnerabilities arise and why it is so difficult to avoid them in real-world Web application software development. Software design patterns developed at Google to address the problem are then described.

A key goal of these design patterns

IMAGE BY PHOTOBANK GALLERY

is to confine the potential for XSS bugs to a small fraction of an application's code base, significantly improving one's ability to reason about the absence of this class of security bugs. In several software projects within Google, this approach has resulted in a substantial reduction in the incidence of XSS vulnerabilities.

Most commonly, XSS vulnerabilities result from insufficiently validating, sanitizing, or escaping strings that are derived from an *untrusted source* and passed along to a *sink* that interprets them in a way that may result in script execution.

Common sources of untrustworthy data include HTTP request parameters, as well as user-controlled data located in persistent data stores. Strings are often concatenated with or interpolated into larger strings before assignment to a sink. The most frequently encountered sinks relevant to XSS vulnerabilities are those that interpret the assigned value as HTML markup, which includes server-side HTTP responses of MIME-type `text/html`, and the `Element.prototype.innerHTML` Document Object Model (DOM)[8] property in browser-side JavaScript code.

Figure 1a shows a slice of vulnerable code from a hypothetical photo-sharing application. Like many modern Web applications, much of its user-interface logic is implemented in browser-side JavaScript code, but the observations made in this article transfer readily to applications whose UI is implemented via traditional server-side HTML rendering.

In code snippet (1) in the figure, the application generates HTML markup for a notification to be shown to a user when another user invites the former to view a photo album. The generated markup is assigned to the `innerHTML` property of a DOM

A Subtle XSS Bug

The following code snippet intends to populate a DOM element with markup for a hyperlink (an HTML anchor element):

```
var escapedCat = goog.string.htmlEscape(category);
var jsEscapedCat = goog.string.escapeString(escapedCat);
catElem.innerHTML = '<a onclick="createCategoryList(\'' +
         jsEscapedCat + '\')">' + escapedCat + '</a>';
```

The anchor element's click-event handler, which is invoked by the browser when a user clicks on this UI element, is set up to call a JavaScript function with the value of `category` as an argument. Before interpolation into the HTML markup, the value of `category` is HTML-escaped using an escaping function from the JavaScript Closure Library. Furthermore, it is JavaScript-string-literal-escaped (replacing `'` with `\'` and so forth) before interpolation into the string literal within the `onclick` handler's JavaScript expression. As intended, for a value of `Flowers & Plants` for variable `category`, the resulting HTML markup is:

```
<a onclick="createCategoryList('Flowers &amp; Plants')">
      Flowers &amp; Plants</a>
```

So where's the bug? Consider a value for `category` of:

```
');attackScript();//
```

Passing this value through `htmlEscape` results in:

```
&#39;);attackScript();//
```

because `htmlEscape` escapes the single quote into an HTML character reference. After this, JavaScript-string-literal escaping is a no-op, since the single quote at the beginning of the page is *already HTML-escaped*. As such, the resulting markup becomes:

```
<a onclick="createCategoryList('&#39;);attackScript();//')">
      &#39;);attackScript();//</a>
```

When evaluating this markup, a browser will first HTML-unescape the value of the `onclick` attribute before evaluation as a JavaScript expression. Hence, the JavaScript expression that is evaluated results in execution of the attacker's script:

```
createCategoryList('');attackScript();//')
```

Thus, the underlying bug is quite subtle: the programmer invoked the appropriate escaping functions, but in the wrong order.

element (a node in the hierarchical object representation of UI elements in a browser window), resulting in its evaluation and rendering.

The notification contains the album's title, chosen by the *second* user. A malicious user can create an album titled:

```
<script>attackScript;</script>
```

Since no escaping or validation is applied, this attacker-chosen HTML is interpolated as-is into the markup generated in code snippet (1). This markup is assigned to the `innerHTML` sink, and hence evaluated in the context of the victim's session, executing the attacker-chosen JavaScript code.

To fix this bug, the album's title must be *HTML-escaped* before use in markup, ensuring that it is interpreted as plain text, not markup. HTML-escaping replaces HTML metacharacters such as `<`, `>`, `"`, `'`, and `&` with corresponding character entity references or numeric character references: `&lt;`, `&gt;`, `&quot;`, `&#39;`, and `&amp;`. The result will then be parsed as a substring in a text node or attribute value and will not introduce element or attribute boundaries.

As noted, most data flows with a potential for XSS are into sinks that interpret data as HTML markup. But other types of sinks can result in XSS bugs as well: Figure 1b shows another slice of the previously mentioned photo-sharing application, responsible for navigating the user interface after a login operation. After a fresh login, the app navigates to a preconfigured URL for the application's main page. If the login resulted from a session time-out, however, the app navigates back to the URL the user had visited before the time-out. Using a common technique for short-term state storage in Web applications, this URL is encoded in a parameter of the current URL.

The page navigation is implemented via assignment to the `window.location.href` DOM property, which browsers interpret as instruction to navigate the current window to the provided URL. Unfortunately, navigating a browser to a URL of the form `javascript:attackScript` causes execution of the URL's body as Java Script. In this scenario, the target URL is extracted from a parameter of the *current* URL, which is generally under attacker control (a malicious page visited by a victim can instruct the browser to navigate to an attacker-chosen URL).

Thus, this code is also vulnerable to XSS. To fix the bug, it is necessary to *validate* that the URL will not result in script execution when dereferenced, by ensuring that its scheme is benign—for example, `https`.

Why Is XSS So Difficult to Avoid?

Avoiding the introduction of XSS into nontrivial applications is a difficult problem in practice: XSS remains among the top vulnerabilities in Web applications, according to the Open Web Application Security Project (OWASP);[4] within Google it is the most common class of Web application vulnerabilities among those reported under Google's Vulnerability Reward Program (https://goo.gl/82zcPK).

Traditionally, advice (including my own) on how to prevent XSS has largely focused on:

- Training developers how to treat (by sanitization, validation, and/or escaping) untrustworthy values interpolated into HTML markup.[2,5]
- Security-reviewing and/or testing code for adherence to such guidance.

In our experience at Google, this approach certainly helps reduce the incidence of XSS, but for even moderately complex Web applications, it does not prevent introduction of XSS to a reasonably high degree of confidence. We see a combination of factors leading to this situation.

Subtle security considerations. As seen, the requirements for secure handling of an untrustworthy value depend on the context in which the value is used. The most commonly encountered context is string interpolation within the content of HTML markup elements; here, simple HTML-escaping suffices to prevent XSS bugs. Several special contexts, however, apply to various DOM elements and within certain kinds of markup, where embedded strings are interpreted as URLs, Cascading Style Sheets (CSS) expressions, or JavaScript code. To avoid XSS bugs, each of these contexts requires specific validation or escaping, or a combination of the two.[2,5] The accompanying sidebar, "A Subtle XSS Bug," shows this can be quite tricky to get right.

Complex, difficult-to-reason-about data flows. Recall that XSS arises from flows of untrustworthy, unvalidated/escaped data into injection-prone sinks. To assert the absence of XSS bugs in an application, a security reviewer must first find all such data sinks, and then inspect the surrounding code for context-appropriate validation and escaping of data transferred to the sink. When encountering an assignment that lacks validation and escaping, the reviewer must backward-trace this data flow until one of the following situations can be determined:

- The value is entirely under application control and hence cannot result in attacker-controlled injection.
- The value is validated, escaped, or otherwise safely constructed somewhere along the way.
- The value is in fact not correctly validated and escaped, and an XSS vulnerability is likely present.

Let's inspect the data flow into the `innerHTML` sink in code snippet (1) in Figure 1a. For illustration purposes, code snippets and data flows that require investigation are shown in red. Since no escaping is applied to `sharedAlbum.title`, we trace its origin to the `albums` entity (4) in persistent storage, via Web front-end code (2). This is, however, not the data's ultimate origin—the album name was previously entered by a different user (that is, originated in a different time context). Since no escaping was applied to this value anywhere along its flow from an ultimately untrusted source, an XSS vulnerability arises.

The primary goal of this approach is to limit code that could potentially give rise to XSS vulnerabilities to a very small fraction of an application's code base.

Similar considerations apply to the data flows in Figure 1b: no validation occurs immediately prior to the assignment to `window.location.href` in (5), so back-tracing is necessary. In code snippet (6), the code exploration branches: in the true branch, the value originates in a configuration entity in the data store (3) via the Web front end (8); this value can be assumed application-controlled and trustworthy and is safe to use without further validation. It is noteworthy that the persistent storage contains both trustworthy and untrustworthy data in different entities of the same schema—no blanket assumptions can be made about the provenance of stored data.

In the else-branch, the URL originates from a parameter of the *current* URL, obtained from `window.location.href`, which is an attacker-controlled source (7). Since there is no validation, this code path results in an XSS vulnerability.

Many opportunities for mistakes. Figures 1a and 1b show only two small slices of a hypothetical Web application. In reality, a large, nontrivial Web application will have hundreds if not thousands of branching and merging data flows into injection-prone sinks. Each such flow can potentially result in an XSS bug if a developer makes a mistake related to validation or escaping.

Exploring all these data flows and asserting absence of XSS is a monumental task for a security reviewer, especially considering an ever-changing code base of a project under active development. Automated tools that employ heuristics to statically analyze data flows in a code base can help. In our experience at Google, however, they do not substantially increase confidence in review-based assessments, since they are necessarily incomplete in their reasoning and subject to both false positives and false negatives. Furthermore, they have similar difficulties as human reviewers with reasoning about whole-system data flows across multiple system components, using a variety of programming languages, RPC (remote procedure call) mechanisms, and so forth, and involving flows traversing multiple time contexts across data stores.

Similar limitations apply to dynamic testing approaches: it is difficult to ascertain whether test suites provide adequate coverage for whole-system data flows.

Templates to the rescue? In practice, HTML markup, and interpolation points therein, are often specified using *HTML templates*. Template systems expose domain-specific languages for rendering HTML markup. An HTML markup template induces a function from template variables into strings of HTML markup.

Figure 1c illustrates the use of an HTML markup template (9): this example renders a user profile in the photo-sharing application, including the user's name, a hyperlink to a personal blog site, as well as free-form text allowing the user to express any special interests.

Some template engines support automatic escaping, where escaping operations are automatically inserted around each interpolation point into the template. Most template engines' auto-escape facilities are noncontextual and indiscriminately apply HTML escaping operations, but do not account for special HTML contexts such as URLs, CSS, and JavaScript.

Contextually auto-escaping template engines[6] infer the necessary validation and escaping operations required for the context of each template substitution, and therefore account for such special contexts.

Use of contextually auto-escaping template systems dramatically reduces the potential for XSS vulnerabilities: in (9), the substitution of untrustworthy values `profile.name` and `profile.blogUrl` into the resulting markup cannot result in XSS—the template system automatically infers the required HTML-escaping and URL-validation.

XSS bugs can still arise, however, in code that does not make use of templates, as in Figure 1a (1), or that involves non-HTML sinks, as in Figure 1b (5).

Furthermore, developers occasionally need to exempt certain substitutions from automatic escaping: in Figure 1c (9), escaping of `profile.aboutHtml` is explicitly suppressed because that field is assumed to contain a user-supplied message with simple, safe HTML markup (to support use of fonts, colors, and hyperlinks in the "about myself" user-profile field).

Unfortunately, there is an XSS bug: the markup in `profile.aboutHtml` ultimately originates in a rich-text editor implemented in browser-side code, but there is no server-side enforcement preventing an attacker from injecting malicious markup using a tampered-with client. This bug could arise in practice from a misunderstanding between front-end and back-end developers regarding responsibilities for data validation and sanitization.

Reliably Preventing the Introduction of XSS Bugs

In our experience in Google's security team, code inspection and testing do not ensure, to a reasonably high degree of confidence, the absence of XSS bugs in large Web applications. Of course, both inspection and testing provide tremendous value and will typically find *some* bugs in an application (perhaps even *most* of the bugs), but it is difficult to be sure whether or not they discovered *all* the bugs (or even *almost all* of them).

The primary goal of this approach is to limit code that could potentially give rise to XSS vulnerabilities to a very small fraction of an application's code base.

A key goal of this approach is to drastically reduce the fraction of code that could potentially give rise to XSS bugs. In particular, with this approach, an application is structured such that most of its code cannot be responsible for XSS bugs. The potential for vulnerabilities is therefore confined to infrastructure code such as Web application frameworks and HTML templating engines, as well as small, self-contained application-specific utility modules.

A second, equally important goal is to provide a developer experience that does not add an unacceptable degree of friction as compared with existing developer workflows.

Key components of this approach are:

- *Inherently safe APIs.* Injection-prone Web-platform and HTML-rendering APIs are encapsulated in wrapper APIs designed to be inherently safe against XSS in the sense that no use of such APIs can result in XSS vulnerabilities.
- *Security type contracts.* Special types are defined with contracts stipulating that their values are safe to use in specific contexts without further escaping and validation.
- *Coding guidelines.* Coding guidelines restrict direct use of injection-prone APIs, and ensure security review of certain security-sensitive APIs. Adherence to these guidelines can be enforced through simple static checks.

Inherently safe APIs. Our goal is to provide inherently safe wrapper APIs for injection-prone browser-side Web platform API sinks, as well as for server- and client-side HTML markup rendering.

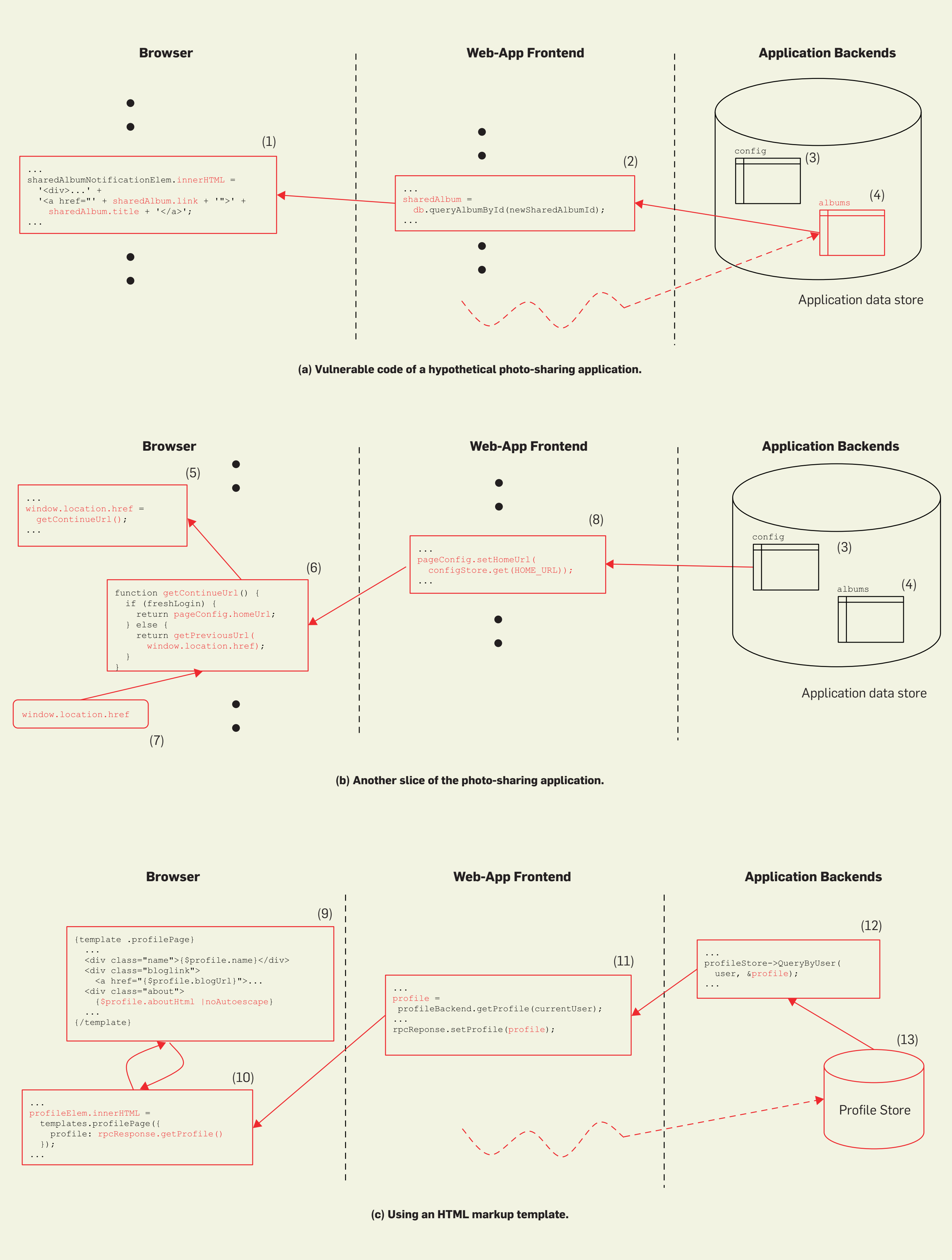
For some APIs, this is straightforward. For example, the vulnerable assignment in Figure 1b (5) can be replaced with the use of an inherently safe wrapper API, provided by the JavaScript Closure Library, as shown in Figure 2b (5'). The wrapper API validates at runtime that the supplied URL represents either a scheme-less URL or one with a known benign scheme.

Using the safe wrapper API ensures this code will not result in an XSS vulnerability, regardless of the provenance of the assigned URL. Crucially, none of the code in (5') nor its fan-in in (6-8) needs to be inspected for XSS bugs. This benefit comes at the very small cost of a runtime validation that is technically unnecessary if (and only if) the first branch is taken—the URL obtained from the configuration store is validated even though it is actually a trustworthy value.

In some special scenarios, the runtime validation imposed by an inherently safe API may be too strict. Such cases are accommodated via variants of inherently safe APIs that accept types with a security contract appropriate for the desired use context. Based on their contract, such values are exempt from runtime validation. This approach is discussed in more detail in the next section.

Strictly contextually auto-escaping template engines. Designing an inherently safe API for HTML rendering is more challenging. The goal is to devise APIs that guarantee that at each substitution point of data into a particular context within trusted HTML markup, data is appropriately validated, sanitized, and/or escaped, unless it can be demonstrated that a specific data item is safe to use in that context based on

Figure 1. XSS vulnerabilities in a hypothetical Web application.



(1)

```
...
sharedAlbumNotificationElem.innerHTML =
  '<div>...' +
  '<a href="' + sharedAlbum.link + '">' +
    sharedAlbum.title + '</a>';
...
```

(2)

```
...
sharedAlbum =
  db.queryAlbumById(newSharedAlbumId);
...
```

(a) Vulnerable code of a hypothetical photo-sharing application.

(5)

```
...
window.location.href =
  getContinueUrl();
...
```

(6)

```
function getContinueUrl() {
  if (freshLogin) {
    return pageConfig.homeUrl;
  } else {
    return getPreviousUrl(
      window.location.href);
  }
}
```

(7)

```
window.location.href
```

(8)

```
...
pageConfig.setHomeUrl(
  configStore.get(HOME_URL));
...
```

(b) Another slice of the photo-sharing application.

(9)

```
{template .profilePage}
  ...
  <div class="name">{$profile.name}</div>
  <div class="bloglink">
    <a href="{$profile.blogUrl}">...
  <div class="about">
    {$profile.aboutHtml |noAutoescape}
  ...
{/template}
```

(10)

```
...
profileElem.innerHTML =
  templates.profilePage({
    profile: rpcResponse.getProfile()
  });
...
```

(11)

```
...
profile =
 profileBackend.getProfile(currentUser);
...
rpcReponse.setProfile(profile);
```

(12)

```
...
profileStore->QueryByUser(
  user, &profile);
...
```

(c) Using an HTML markup template.

its provenance or prior validation, sanitization, or escaping.

These inherently safe APIs are created by strengthening the concept of contextually auto-escaping template engines[6] into SCAETEs (*strictly* contextually auto-escaping template engines). Essentially, a SCAETE places two additional constraints on template code:

- Directives that disable or modify the automatically inferred contextual escaping and validation are not permitted.
- A template may use only sub-templates that recursively adhere to the same constraint.

Security type contracts. In the form just described, SCAETEs do not account for scenarios where template parameters are intended to be used without validation or escaping, such as `aboutHtml` in Figure 1c—the SCAETE unconditionally validates and escapes all template parameters, and disallows directives to disable the auto-escaping mechanism.

Such use cases are accommodated through types whose contracts stipulate their values are safe to use in corresponding HTML contexts, such as "inner HTML," hyperlink URLs, executable resource URLs, and so forth. Type contracts are informal: a value satisfies a given type contract if it is known that it has been validated, sanitized, escaped, or constructed in a way that guarantees its use in the type's target context will not result in attacker-controlled script execution. Whether or not this is indeed the case is established by expert reasoning about code that creates values of such types, based on expert knowledge of the relevant behaviors of the Web platform.[8] As will be seen, such security-sensitive code is encapsulated in a small number of special-purpose libraries; application code uses those libraries but is itself not relied upon to correctly create instances of such types and hence does not need to be security-reviewed.

The following are examples of types and type contracts in use:

- `SafeHtml`. A value of type `SafeHtml`, converted to string, will not result in attacker-controlled script execution when used as HTML markup.
- `SafeUrl`. Values of this type will not result in attacker-controlled script execution when dereferenced as hyperlink URLs.
- `TrustedResourceUrl`. Values of this type are safe to use as the URL of an executable or "control" resource, such as the `src` attribute of a `<script>` element, or the source of a CSS style sheet. Essentially, this type promises that its value is the URL of a resource that is itself trustworthy.

These types are implemented as simple wrapper objects containing the underlying string value. Type membership in general cannot be established by runtime predicate, and it is the responsibility of the types' security-reviewed factory methods and builders to guarantee the type contract of any instance they produce. Type membership can be based on processing (for example, validation or sanitization), construction, and provenance, or a combination thereof.

SCAETEs use security contracts to designate exemption from automatic escaping: a substituted value is not subject to runtime escaping if the value is of a type whose contract supports its safe use in the substitution's HTML context.

Templates processed by a SCAETE give rise to functions that guarantee to emit HTML markup that will not result in XSS, assuming template parameters adhere to their security contracts, if applicable. Indeed, the result of applying a SCAETE-induced template function itself satisfies the `SafeHtml` type contract.

Figure 2c shows the application of SCAETE and security type contracts to the code slice of Figure 1c. Strict contextual escaping of the template in (9) disallows use of the `noAutoescape` directive. Simply removing it, however, would enable the automatic escaping of this value, which is in this case undesired. Instead, we change the `aboutHtml` field of the profile object to have `SafeHtml` type, which is exempt from automatic escaping. The use of this type is threaded through the system (indicated by the color green), across RPCs all the way to the value's origin in back-end code (12').

Unchecked conversions. Of course, eventually we need to create the required value of type `SafeHtml`. In the example, the corresponding field in persistent storage contains HTML markup that may be maliciously supplied by an attacker. Passing this untrusted markup through an HTML sanitizer to remove any markup that may result in script execution renders it safe to use in HTML context and thus produces a value that satisfies the `SafeHtml` type contract.

To actually create values of these types, *unchecked conversion* factory methods are provided that consume an arbitrary string and return an instance of a given wrapper type (for example, `SafeHtml` or `SafeUrl`) without applying any runtime sanitization or escaping.

Every use of such unchecked conversions must be carefully security reviewed to ensure that in all possible program states, strings passed to the conversion satisfy the resulting type's contract, based on context-specific processing or construction. As such, unchecked conversions should be used as rarely as possible, and only in scenarios where their use is readily reasoned about for security-review purposes.

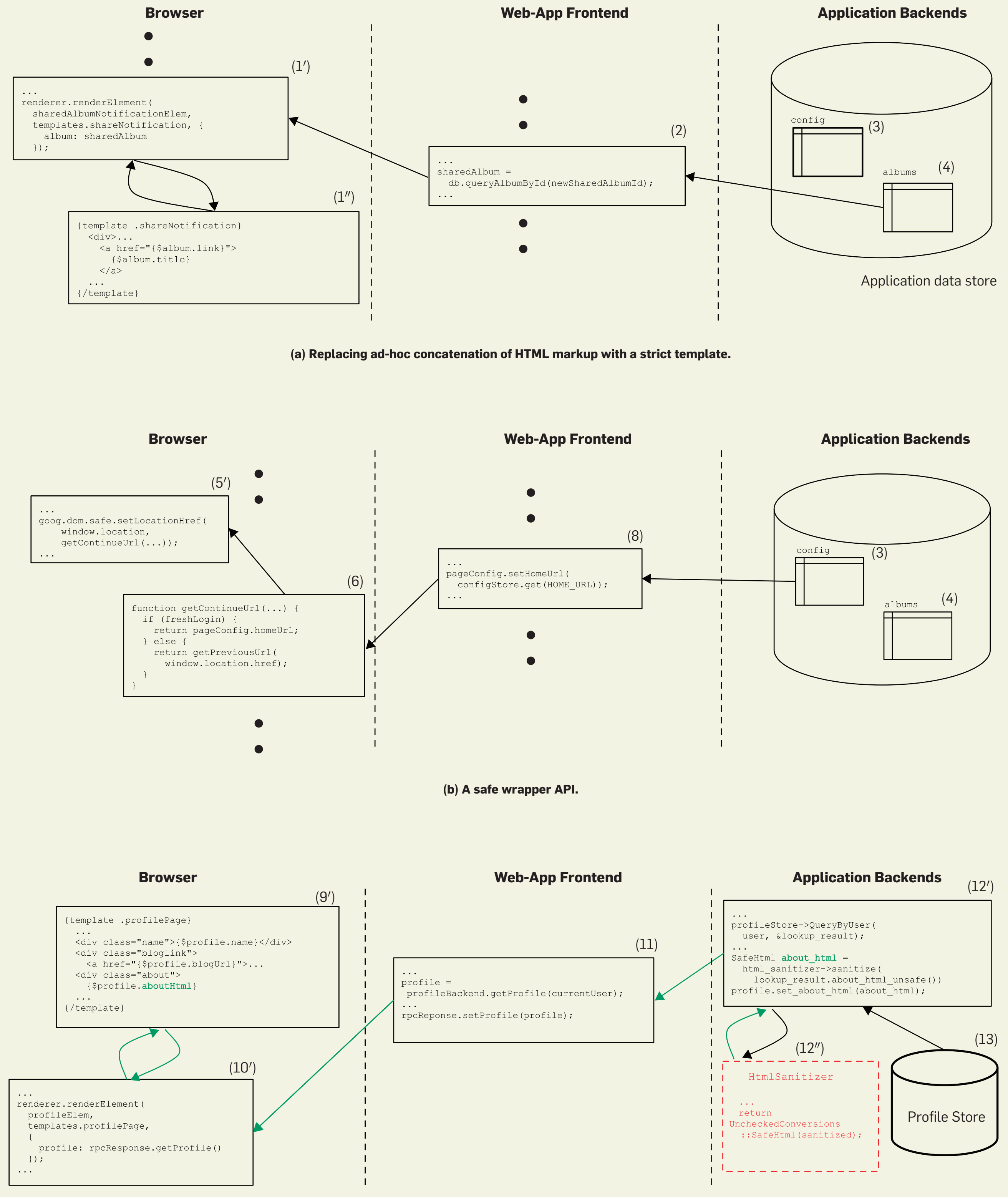
For example, in Figure 2c, the unchecked conversion is encapsulated in a library (12'') along with the HTML sanitizer implementation on whose correctness its use depends, permitting security review and testing in isolation.

Coding guidelines. For this approach to be effective, it must ensure developers never write application code that directly calls potentially injection-prone sinks, and that they instead use the corresponding safe wrapper API. Furthermore, it must ensure uses of unchecked conversions are designed with reviewability in mind, and are in fact security reviewed. Both constraints represent coding guidelines with which all of an application's code base must comply.

In our experience, automated enforcement of coding guidelines is necessary even in moderate-size projects—otherwise, violations are bound to creep in over time.

At Google we use the open source error-prone static checker[1] (https://goo.gl/SQXCvw), which is integrated into Google's Java tool chain, and a feature of Google's open source Closure Compiler (https://goo.gl/UyMVzp) to whitelist uses of specific methods and properties in JavaScript. Errors arising from use of a "banned" API include references to documentation for the corresponding safe API, advising developers on how to address

Figure 2. Preventing XSS through use of inherently safe APIs.



```
...
renderer.renderElement(
  sharedAlbumNotificationElem,
  templates.shareNotification, {
    album: sharedAlbum
  });
```

```
{template .shareNotification}
  <div>...
    <a href="{$album.link}">
      {$album.title}
    </a>
  ...
{/template}
```

```
...
sharedAlbum =
  db.queryAlbumById(newSharedAlbumId);
...
```

(a) Replacing ad-hoc concatenation of HTML markup with a strict template.

```
...
goog.dom.safe.setLocationHref(
    window.location,
    getContinueUrl(...));
...
```

```
function getContinueUrl(...) {
  if (freshLogin) {
    return pageConfig.homeUrl;
  } else {
    return getPreviousUrl(
      window.location.href);
  }
}
```

```
...
pageConfig.setHomeUrl(
  configStore.get(HOME_URL));
...
```

(b) A safe wrapper API.

```
{template .profilePage}
  ...
  <div class="name">{$profile.name}</div>
  <div class="bloglink">
    <a href="{$profile.blogUrl}">...
  <div class="about">
    {$profile.aboutHtml}
  ...
{/template}
```

```
...
renderer.renderElement(
  profileElem,
  templates.profilePage,
  {
    profile: rpcResponse.getProfile()
  });
...
```

```
...
profile =
 profileBackend.getProfile(currentUser);
...
rpcReponse.setProfile(profile);
```

```
...
profileStore->QueryByUser(
  user, &lookup_result);
...
SafeHtml about_html =
  html_sanitizer->sanitize(
    lookup_result.about_html_unsafe())
profile.set_about_html(about_html);
```

```
   HtmlSanitizer

  ...
  return
UncheckedConversions
  ::SafeHtml(sanitized);
```

(c) Using a type to represent safe HTML markup.

the error. The review requirement for uses of unchecked conversions is enforced via a package-visibility mechanism provided by Google's distributed build system.[3]

If tool-chain-integrated checks were not available, coding guidelines could be enforced through simpler lint-like tools.

In the photo-sharing example, such checks would raise errors for the assignments to `innerHTML` and `location.href` in figures 1a–1c and would advise the developer to use a corresponding inherently safe API instead. For assignments to `innerHTML`, this typically means replacing ad hoc concatenation of HTML markup with a strict template, rendered directly into the DOM element by the template system's runtime, as shown in Figure 2a.

Using such APIs prevents XSS bugs and largely relieves developers from thinking about and explicitly specifying escaping and data validation.

Putting It All Together

Revisiting the code slices of the example applications after they have been brought into adherence with the coding guideline shows (figures 2a–2c) that uses of injection-prone data sinks have been replaced with corresponding inherently safe APIs in (1'), (5'), (9') and (10’). Now, none of these code snippets can result in an XSS bug, and neither they nor their fan-in need to be inspected during a security review.

The only piece of code left requiring security code review (aside from infrastructure code such as the implementation of the SCAETE, its runtime, and API wrapper libraries) is the `HtmlSanitizer` package (12"), and specifically the package-local fan-in into the unchecked conversion to `SafeHtml`. Correctness of this conversion relies solely on the correctness of the HTML sanitizer, and this package can be security reviewed and tested in isolation. If a library is shared across multiple applications, its review cost is amortized among users.

Of course, there are limitations to the guarantees this approach can provide: first, the security reviewer may miss bugs in the security-relevant portion of the code (template systems, markup sanitizers, and so forth); second, application code may use constructs such as reflection that violate encapsulation of the types we rely on; finally, some classes of XSS bugs (in practice, relatively rare) cannot be addressed by generally applied contextual data validation and escaping as ensured by our design patterns, and these need to be addressed at other layers in Web application frameworks or in the design of individual Web applications.[7]

Developer impact. Comparing the vulnerable code slices in figures 1a–1c with their safe counterparts in figures 2a–2c shows our approach does not impose significant changes in developer workflow, nor major changes to code. For example, in Figure 2b (5'), we simply use a safe wrapper instead of the "raw" Web-platform API; otherwise, this code and its fan-in remain unchanged.

The coding guidelines do require developers to use safe APIs to generate HTML markup, such as the strict template in Figure 2a (1'). In return, however, using such APIs prevents XSS bugs and largely relieves developers from thinking about and explicitly specifying escaping and data validation.

Only in Figure 2c is a more significant change to the application required: the type of the `aboutHtml` field changes from `String` to `SafeHtml`, and use of this type is threaded through RPCs from back end to front end. Even here, the required changes are relatively confined: a change in the field's type and the addition of a call to the `HtmlSanitizer` library in back end code (12').

Such scenarios tend to be rare in typical Web applications; in the vast majority of uses the automatic runtime validation and escaping is functionally correct: most values of data flows into user-interface markup, both application-controlled and user-input-derived, tend to represent plain text, regular `http/https` URLs, and other values that validate and/or escape cleanly.

Practical Application

This design pattern has been applied in several open source and proprietary Web application frameworks and template engines in use at Google: support for strict contextual auto-escaping has been added to Closure Templates (https://goo/gl/Y4G9LK), AngularJS (https://goo.gl/RvQvXb), as well as a Google-proprietary templating system. Security engineers and infrastructure developers at Google have also implemented libraries of types such as `SafeHtml` and `SafeUrl`, and added inherently safe APIs to the Google Web Toolkit (https://goo.gl/dGk5G8), the JavaScript Closure Library (https://goo.gl/7nbXCg), and various Google-proprietary libraries and frameworks.

Decrease in incidence of XSS bugs. It is challenging to derive precise statistics regarding the impact of any particular approach to bug prevention: our design patterns prevent XSS bugs from being introduced in the first place, but we do not know how many bugs would have been introduced without their use.

We can, however, make observations based on bug counts in existing large projects that adopted our approach over time. Such observations can be considered anecdotal only, since bug counts are likely influenced by many variables such as code size and complexity and security-related developer education. Nevertheless, the observations suggest our approach significantly contributes to notable reductions in XSS vulnerabilities.

Several development teams of flagship Google applications have adopted these design patterns and coding guidelines. They have established static enforcement that all HTML markup is produced by strictly contextually auto-escaped templates, and they have disallowed direct use of certain injection-prone Web-platform APIs such as `innerHTML`.

One of the largest and most complex of these applications, using more than 1,000 HTML templates in the Closure Templates language, migrated to strict auto-escaping in early 2013. Throughout 2012 (before migration), 31 XSS bugs were filed in Google's bug tracker against this application. Post-migration, only four XSS bugs were filed in the year to mid-2014, and none at all in the first half of 2014. For another large application (also using more than 1,000 templates) whose migration is still in progress, there was a reduction from 21 to nine XSS bugs during the same time period.

Even without full compliance with the coding guidelines, some benefits can be realized: as the fraction of compliant code increases, the fraction of code that could be responsible for vulnerabilities shrinks, and confidence in the absence of bugs increases. While there is little reason not to write new code entirely in adherence to the guidelines, we can choose not to refactor certain existing code if the cost of refactoring exceeds benefits and if we already have confidence in that code's security through other means (for example, intensive review and testing).

Conclusion

Software design can be used to isolate the potential for XSS vulnerabilities into a very small portion of an application's code base. This makes it practical to intensively security-review and test just those portions of the code, resulting in a high degree of confidence that a Web application as a whole is not vulnerable to XSS bugs. Our approach is practically applicable to large, complex, real-world Web applications, and it has resulted in significant reduction of XSS bugs in several development projects.

This approach to what is fundamentally a difficult problem involving whole-system data flows incorporates two key principles:

- Based on the observation that in typical Web apps, it is functionally correct to conservatively runtime-escape and -validate the vast majority of data flowing into injection-prone sinks, we choose to treat all string-typed values as potentially untrustworthy and subject to runtime validation and escaping, regardless of their provenance. This design choice altogether obviates the need for whole-program reasoning about the vast majority of whole-system data flows in a typical Web application.
- Only in scenarios where default, runtime validation and escaping is functionally incorrect, we employ type contracts to convey that certain values are already safe to use in a given context. This use of types permits compositional reasoning about whole-system data flows and allows security experts to review security-critical code in isolation, based on package-local reasoning.

Our coding guidelines impose certain constraints on application code (though they typically require only limited changes to existing code). In contrast, many existing approaches to the prevention and detection of XSS aim to be applicable to existing, unmodified code. This requirement makes the problem much more difficult, and generally requires the use of complex whole-program static and/or dynamic data-flow analysis techniques. For an overview of existing work in this area, see Mike Samuel et al.[6] Relaxing this requirement negates the need for special-purpose tools and technologies (such as runtime taint tracking or whole-program static analysis), allowing us to rely solely on the combination of software design, coding guidelines enforceable by very simple static checks, existing language-native type systems, and a small enhancement to existing contextually auto-escaping template systems. Thus, our approach can be used in applications written in a variety of programming languages, without placing special requirements on tool chains, build systems, or runtime environments.

Related articles on queue.acm.org

Fault Injection in Production
John Allspaw
http://queue.acm.org/detail.cfm?id=2353017

High Performance Web Sites
Steve Souders
http://queue.acm.org/detail.cfm?id=1466450

Vicious XSS
George Neville-Neil
http://queue.acm.org/detail.cfm?id=1113330

References

1. Aftandilian, E., Sauciuc, R., Priya, S. and Krishnan, S. Building useful program analysis tools using an extensible Java compiler. *International Working Conference on Source Code Analysis and Manipulation* (2012), 14–23.
2. Daswani, N., Kern, C. and Kesavan, A. *Foundations of Security: What Every Programmer Needs to Know.* Apress, 2007.
3. Morgenthaler, J.D., Gridnev, M., Sauciuc, R. and Bhansali, S. Searching for build debt: Experiences managing technical debt at Google. Third International Workshop on Managing Technical Debt (2012), 1–6.
4. OWASP. Top 10 List, 2013; https://www.owasp.org/index.php/Top_10_2013-Top_10.
5. OWASP. XSS (cross site scripting) prevention cheat sheet, 2014; https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet.
6. Samuel, M., Saxena, P. and Song, D. Context-sensitive auto-sanitization in Web templating languages using type qualifiers. *Proceedings of the 18th ACM Conference on Computer and Communications Security* (2011), 587–600.
7. Su, Z. and Wasserman, G. The essence of command injection attacks in Web applications. In Proceedings of POPL (2006); http//dl.acm.org/citation.cfm?=1111070
8. Zalewski, M. *The Tangled Web: A Guide to Securing Modern Web Applications.* No Starch Press, 2012.

Christoph Kern (xtof@google.com) is an information security engineer at Google. His primary focus is on designing APIs and frameworks that make it easy for developers to write secure software and eliminate or reduce the risk of accidentally introducing security bugs.

DOI:10.1145/2643130

Article development led by acmqueue
queue.acm.org

An informal survey of real-world communications failures.

BY PETER BAILIS AND KYLE KINGSBURY

The Network Is Reliable

"THE NETWORK IS RELIABLE" tops Peter Deutsch's classic list of "Eight fallacies of distributed computing," all [of which] "prove to be false in the long run and all [of which] cause big trouble and painful learning experiences" (https://blogs.oracle.com/jag/resource/Fallacies.html). Accounting for and understanding the implications of network behavior is key to designing robust distributed programs—in fact, six of Deutsch's "fallacies" directly pertain to limitations on networked communications. This should be unsurprising: the ability (and often requirement) to communicate over a shared channel is a defining characteristic of distributed programs, and many of the key results in the field pertain to the possibility and impossibility of performing distributed computations under particular sets of network conditions.

For example, the celebrated FLP impossibility result[9] demonstrates the inability to guarantee consensus in an asynchronous network (that is, one facing indefinite communication *partitions* between processes) with one faulty process. This means that, in the presence of unreliable (untimely) message delivery, basic operations such as modifying the set of machines in a cluster (that is, maintaining group membership, as systems such as Zookeeper are tasked with today) are not guaranteed to complete in the event of both network asynchrony and individual server failures. Related results describe the inability to guarantee the progress of serializable transactions,[7] linearizable reads/writes,[11] and a variety of useful, programmer-friendly guarantees under adverse conditions.[3] The implications of these results are not simply academic: these impossibility results have motivated a proliferation of systems and designs offering a range of alternative guarantees in the event of network failures.[5] However, under a friendlier, more reliable network that guarantees timely message delivery, FLP and many of these related results no longer hold:[8] by making stronger guarantees about network behavior, we can circumvent the programmability implications of these impossibility proofs.

Therefore, the degree of reliability in deployment environments is critical in robust systems design and directly determines the kinds of operations that systems can reliably perform without waiting. Unfortunately, the degree to which networks are *actually* reliable in the real world is the subject of considerable and evolving debate. Some have claimed that networks are reliable (or that partitions are rare enough in practice) and that we are too concerned with designing for theoretical failure modes. Conversely, others attest that partitions do occur in their deployments, and that, as James Hamilton

IMAGE BY DALIBOR ZIVOTIC

of Amazon Web Services neatly summarizes "network partitions should be rare but net gear continues to cause more issues than it should" (http://bit.ly/1mD8E3q). So who's right?

A key challenge in this discussion is the lack of evidence. We have few normalized bases for comparing network and application reliability—and even less data. We can track link availability and estimate packet loss, but understanding the end-to-end effect on applications is more difficult. The scant evidence we have is difficult to generalize: it is often deployment-specific and closely tied to particular vendors, topologies, and application designs. Worse, even when organizations have a clear picture of their network's behavior, they rarely share specifics. Finally, distributed systems are designed to resist failure, which means *noticeable* outages often depend on complex interactions of failure modes. Many applications silently degrade when the network fails, and resulting problems may not be understood for some time, if ever.

As a result, much of what we believe about the failure modes of real-world distributed systems is founded on guesswork and rumor. Sysadmins and developers will swap stories over beer, but detailed, public postmortems and comprehensive surveys of network availability are few and far between. In this article, we'd like to informally bring a few of these stories (which, in most cases, are unabashedly anecdotal) together. Our focus is on descriptions of actual network behavior when possible, and (more often), when not, on the implications of network failures and asynchrony for real-world systems deployments. We believe this is a first step toward a more open and honest discussion of real-world partition behavior, and, ultimately, toward more robust distributed systems design.

Rumblings From Large Deployments

To start off, let's consider evidence from big players in distributed systems: companies running globally distributed infrastructure with hundreds of thousands of servers. These reports perhaps best summarize operations in the large, distilling the experience of operating what are likely the biggest distributed systems ever deployed. These companies' publications (unlike many of the reports we will examine later) often capture aggregate system behavior and large-scale statistical trends and indicate (often obliquely) that partitions are of concern in their deployments.

The degree to which networks are *actually* reliable in the real world is the subject of considerable and evolving debate.

A team from the University of Toronto and Microsoft Research studied the behavior of network failures in several of Microsoft's datacenters.[12] They found an average failure rate of 5.2 devices per day and 40.8 links per day, with a median time to repair of approximately five minutes (and a maximum of one week). While the researchers note that correlating link failures and communication partitions is challenging, they estimate a median packet loss of 59,000 packets per failure. Perhaps more concerning is their finding that network redundancy improves median traffic by only 43%; that is, network redundancy does not eliminate common causes of network failure.

A joint study between researchers at UCSD and HP Labs examined the causes and severity of network failures in HP's managed networks by analyzing support ticket data (http://www.hpl.hp.com/techreports/2012/HPL-2012-101.pdf). Connectivity-related tickets accounted for 11.4% of support tickets (14% of which were of the highest priority level), with a median incident duration of 2 hours and 45 minutes for the highest-priority tickets and a median duration of 4 hours 18 minutes for all tickets.

Google's paper describing the design and operation of Chubby, their distributed lock manager, outlines the root causes of 61 outages over 700 days of operation across several clusters (http://research.google.com/archive/chubby-osdi06.pdf). Of the nine outages that lasted more than 30 seconds, four were caused by network maintenance and two were caused by "suspected network connectivity problems."

In "Design Lessons and Advice from Building Large Scale Distributed Systems" (http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf), Jeff Dean suggested a typical first year for a new Google cluster involves:

- Five racks go wonky (40–80 machines seeing 50% packet loss)
- Eight network maintenances (four might cause ~30-minute random connectivity losses)
- Three router failures (have to immediately pull traffic for an hour)

While Google does not tell us much about the application-level consequences of their network partitions, Dean suggested they were of concern, citing the perennial challenge of creating "easy-to-use abstractions for resolving conflicting updates to multiple versions of a piece of state," useful for "reconciling replicated state in different data centers after repairing a network partition."

Amazon's Dynamo paper (http://bit.ly/1mDs0Yh) frequently cites the incidence of partitions as a key design consideration. Specifically, the authors note they rejected designs from "traditional replicated relational database systems" because they "are not capable of handling network partitions."

Yahoo! PNUTS/Sherpa was designed as a distributed database operating in geographically distinct datacenters. Originally, PNUTS supported a strongly consistent timeline consistency operation, with one master per data item. However, the developers noted that, in the event of network partitioning or server failures, this design decision was too restrictive for many applications:[16]

The first deployment of Sherpa supported the timeline-consistency model—namely, all replicas of a record apply all updates in the same order—and has API-level features to enable applications to cope with asynchronous replication. Strict adherence leads to difficult situations under network partitioning or server failures. These can be partially addressed with override procedures and local data replication, but in many circumstances, applications need a relaxed approach.

According to the same report, PNUTS now offers weaker consistency alternatives providing availability during partitions.

Datacenter Network Failures

Datacenter networks are subject to power failure, misconfiguration, firmware bugs, topology changes, cable damage, and malicious traffic. Their failure modes are accordingly diverse.

As Microsoft's SIGCOMM paper suggests, redundancy does not always prevent link failure. When a power distribution unit failed and took down one of two redundant top-of-rack switches, Fog Creek lost service for a subset of customers on that rack but remained consistent and available for most users. However, the other switch in that rack *also* lost power for undetermined reasons. That failure isolated the two neighboring racks from each other, taking down all On Demand services.

During a planned network reconfiguration to improve reliability, Fog Creek suddenly lost access to its network.[10]

A network loop had formed between several switches. The gateways controlling access to the switch management network were isolated from each other, generating a split-brain scenario. Neither was accessible due to a…multi-switch BPDU (bridge protocol data unit) flood, indicating a spanning-tree flap. This is most likely what was changing the loop domain.

According to the BPDU standard, the flood should not have happened. But it did, and this deviation from the system's assumptions resulted in two hours of total service unavailability.

To address high latencies caused by a daisy-chained network topology, Github installed a set of aggregation switches in its datacenter (https://github.com/blog/1346-network-problems-last-friday). Despite a redundant network, the installation process resulted in bridge loops, and switches disabled links to prevent failure. This problem was quickly resolved, but later investigation revealed that many interfaces were still pegged at 100% capacity.

While that problem was under investigation, a misconfigured switch triggered aberrant automatic fault detection behavior: when one link was disabled, the fault detector disabled *all* links, leading to 18 minutes of downtime. The problem was traced to a firmware bug preventing switches from updating their MAC address caches correctly, forcing them to broadcast most packets to every interface.

In December 2012, a planned software update on an aggregation switch caused instability at Github (https://github.com/blog/1364-downtime-last-saturday). To collect diagnostic information, the network vendor killed a particular software agent running on one of the aggregation switches.

Github's aggregation switches are clustered in pairs using a feature called MLAG, which presents two physical switches as a single layer-2 device. The MLAG failure detection protocol relies on both Ethernet link state and a logical heartbeat message exchanged between nodes. When the switch agent was killed, it was unable to shut down the Ethernet link, preventing the still-healthy aggregation switch from handling link aggregation, spanning-tree, and other L2 protocols. This forced a spanning-tree leader election and reconvergence for all links, blocking all traffic between access switches for 90 seconds.

This 90-second network partition caused fileservers using Pacemaker and DRBD for HA failover to declare each other dead, and to issue STONITH (Shoot The Other Node In The Head) messages to one another. The network partition delayed delivery of those messages, causing some fileserver pairs to believe they were *both* active. When the network recovered, both nodes shot each other at the same time. With both nodes dead, files belonging to the pair were unavailable.

To prevent filesystem corruption, DRBD requires that administrators ensure the original primary node is still the primary node before resuming replication. For pairs where both nodes were primary, the ops team had to examine log files or bring each node online in isolation to determine its state. Recovering those downed fileserver pairs took five hours, during which Github service was significantly degraded.

Cloud Networks

Large-scale virtualized environments are notorious for transient latency, dropped packets, and full-blown network partitions, often affecting a particular software version or availability zone. Sometimes the failures occur between specific subsections of the provider's datacenter, revealing planes of cleavage in the underlying hardware topology.

In a comment on Call me maybe: MongoDB (http://aphyr.com/posts/284-call-me-maybe-mongodb), Scott Bessler observed exactly the same failure mode Kyle demonstrated in the Jepsen post:

[This scenario] happened to us today when EC2 West region had network issues that caused a network partition that separated PRIMARY from its 2 SECONDARIES in a 3 node replset. 2 hours later the old primary rejoined and rolled back everything on the new primary.

This partition caused two hours of write loss. From our conversations with large-scale MongoDB users, we gather that network events causing failover on EC2 are common. Simultaneous primaries accepting writes for multiple days are anecdotally common.

Outages can leave two nodes connected to the Internet but unable to see each other. This type of partition is especially dangerous, as writes to both sides of a partitioned cluster can cause inconsistency and lost data. Paul Mineiro reports exactly this scenario in a Mnesia cluster (http://bit.ly/1zrVxI1), which diverged overnight. The cluster's state was not critical, so the operations team simply nuked one side of the cluster. They concluded "the experience has convinced us that we need to prioritize up our network partition recovery strategy."

Network disruptions in EC2 can affect only certain groups of nodes. For instance, one report of a total partition between the front-end and back-end servers states that a site's servers lose their connections to all back-end instances for a few seconds, several times a month (https://forums.aws.amazon.com/thread.jspa?messageID=454155). Even though the disruptions were short, they resulted in 30–45 minute outages and a corrupted index for ElasticSearch. As problems escalated, the outages occurred "2 to 4 times a day."

On April 21, 2011, Amazon Web Services suffered unavailability for 12 hours,[2] causing hundreds of high-profile websites to go offline. As a part of normal AWS scaling activities, Amazon engineers had shifted traffic away from a router in the Elastic Block Store (EBS) network in a single U.S. East Availability Zone (AZ), but, due to incorrect routing policies:

…many EBS nodes in the affected Availability Zone were completely isolated from other EBS nodes in its cluster. Unlike a normal network interruption, this change disconnected both the pri-

mary and secondary network simultaneously, leaving the affected nodes completely isolated from one another.

The partition, coupled with aggressive failure-recovery code, caused a mirroring storm that caused network congestion and triggered a previously unknown race condition in EBS. EC2 was unavailable for approximately 12 hours, and EBS was unavailable or degraded for over 80 hours.

The EBS failure also caused an outage in Amazon's Relational Database Service. When one AZ fails, RDS is designed to fail over to a different AZ. However, 2.5% of multi-AZ databases in US-East failed to fail over due to a bug in the fail-over protocol.

This correlated failure caused widespread outages for clients relying on AWS. For example, Heroku reported between 16 and 60 hours of unavailability for their users' databases.

On July 18, 2013, Twilio's billing system, which stores account credits in Redis, failed.[19] A network partition isolated the Redis primary from all secondaries. Because Twilio did not promote a new secondary, writes to the primary remained consistent. However, when the primary became visible to the secondaries again, all secondaries simultaneously initiated a full resynchronization with the primary, overloading it and causing Redis-dependent services to fail.

The ops team restarted the Redis primary to address the high load. However, upon restart, the Redis primary reloaded an incorrect configuration file, which caused it to enter read-only mode. With all account balances at zero, and in read-only mode, every Twilio API call caused the billing system to automatically recharge customer credit cards. 1.1% of customers were overbilled over a period of 40 minutes. For example, Appointment Reminder reported that every SMS message and phone call they issued resulted in a $500 charge to their credit card, which stopped accepting charges after $3,500.

Twilio recovered the Redis state from an independent billing system—a relational datastore—and after some hiccups, restored proper service, including credits to affected users.

Hosting Providers

Running your own datacenter can be cheaper and more reliable than using public cloud infrastructure, but it means you have to be a network and server administrator. What about hosting providers, which rent dedicated or virtualized hardware to users and often take care of the network and hardware setup for you?

Freistil IT hosts their servers with a colocation/managed-hosting provider. Their monitoring system alerted Freistil to 50%–100% packet loss localized to a specific datacenter.[15] The network failure, caused by a router firmware bug, returned the next day. Elevated packet loss caused the GlusterFS distributed filesystem to enter split-brain undetected:

...we became aware of [problems] in the afternoon when a customer called our support hotline because their website failed to deliver certain image files. We found that this was caused by a split-brain situation...and the self-heal algorithm built into the Gluster filesystem was not able to resolve this inconsistency between the two data sets.

Repairing that inconsistency led to a "brief overload of the web nodes because of a short surge in network traffic."

Anecdotally, many major managed hosting providers experience network failures. One company running 100–200 nodes on a major hosting provider reported that in a 90-day period the provider's network went through five distinct periods of partitions. Some partitions disabled connectivity between the provider's cloud network and the public Internet, and others separated the cloud network from the provider's internal managed-hosting network.

A post to Linux-HA details a long-running partition between a Heartbeat pair (http://bit.ly/1k9Ym6V), in which two Linode VMs each declared the other dead and claimed a shared IP for themselves. Successive posts suggest further network problems: email messages failed to dispatch due to DNS resolution failure, and nodes reported the network unreachable. In this case, the impact appears to have been minimal, in part because the partitioned application was just a proxy.

Wide Area Networks

While we have largely focused on failures over local area networks (or near-local networks), wide area network (WAN) failures are also common, if less frequently documented. These failures are particularly interesting because there are often fewer redundant WAN routes and because systems guaranteeing high availability (and disaster recovery) often require distribution across multiple datacenters. Accordingly, graceful degradation under partitions or increased latency is especially important for geographically widespread services.

Researchers at the UCSD analyzed five years of operation in the CENIC wide-area network,[18] which contains over 200 routers across California. By cross-correlating link failures and additional external BGP and traceroute data, they discovered over 508 "isolating network partitions" that caused connectivity problems between hosts. Average partition duration ranged from six minutes for software-related failures to over 8.2 hours for hardware-related failures (median 2.7 and 32 minutes; 95th percentile of 19.9 minutes and 3.7 days, respectively).

PagerDuty designed their system to remain available in the face of node, datacenter, or even provider failure; their services are replicated between two EC2 regions and a datacenter hosted by Linode. On April 13, 2013, an AWS peering point in northern California degraded, causing connectivity issues for one of PagerDuty's EC2 nodes. As latencies between AWS availability zones rose, the notification dispatch system lost quorum and stopped dispatching messages entirely.

Even though PagerDuty's infrastructure was designed with partition tolerance in mind, correlated failures due to a shared peering point between two datacenters caused 18 minutes of unavailability, dropping inbound API requests and delaying queued pages until quorum was re-established.

Global Routing Failures

Despite the high level of redundancy in Internet systems, some network failures take place on a global scale.

CloudFlare runs 23 datacenters with redundant network paths and anycast failover. In response to a DDoS attack against one of their customers, the CloudFlare operations team deployed a new firewall rule to drop packets of a specific size.[17] Juniper's FlowSpec protocol propagated that rule to all CloudFlare edge routers, but then:

What should have happened is that no packet should have matched that rule because no packet was actually that large. What happened instead is that the routers encountered the rule and then proceeded to consume all their RAM until they crashed.

Recovering from the failure was complicated by routers that failed to reboot automatically and by inaccessible management ports.

Even though some data centers came back online initially, they fell back over again because all the traffic across our entire network hit them and overloaded their resources.

CloudFlare monitors its network carefully, and the operations team had immediate visibility into the failure. However, coordinating globally distributed systems is complex, and calling on-site engineers to find and reboot routers by hand takes time. Recovery began after 30 minutes, and was complete after an hour of unavailability.

A firmware bug introduced as a part of an upgrade in Juniper Networks's routers caused outages in Level 3 communications' networking backbone in 2011. This subsequently knocked services including Time Warner Cable, RIM BlackBerry, and several U.K. Internet service providers offline.

There have been several global Internet outages related to BGP misconfiguration. Notably, in 2008, Pakistan Telecom, responding to a government edict to block YouTube.com, incorrectly advertised its (blocked) route to other providers, which hijacked traffic from the site and briefly rendered it unreachable.

In 2010, a group of Duke University researchers achieved a similar effect by testing an experimental flag in the BGP protocol (http://bit.ly/1rbAl4j). Similar incidents occurred in 2006, knocking sites such as Martha Stewart Living and the *New York Times* offline; in 2005, where a misconfiguration in Turkey attempted in a redirect for the entire Internet; and in 1997.

NICs and Drivers

Unreliable networking hardware and/or drivers are implicated in a broad array of partitions.

As a classic example of NIC unreliability, Marc Donges and Michael Chan describe how their popular Broadcom BCM5709 chip dropped inbound but not outbound packets (http://www.spinics.net/lists/netdev/msg210485.html). The primary server was unable to service requests, but, because it could still send heartbeats to its hot spare, the spare considered the primary alive and refused to take over. Their service was unavailable for five hours and did not recover without a reboot.

Despite the high level of redundancy in Internet systems, some network failures take place on a global scale.

Sven Ulland followed up, reporting the same symptoms with the BCM5709S chipset on Linux 2.6.32-41squeeze2. Despite pulling commits from mainline that supposedly fixed a similar set of issues with the bnx2 driver, Ulland's team was unable to resolve the issue until version 2.6.38.

As a large number of servers shipped the BCM5709, the larger impact of these firmware bugs was widely observed. For instance, the 5709 had a bug in their 802.3x flow control, leading to extraneous PAUSE frames when the chipset crashed or its buffer filled up. This problem was magnified by the BCM56314 and BCM56820 switch-on-a-chip devices (found in many top-of-rack switches), which, by default, sent PAUSE frames to any interface communicating with the offending 5709 NIC. This led to cascading failures on entire switches or networks.

The bnx2 driver could also cause transient or flapping network failures, as described in an ElasticSearch failure report. Meanwhile, the Broadcom 57711 was notorious for causing high latencies under load with jumbo frames, a particularly thorny issue for ESX users with iSCSI-backed storage.

A motherboard manufacturer failed to flash the EEPROM correctly for its Intel 82574–based system. The result was a very-hard-to-diagnose error in which an inbound SIP packet of a particular structure would disable the NIC.[14] Only a cold restart would bring the system back to normal.

After a scheduled upgrade, CityCloud noticed unexpected network failures in two distinct GlusterFS pairs, followed by a third.[6] Suspecting link aggregation, CityCloud disabled the feature on its switches and allowed self-healing operations to proceed.

Roughly 12 hours later, the network failures returned. CityCloud identified the cause as a driver issue and updated the downed node, returning service. However, the outage resulted in data inconsistency between GlusterFS pairs and data corruption between virtual machine file systems.

Application-Level Failures

Not all asynchrony originates in the physical network. Sometimes dropped or delayed messages are a consequence of crashes, program errors, OS scheduler latency, or overloaded processes. The following studies highlight the fact that communication failures—wherein the system delays or drops messages—can occur at any layer of the software stack, and designs that expect synchronous communication may behave unexpectedly during periods of asynchrony.

Bonsai.io (http://www.bonsai.io/blog/2013/03/05/outage-post-mortem) discovered high CPU and memory use on an ElasticSearch node combined with difficulty connecting to various cluster components, likely a consequence of an "excessively high number of expensive requests being allowed through to the cluster."

Upon restarting the servers, the cluster split into two independent components. A subsequent restart resolved the split-brain behavior, but customers complained they were unable to delete or create indices. The logs revealed that servers were repeatedly trying to recover unassigned indices, which "poisoned the cluster's attempt to service normal traffic which changes the cluster state." The failure led to 20 minutes of unavailability and six hours of degraded service.

Stop-the-world garbage collection and blocking for disk I/O can cause runtime latencies on the order of seconds to minutes. As Searchbox IO and several other production users have found, GC pressure in an ElasticSearch cluster can cause secondary nodes to declare a primary dead and to attempt a new election (https://github.com/elasticsearch/elasticsearch/issues/2488). Due to non-majority quorum configuration, ElasticSearch elected two different primaries, leading to inconsistency and downtime. Surprisingly, even with majority quorums, due to protocol design, ElasticSearch does not currently prevent simultaneous master election; GC pauses and high IO_WAIT times due to I/O can cause split-brain behavior, write loss, and index corruption.

Stop-the-world garbage collection and blocking for disk I/O can cause runtime latencies on the order of seconds to minutes.

In 2012, a routine database migration caused unexpectedly high load on the MySQL primary at Github.[13] The cluster coordinator, unable to perform health checks against the busy MySQL server, decided the primary was down and promoted a secondary. The secondary had a cold cache and performed poorly, causing failover back to the original primary. The operations team manually halted this automatic failover and the site appeared to recover.

The next morning, the operations team discovered the standby MySQL node was no longer replicating changes from the primary. Operations decided to disable the coordinator's maintenance mode and allow the replication manager to fix the problem. Unfortunately, this triggered a segfault in the coordinator, and, due to a conflict between manual configuration and the automated replication tools, github.com was rendered unavailable.

The partition caused inconsistency in the MySQL database—both between the secondary and primary, and between MySQL and other data stores such as Redis. Because foreign key relationships were not consistent, Github showed private repositories to the wrong users' dashboards and incorrectly routed some newly created repositories.

When a two-node cluster partitions, there are no cases in which a node can reliably declare itself to be the primary. When this happens to a DRBD filesystem, as one user reported (http://bit.ly/1nbv4E), both nodes can remain online and accept writes, leading to divergent filesystem-level changes.

Short-lived failures can lead to long outages. In a Usenet post to novell.support.cluster-services, an admin reports that a two-node failover cluster running Novell NetWare experienced transient network outages. The secondary node eventually killed itself, and the primary (though still running) was no longer reachable by other hosts on the network. The post goes on to detail a series of network partition events correlated with backup jobs!

One VoltDB user reports regular network failures causing replica divergence (http://bit.ly/1mDeC4d) but also indicates their network logs included no dropped packets. Because this cluster had not enabled split-brain detection, both nodes ran as isolated primaries, causing significant data loss.

Sometimes, nobody knows why a system partitions. This RabbitMQ failure seems like one of those cases: few retransmits, no large gaps between messages, and no clear loss of connectivity between nodes (http://bit.ly/1qZROze). Increasing the partition detection timeout to two minutes reduced the frequency of partitions but did not prevent them altogether.

Another EC2 split-brain (http://bit.ly/1mDeIZA): a two-node cluster failed to converge on "roughly 1 out of 10 startups" when discovery messages took longer than three seconds to exchange. As a result, both nodes would start as primaries with the same cluster name. Since ElasticSearch does not demote primaries automatically, split-brain persisted until administrators intervened. Increasing the discovery timeout to 15 seconds resolved the issue.

There are a few scattered reports of Windows Azure partitions, such as this account of a RabbitMQ cluster that entered split-brain on a weekly basis (http://bit.ly/1sCN4Nw). There's also this report of an ElasticSearch split-brain (http://bit.ly/U5xAFS), but since Azure is a relative newcomer compared to EC2, descriptions of its network reliability are limited.

Where Do We Go From Here?

This article is meant as a reference point to illustrate that, according to a wide range of (often informal) accounts, communication failures occur in many real-world environments. Processes, servers, NICs, switches, and local and wide area networks can all fail, with real economic consequences. Network outages can suddenly occur in systems that have been stable for months at a time, during routine upgrades, or as a result of emergency maintenance. The consequences of these outages range from increased latency and temporary unavailability to inconsistency, corruption, and data loss. Split-brain is not an academic concern: it happens to all kinds of systems, sometimes for days on end. Partitions deserve serious consideration.

On the other hand, some networks really are reliable. Engineers at major financial firms have anecdotally reported that despite putting serious effort into designing systems that gracefully tolerate partitions, their networks rarely, if ever, exhibit partition behavior. Cautious engineering and aggressive network advances (along with lots of money) can prevent outages. Moreover, in this article, we have presented failure scenarios; we acknowledge it is much more difficult to demonstrate that network failures have not occurred!

However, not all organizations can afford the cost or operational complexity of highly reliable networks. From Google and Amazon (who operate commodity and/or low-cost hardware due to sheer scale) to one-person startups built on shoestring budgets, communication-isolating network failures are a real risk, in addition to the variety of other failure modes (including human error) that real-world distributed systems face.

It is important to consider this risk before a partition occurs, because it is much easier to make decisions about partition behavior on a whiteboard than to redesign, reengineer, and upgrade a complex system in a production environment—especially when it is throwing errors at your users. For some applications, failure *is* an option, but you should characterize and explicitly account for it as a part of your design. And finally, given the additional latency[1] and coordination benefits[4] of partition-aware designs, you might just find that accounting for these partitions delivers benefits in the average case as well.

We invite you to contribute your own experiences with or without network partitions. Open a pull request on https://github.com/aphyr/partitions-post (which, incidentally, contains all references), leave a comment, write a blog post, or release a post-mortem. Data will inform this conversation, future designs, and, ultimately, the availability of the systems we all depend on.

Related articles on queue.acm.org

Eventual Consistency Today: Limitations, Extensions, and Beyond
Peter Bailis and Ali Ghodsi
http://queue.acm.org/detail.cfm?id=2462076

The Antifragile Organization
Ariel Tseitlin
http://queue.acm.org/detail.cfm?id=2499552

Self-Healing Networks
Robert Poor, Cliff Bowman and Charlotte Burgess Auburn
http://queue.acm.org/detail.cfm?id=864027

References

1. Abadi, D. Consistency trade-offs in modern distributed database system design: CAP is only part of the story. *Computer 45* (2 (2012), 37–42; http://dl.acm.org/citation.cfm?id=2360959.
2. Amazon Web Services. Summary of the Amazon EC2 and Amazon RDS service disruption in the US East region, 2011; http://aws.amazon.com/message/65648/.
3. Bailis, P., Davidson, A., Fekete, A., Ghodsi, A., Hellerstein, J.M. and Stoica, I. Highly available transactions: virtues and limitations. In *Proceedings of VLDB 2014* (to appear); http://www.bailis.org/papers/hat-vldb2014.pdf.
4. Bailis, P., Fekete, A., Franklin, M.J., Ghodsi, A., Hellerstein, J.M. and Stoica, I. Coordination-avoiding database systems, 2014; http://arxiv.org/abs/1402.2237
5. Bailis, P. and Ghodsi, A. Eventual consistency today: Limitations, extensions, and beyond. *ACM Queue 11*, 3 (2013); http://queue.acm.org/detail.cfm?id=2462076 .
6. CityCloud, 2011; https://www.citycloud.eu/cloud-computing/post-mortem/.
7. Davidson, S.B., Garcia-Molina, H. and Skeen, D. Consistency in a partitioned network: A survey. *ACM Computing Surveys 17*, 3 (1985), 341–370; http://dl.acm.org/citation.cfm?id=5508.
8. Dwork, C., Lynch, M. and Stockmeyer, L. Consensus in the presence of partial synchrony. *JACM 35*, 2 (1988); 288–323. http://dl.acm.org/citation.cfm?id=42283.
9. Fischer, M.J., Lynch, N.A., Patterson, M.S. Impossibility of distributed consensus with one faulty process. *JACM 32*, 2 (1985), 374–382; http://dl.acm.org/citation.cfm?id=214121
10. Fog Creek Software. May 5–6 network maintenance post-mortem; http://status.fogcreek.com/2012/05/may-5-6-network-maintenance-post-mortem.html.
11. Gilbert, S. and Lynch, N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News 33*, 2 (2002), 51–59; http://dl.acm.org/citation.cfm?id=564601.
12. Gill, P., Jain, N., Nagappan, N. Understanding network failures in data centers: Measurement, analysis, and implications. In *Proceedings of SIGCOMM '11*; http://research.microsoft.com/enus/um/people/navendu/papers/sigcomm11netwiser.pdf.
13. Github. Github availability this week, 2012; https://github.com/blog/1261-github-availability-this-week.
14. Kielhofner, K. Packets of death; http://blog.krisk.org/2013/02/packets-of-death.html.
15. Lillich, J. Post mortem: Network issues last week; http://www.freistil.it/2013/02/post-mortem-network-issues-last-week/.
16. Narayan, P.P.S. Sherpa update, 2010; https://developer.yahoo.com/blogs/ydn/sherpa-7992.html#4.
17. Prince, M. Today's outage post mortem, 2013; http://blog.cloudflare.com/todays-outage-post-mortem-82515.
18. Turner, D., Levchenko, K., Snoeren, A. and Savage, S. California fault lines: Understanding the causes and impact of network failures. In *Proceedings of SIGCOMM '10*; http://cseweb.ucsd.edu/~snoeren/papers/cenic-sigcomm10.pdf.
19. Twilio. Billing incident post-mortem: breakdown, analysis and root cause; http://www.twilio.com/blog/2013/07/billing-incident-post-mortem.html.

Peter Bailis is a graduate student of computer science and a member of the AMPLab and BOOM projects at UC Berkeley. He studies database and distributed systems and blogs at http://bailis.org/blog and tweets as @pbailis.

Kyle Kingsbury (@aphyr) is the author of Riemann, Timelike, and a number of other open source packages. He also verifies distributed systems' safety claims as part of the Jepsen project.

DOI:10.1145/2643132

Article development led by acmqueue
queue.acm.org

Quality social science research and the privacy of human subjects require trust.

BY JON P. DARIES, JUSTIN REICH, JIM WALDO, ELISE M. YOUNG, JONATHAN WHITTINGHILL, ANDREW DEAN HO, DANIEL THOMAS SEATON, AND ISAAC CHUANG

Privacy, Anonymity, and Big Data in the Social Sciences

OPEN DATA HAS tremendous potential for science, but, in human subjects research, there is a tension between privacy and releasing high-quality open data. Federal law governing student privacy and the release of student records suggests that anonymizing student data protects student privacy. Guided by this standard, we de-identified and released a dataset from 16 massive open online courses (MOOCs) from MITx and HarvardX on the edX platform. In this article, we show that these and other de-identification procedures necessitate changes to datasets that threaten replication and extension of baseline analyses. In order to balance student privacy and the benefits of open data, we suggest focusing on protecting privacy *without* anonymizing data by instead expanding policies that compel researchers to uphold the privacy of the subjects in open datasets. If we want to have high-quality social science research and also protect the privacy of human subjects, we must eventually have trust in researchers. Otherwise, we will always have the strict trade-off between anonymity and science illustrated here.

The "open" in "massive open online courses" has many interpretations. Some MOOCs are hosted on open-source platforms, some use only openly licensed content, and most MOOCs are openly accessible to any learner without fee or prerequisites. We would like to add one more notion of openness: open access to data generated by MOOCs. We argue this is part of the responsibility of MOOCs, and that fulfilling this responsibility threatens current conventions of anonymity in policy and public perception.

In this spirit of open data, on May 30, 2014, as a team of researchers from Harvard and MIT that includes this author team, we announced the release of an open dataset containing student records from 16 courses conducted in the first year of the edX platform. (In May 2012, MIT and Harvard launched edX, a nonprofit platform for hosting and marketing MOOCs. MITx and HarvardX are the two respective institutional organizations focused on MOOCs.)[6] The dataset is a de-identified version of the dataset used to publish *HarvardX and MITx: The First Year of Open Online Courses*, a report revealing findings about student demographics, course-taking patterns, certification rates, and other measures of student behavior.[6] The goal for this data release was twofold: first, to allow other researchers to replicate the results of the analysis; and second, to allow researchers to conduct novel analyses beyond the original work, adding to the body of literature about open online courses.

PHOTOGRAPH BY FRANCK CAMHI

Within hours of the release, original analysis of the data began appearing on Twitter, with figures and source code. Two weeks after the release, the data journalism team at *The Chronicle of Higher Education* published "8 Things You Should Know about MOOCs," an article that explored new dimensions of the dataset, including the gender balance of the courses.[13] Within the first month of the release, the data had been downloaded more than 650 times. With surprising speed, the dataset began fulfilling its purpose: to allow the research community to use open data from online learning platforms to advance scientific progress.

The rapid spread of new research from this data is exciting, but this excitement is tempered by a necessary limitation of the released data: they represent a subset of the complete data. In order to comply with federal regulations on student privacy, the released dataset had to be de-identified. In this article, we demonstrate trade-offs between our need to meet the demands of federal regulations of student privacy, on the one hand, and our responsibility to release data for replication and downstream analyses, on the other. For example, the original analysis found approximately 5% of course registrants earned certificates. Some methods of de-identification cut that percentage in half.

It is impossible to anonymize identifiable data without the possibility of affecting some future analysis in some way. It is possible to quantify the difference between replications from the de-identified data and original findings; however, it is difficult to fully anticipate whether findings from novel analyses will result in valid insights or artifacts of de-identification. Higher standards for de-identification can lead to lower-value de-identified data. This could have a chilling effect on the motivations of social science researchers. If findings are likely to be biased by the de-identification process, why should researchers spend their scarce time on de-identified data?

At the launch of edX in May of 2012, the presidents of MIT and Harvard spoke about the edX platform, and the data generated by it, as a public good. If academic and independent researchers alike have access to data from MOOCs, the progress of research into online education will be faster and results can be furthered, refined, and tested. However, these ideals for open MOOC data are undermined if protecting student privacy means that open datasets are markedly different from the original data. The tension between privacy and open data is in need of a better solution than anonymized datasets. Indeed, the fundamental problem in our current regulatory framework may be an unfortunate and unnecessary conflation of privacy and anonymity. Skopek[17] outlines the difference between the two as follows:

> *...under the condition of privacy, we have knowledge of a person's identity, but not of an associated personal fact, whereas under the condition of anonymity, we have knowledge of a personal fact, but not of the associated person's identity. In this sense, privacy and anonymity are flip sides of each other. And for this*

reason, they can often function in opposite ways: whereas privacy often hides facts about someone whose identity is known by removing information and other goods associated with the person from public circulation, anonymity often hides the identity of someone about whom facts are known for the purpose of putting such goods into public circulation.

Realizing the potential of open data in social science requires a new paradigm for the protection of student privacy: either a technological solution such as differential privacy,[3] which separates analysis from possession of the data, or a policy-based solution that allows open access to possibly re-identifiable data while policing the uses of the data.

This article describes the motivations behind efforts to release learner data, the contemporary regulatory framework of student privacy, our efforts to comply with those regulations in creating an open dataset from MOOCs, and some analytical consequences of de-identification. From this case study in de-identification, we conclude that the scientific ideals of open data and the current regulatory requirements concerning anonymizing data are incompatible. Resolving that incompatibility will require new approaches that better balance the protection of privacy and the advancement of science in educational research and the social sciences more broadly.

Balancing Open Data and Student Privacy Regulations

As with open source code and openly licensed content, support for open data has been steadily building. In the U.S., government agencies have increased their expectations for sharing research data.[5] In 2003, the National Institutes of Health became the first federal agency to require research grant applicants to describe their plans for data sharing.[12] In 2013, the Office of Science and Technology Policy released a memorandum requiring the public storage of digital data from unclassified, federally funded research.[7] These trends dovetailed with growing interest in data sharing in the learning sciences community. In 2006, researchers from Carnegie Mellon University opened DataShop, a repository of event logs from intelligent tutoring systems and one of the largest sources of open data in educational research outside the federal government.[8]

Open data has tremendous potential across the scientific disciplines to facilitate greater transparency through replication and faster innovation through novel analyses. It is particularly important in research into open, online learning such as MOOCs. A study released earlier this year[1] estimates there are over seven million people in the U.S. alone who have taken at least one online course, and that that number is growing by 6% each year. These students are taking online courses at a variety of institutions, from community colleges to research universities, and open MOOC data will facilitate research that could be helpful to all institutions with online offerings.

Open data can also facilitate cooperation between researchers with different domains of expertise. As George Siemens, the president of the Society for Learning Analytics Research, has argued, learning research involving large and complex datasets requires interdisciplinary collaboration between data scientists and educational researchers.[16] Open data sets make it easier for researchers in these two distinct domains to come together.

While open educational data has great promise for advancing science, it also raises important questions about student privacy. In higher education, the cornerstone of student privacy law is the Family Educational Rights and Privacy Act (FERPA)—a federal privacy statute that regulates access to and disclosure of a student's educational records. In our de-identification procedures, we aimed to comply with FERPA, although not all institutions consider MOOC learners to be subject to FERPA.[11]

FERPA offers protections for personally identifiable information (PII) within student records. Per FERPA, PII cannot be disclosed, but if PII is removed from a record, then the student becomes anonymous, privacy is protected, and the resulting de-identified data can be disclosed to anyone. FERPA thus equates anonymity—the removal of PII—with privacy.

FERPA's PII definition includes some statutorily defined categories, such as name, address, Social Security Number, and mother's maiden name, but also:

> *...other information that, alone or in combination, is linked or linkable to a specific student that would allow a reasonable person in the school community, who does not have personal knowledge of the relevant circumstances, to identify the student with reasonable certainty.*

In assessing the reasonable certainty of identification, the educational institution is supposed to take into account other data releases that might increase the chance of identification.[22] Therefore, an adequate de-identification procedure must not only remove statutorily required elements, but also quasi-identifiers. These quasi-identifiers are pieces of information that can be uniquely identifying in combination with each other or with additional data sources from outside the student records. They are not defined by statute or regulatory guidance from the Department of Education but left up to the educational institution to define.[22]

The potential for combining quasi-identifiers to uniquely identify individuals is well established. For example, Sweeney[21] has demonstrated that 87% of the U.S. population can be uniquely identified with a reasonable degree of certainty by a combination of ZIP code, date of birth, and gender. These risks are further heightened in open, online learning environments because of the public nature of the activity. As another example, some MOOC students participate in course discussion forums—which, for many courses, remain available online beyond the course end date. Students' usernames are displayed beside their posts, allowing for linkages of information across courses, potentially revealing students who enroll for unique combinations of courses. A very common use of the discussion forums early in a course is a self-introduction thread where students state their age and location among other PII. Meanwhile, another source of identifying data is social media. It is conceivable that students could verbosely log their online education on Facebook or Twitter, tweeting as soon as they register for a new course or mentioning their course grade in a Facebook post. Given these external sources, an argument can be made that many columns in the dataset person-course that would not typically be thought of as identifiers could qualify as quasi-identifiers.

The regulatory framework defined by FERPA guided our efforts to de-identify the person-course dataset for an open release. Removing direct identifiers such as students' usernames and IP addresses was straightforward, but the challenge of dealing with quasi-identifiers was more complicated. We opted for a framework of *k*-anonymity.[20] A dataset is *k*-anonymous if any one individual in the dataset cannot be distinguished from at least *k*–1 other individuals in the same dataset. This requires ensuring that no individual has a combination of quasi-identifiers different from *k*–1 others. If a dataset cannot meet these requirements, then the data must be modified to meet *k*-anonymity, either by generalizing data within cases or suppressing entire cases. For example, if a single student in the dataset is from Latvia, two remedies exist: we can generalize her location by reporting her as from "Europe" rather than Latvia; we can suppress her location information; or we can suppress her case entirely.

This begins to illustrate the fundamental tension between generating datasets that meet the requirements of anonymity mandates and advancing the science of learning through public releases of data. Protecting student privacy under the current regulatory regime requires modifying data to ensure individual students cannot be identified. These modifications can, however, change the dataset considerably, raising serious questions about the utility of the open data for replication or novel analysis. Here, we describe our approach to generating a *k*-anonymous dataset, and then examine the consequences of our modifications to the size and nature of the dataset.

As with open source code and openly licensed content, support for open data has been steadily building.

De-Identification Methods

The dataset we wished to release was a "person-course" dataset, meaning each row represents one course registration for one person (a person with three course registrations will have three rows in the dataset). The original dataset contained:

- information about students (username, IP address, country, self-reported level of education, self-reported year of birth, and self-reported gender);
- the course ID (a string identifying the institution, semester, and course);
- information about student activity in the course (date and time of first interaction, date and time of last interaction, number of days active, number of chapters viewed, number of events recorded by the edX platform, number of video play events, number of forum posts, and final course grade); and
- four variables we computed to indicate level of course involvement (registered: enrolled in the course; viewed: interacted with the courseware at least once; explored: interacted with content from more than 50% of course chapters; and certified: earned a passing grade and received a certificate).

Transforming this person-course dataset into a *k*-anonymous dataset we believed met FERPA guidelines required four steps: defining identifiers and quasi-identifiers, defining the value for *k*, removing identifiers, and modifying or deleting values of quasi-identifiers from the dataset in a way that ensures *k*-anonymity while minimizing changes to the dataset.

We defined two variables in the original dataset as identifiers and six variables as quasi-identifiers. The username was considered identifying in and of itself, so we replaced it with a random ID. IP address was also removed. Four student demographic variables were defined as quasi-identifiers: country, gender, age, and level of education. Course ID was considered a quasi-identifier since students might take unique combinations of courses and because it provides a link between PII posted in forums and the person-course dataset. The number of forum posts made by a student was also a quasi-identifier because a determined individual could scrape the content of the forums from the archived courses and then identify users with unique numbers of forum posts.

Once the quasi-identifiers were chosen, we had to determine a value of *k* to use for implementing *k*-anonymity. In general, larger values of *k* require greater changes to de-identify, and smaller values of *k* leave datasets more vulnerable to re-identification. The U.S. Department of Education offers guidance to the de-identification process in a variety of contexts, but it does not recommend or require specific values of *k* for specific contexts. In one FAQ, the Department's Privacy

Technical Assistance Center states that many "statisticians consider a cell size of 3 to be the absolute minimum" and goes on to say that values of 5 to 10 are even safer.[15] We chose a *k* of five for our de-identification.

Since our dataset contained registrations for 16 courses, registrations in multiple courses could be used for re-identification. The *k*-anonymity approach would ensure no individual was uniquely identifiable using the quasi-identifiers *within* a course, but further care had to be taken in order to remove the possibility that a registrant could be uniquely identified based upon registering in a unique combination or number of courses. For example, if only three people registered for all 16 courses, then those three registrants would not be *k*-anonymous across courses, and some of their registration records would need to be suppressed in order to lower the risk of their re-identification.

The key part of the de-identification process was modifying the data such that no combination of quasi-identifiers described groups of students smaller than five. The two tools employed for this task were generalization and suppression. *Generalization* is the combining of more granular values into categories (for example, 1, 2, 3, 4, and 5 become "1–5"), and *suppression* is the deletion of data that compromises *k*-anonymity.[21] Many strategies for de-identification, including Sweeney's Datafly algorithm, implement both tools with different amounts of emphasis on one technique or the other.[18] More generalization would mean fewer records are suppressed, but the remaining records would be less specific than the original data. A heavier reliance on suppression would remove more records from the data, but the remaining records would be less altered.

Here, we illustrate differential trade-offs between valid research inferences and de-identification methods by comparing two de-identification approaches: one that favors generalization over suppression (hereafter referred to as the Generalization Emphasis, or GE, method), and one that favors suppression over generalization (hereafter referred to as the Suppression Emphasis, or SE, method). There are other ways to approach the problem of de-identification, but these were two that were easily implemented. Our intent is not to discern the dominance of one technique over the other in any general case but rather to show that trade-offs between anonymity and valid research inferences a) are unavoidable and b) will depend on the method of de-identification.

The key part of the de-identification process was modifying the data such that no combination of quasi-identifiers described groups of students smaller than five.

The Suppression Emphasis (SE) method used generalization for the names of countries (grouping them into continent/region names for countries with fewer than 5,000 rows) and for the first event and last event time stamps (grouping them into dates by truncating the hour and minute portion of the time stamps). Suppression was then employed for rows that were not *k*-anonymous across the quasi-identifying variables. For more information on the specifics of the implementation, please refer to the documentation accompanying the data release.[10]

The Generalization Emphasis (GE) method generalized year of birth into groups of two (for example, 1980–1981), and number of forum posts into groups of five for values greater than 10 (for example, 11–15). Suppression was then employed for rows that were not *k*-anonymous across the quasi-identifying variables. The generalizations resulted in a dataset that needed less suppression than in the SE method, but also reduced the precision of the generalized variables.

Both de-identification processes are more likely to suppress registrants in smaller courses: the smaller a course, the higher the chances that any given combination of demographics would not be *k*-anonymous, and the more likely this row would need to be suppressed. Furthermore, since an activity variable (number of forum posts) was included as a quasi-identifier, both methods were likely to remove users who were more active in the forums. Since only 8% of students had any posts in the forums at all, and since these students were typically active in other ways, the records of many of the most active students were suppressed.

The Consequences of Two Approaches to De-Identification

Both of the de-identified datasets differ from the original dataset in substantial ways. We reproduced analyses conducted on the original dataset and evaluated the magnitude of changes in the new datasets. Those differences are highlighted here.

Both de-identified datasets are substantially smaller than the original dataset (see Table 1), but de-identification did not affect enrollment numbers uniformly across courses. Table 1 shows the percentage decrease of enrollment in each de-identified dataset compared to the original file. Only a small percentage of records from CS50x were removed because CS50x was hosted off the edX platform, and so we have no data about forum usage (one of our quasi-identifying variables).

Table 2 shows that de-identification has a disproportionate impact on the most active students. Ho et al.[6] identified four mutually exclusive categories of students: Only Registered enrolled in the course but did not interact with the courseware; Only Viewed interacted with at least one, and fewer than half, of the course chapters; Only Explored interacted with content from half or more of the course chapters but did not earn a certificate; and Certified earned a certificate in the course. In Table 2, we see that the proportions of students in each category seem to change only slightly after de-identification; however, the percentage of certified students in the de-identified dataset is nearly half the percentage in the original dataset. Given the policy concerns around MOOC certification rates, this is a substantially important difference, even if only a small change in percentage points.

Demographic data from the de-identified datasets was similar to the original person-course dataset. Table 3 shows the distributions of gender and bachelor's degree attainment, respectively, for each dataset. The proportions of bachelor's degree holders in all three datasets are nearly identical. The de-identified datasets report slightly lower percentages of female students than the original dataset. The gender bias of MOOCs is a sensitive policy issue, so this difference raises concerns about analyses conducted with the de-identified datasets.

The suppression of highly active users substantially reduces the median number of total events in the courseware. Table 3 shows the median events for all three datasets, and the de-identified datasets have median event values that are two-thirds of the value reported by the original dataset.

Finally, we analyzed the correlations among variables in all three of the datasets. We use correlations to illustrate possible changes in predictive models that rely on correlation and covariance matrices, from the regression-based prediction of grades to principal components analyses and other multivariate methods. Although straight changes in correlations are dependent on base rates, and averages of correlations are not well formed, we present these simple statistics here for ease of interpretation. No correlation changed direction, and all remain sig-

Table 1. Percent decrease in records by course and by de-identification method.

Institution	Course Code	Baseline N	GE Reduction	SE Reduction	Average Reduction
HarvardX	CS50x	181,410	4%	6%	5%
MITx	6.002x	51,394	15%	21%	18%
MITx	6.00x	72,920	15%	21%	18%
MITx	6.00x	84,511	16%	21%	18%
MITx	6.002x	29,050	17%	23%	20%
MITx	8.02x	41,037	17%	24%	21%
HarvardX	PH278x	53,335	18%	26%	22%
HarvardX	ER22x	79,750	21%	28%	25%
MITx	14.73x	39,759	22%	30%	26%
HarvardX	CB22x	43,555	23%	31%	27%
HarvardX	PH207x	61,170	25%	32%	28%
MITx	3.091x	24,493	33%	42%	37%
MITx	8.MReV	16,787	33%	44%	38%
MITx	7.00x	37,997	35%	45%	40%
MITx	3.091x	12,276	39%	50%	44%
MITx	2.01x	12,243	44%	54%	49%
Total		841,687	18%	24%	21%

Table 2. Percent decrease in records by activity category and by de-identification method.

Activity Category	Baseline N	Baseline Percentage	GE Percentage	SE Percentage	GE Change	SE Change
Only Registered	292,852	34.8%	37.3%	37.6%	+2.5%	+2.8%
Only Viewed	469,702	55.8%	56.2%	56.1%	+0.4%	+0.3%
Only Explored	35,937	4.3%	3.6%	3.5%	–0.7%	–0.7%
Certified	43,196	5.1%	2.9%	2.8%	–2.2%	–2.4%
Total	841,687	100%	100%	100%		
MITx	8.02x	41,037	17%	24%	24%	21%
Total		841,687	18%	24%	24%	21%

Table 3. Changes in demographics and activity by de-identification method.

Statistic	Baseline	GE	SE	GE % Change	SE % Change
Percent Bachelor's or Higher	63%	63%	63%	0.1%	–0.2%
Percent Female	29%	26%	26%	–2.2%	–2.9%
Median Number of Events (explored + certified)	3645	2194	2052	–40%	–44%
MITx	8.02x	41,037	17%	24%	21%

nificant at the 0.05 level. For all registrants, the SE dataset reported correlations marginally closer to the original dataset than the GE method, while for explored and certified students only, the GE dataset was slightly closer to the original (see Table 4).

It is possible to use the results from the previous tables to formulate a multivariate model that has population parameters in these tables. By generating data from such a model in proportion to the numbers we have in the baseline dataset, we would enable researchers to replicate the correlations and mean values above. However, such a model would lead to distorted results for any analysis that is not implied by the multivariate model we select. In addition, the unusual distributions we see in MOOC data[2] would be difficult to model using conventional distributional forms.

The comparisons presented here between the de-identified datasets and the original dataset provide evidence for the tension between protecting anonymity and releasing useful data. We emphasize the differences identified here are not those that may be most concerning. These analyses characterize the difference that researchers conducting replication studies might expect to see. For novel analyses that have yet to be performed on the data, it is difficult to formulate an a priori estimate of the impact of de-identification. For researchers hoping to use de-identified, public datasets to advance research, this means that any given finding might be the result of perturbations from de-identification.

Better Options for Science and Privacy with Respect to MOOC Data

As illustrated in the previous section, the differences between the de-identified dataset and the original data range from small changes in the proportion of various demographic categories to large decreases in activity variables and certification rates. It is quite possible that analyses not yet thought of would yield even more dramatic differences between the two datasets. Even if a de-identification method is found that maintains many of the observed research results from the original dataset, there can be no guarantee that other analyses will not have been corrupted by de-identification.

At this point it may be possible to take for granted that any standard for de-identification will increase over time. Information is becoming more accessible, and researchers are increasingly sophisticated and creative about possible re-identification strategies. Cynthia Dwork, in a presentation on "big data and privacy" sponsored by MIT and the White House in early 2014, pointed out that de-identification efforts have been progressing as a sort of arms race, similar to advances in the field of cryptography.[4] Although *k*-anonymity is a useful heuristic, researchers have challenged that it alone is not sufficient. Machanavajjhala et al.[9] point out that a *k*-anonymous dataset is still vulnerable to a "homogeneity attack." If, after undergoing a process that ensures *k*-anonymity, there exists a group of size *k* or larger for whom the value of a sensitive variable is homogenous (that is, all members of the group have the same value), then the value of that sensitive variable is effectively disclosed even if the attacker does not

Table 4. Changes in Pearson Correlations by de-identification method and activity category.

Variable 1	Variable 2	Registrants	Baseline Correlation	GE Correlation	SE Correlation	GE Change (+/–)	SE Change (+/–)
Grade	Number of days active	All	0.800	0.750	0.745	–0.050	–0.055
Grade	Number of days active	Explored + Certified	0.553	0.558	0.564	+0.005	+0.011
Grade	Number of events	All	0.722	0.701	0.697	–0.021	–0.025
Grade	Number of events	Explored + Certified	0.458	0.495	0.501	+0.037	+0.043
Grade	Number of forum posts	All	0.146	0.064	0.156	–0.082	+0.010
Grade	Number of forum posts	Explored + Certified	0.074	0.036	0.108	–0.038	+0.034
Grade	Number of video plays	All	0.396	0.397	0.403	+0.001	+0.007
Grade	Number of video plays	Explored + Certified	0.159	0.194	0.189	+0.035	+0.030
Number of events	Number of days active	All	0.844	0.837	0.835	–0.007	–0.009
Number of events	Number of days active	Explored + Certified	0.736	0.773	0.776	+0.037	+0.040
Number of events	Number of video plays	All	0.665	0.698	0.714	+0.033	+0.049
Number of events	Number of video plays	Explored + Certified	0.587	0.628	0.634	+0.041	+0.047
Number of forum posts	Number of days active	All	0.207	0.104	0.207	–0.103	+0.000
Number of forum posts	Number of days active	Explored + Certified	0.180	0.103	0.200	–0.077	+0.020
Number of forum posts	Number of events	All	0.287	0.117	0.194	–0.170	–0.093
Number of forum posts	Number of events	Explored + Certified	0.279	0.113	0.176	–0.166	–0.103
Number of forum posts	Number of video plays	All	0.091	0.035	0.100	–0.056	+0.009
Number of forum posts	Number of video plays	Explored + Certified	0.051	0.014	0.050	–0.037	–0.001
Number of video plays	Number of days active	All	0.474	0.492	0.505	+0.018	+0.031
Number of video plays	Number of days active	Explored + Certified	0.311	0.404	0.407	+0.093	+0.096
Average		All	0.463	0.420	0.456	–0.044	–0.008
Average		Explored + Certified	0.339	0.332	0.361	–0.007	+0.022

know exactly which record belongs to the target. Machanavajjhala et al. define this principle as *l*-diversity. Other researchers have advanced an alphabet soup of critiques to *k*-anonymity such as *m*-invariance and *t*-similarity.[4] Even if it were possible to devise a de-identification method that did not impact statistical analysis, it could quickly become outmoded by advances in re-identification techniques.

This example of our efforts to de-identify a simple set of student data—a tiny fraction of the granular event logs available from the edX platform—reveals a conflict between open data, the replicability of results, and the potential for novel analyses on one hand, and the anonymity of research subjects on the other. This tension extends beyond MOOC data to much of social science data, but the challenge is acute in educational research because FERPA conflates anonymity—and therefore de-identification—with privacy. One conclusion could be this data is too sensitive to share; so if de-identification has too large an impact on the integrity of a dataset, then the data should not be shared. We believe this is an undesirable position, because the few researchers privileged enough to have access to the data would then be working in a bubble where few of their peers have the ability to challenge or augment their findings. Such limits would, at best, slow down the advancement of knowledge. At worst, these limits would prevent groundbreaking research from ever being conducted.

Neither abandoning open data nor loosening student privacy protections are wise options. Rather, the research community should vigorously pursue technology and policy solutions to the tension between open data and privacy.

A promising technological solution is differential privacy.[3] Under the framework of differential privacy, the original data is maintained, but raw PII is not accessed by the researcher. Instead, they reside in a secure database that has the ability to answer questions about the data. A researcher can submit a model—a regression equation, for example—to the database, and the regression coefficients and R-squared are returned. Differential privacy has challenges of its own, and remains an open research question because implementing such a system would require carefully crafting limits around the number and specificity of questions that can be asked in order to prevent identification of subjects. For example, no answer could be returned if it drew upon fewer than *k* rows, where *k* is the same minimum cell size used in *k*-anonymity.

Policy changes may be more feasible in the short term. An approach suggested by the U.S. President's Council of Advisors on Science and Technology (PCAST) is to accept that anonymization is an obsolete tactic made increasingly difficult by advances in data mining and big data.[14] PCAST recommends that privacy policy emphasize the use of data should not compromise privacy, and should focus "on the 'what' rather than the 'how.'"[14] One can imagine a system whereby researchers accessing an open dataset would agree to use the data only to pursue particular ends, such as research, and not to contact subjects for commercial purposes or to rerelease the data. Such a policy would need to be accompanied by provisions for enforcement and audits, and the creation of practicable systems for enforcement is, admittedly, no small feat.

We propose that privacy can be upheld by researchers bound to an ethical and legal framework, *even if* these researchers can identify individuals and all of their actions. If we want to have high-quality social science research and privacy of human subjects, we must eventually have trust in researchers. Otherwise, we will always have a strict trade-off between anonymity and science.

Related articles on queue.acm.org

Four Billion Little Brothers? Privacy, mobile phones, and ubiquitous data collection
Katie Shilton
http://queue.acm.org/detail.cfm?id=1597790

Communications Surveillance: Privacy and Security at Risk
Whitfield Diffie, Susan Landau
http://queue.acm.org/detail.cfm?id=1613130

Modeling People and Places with Internet Photo Collections
David Crandall, Noah Snavely
http://queue.acm.org/detail.cfm?id=2212756

References

1. Allen, I. E. and Seaman, J. Grade change: Tracking online education in the United States, 2014; http://sloanconsortium.org/publications/survey/grade-change-2013.
2. DeBoer, J., Ho, A.D., Stump, G.S., and Breslow, L. Changing "course:" Reconceptualizing educational variables for Massive Open Online Courses. Educational Researcher. Published online (2013) before print Feb. 7, 2014.
3. Dwork, C. Differential privacy. Automata, languages and programming. Springer Berlin Heidelberg, 2006, 1–12.
4. Dwork, C. State of the Art of Privacy Protection [PowerPoint slides], 2014; http://web.mit.edu/bigdata-priv/agenda.html.
5. Goben, A. and Salo, D. Federal research data requirements set to change. *College & Research Libraries News 74*, 8 (2013), 421–425; http://crln.acrl.org/content/74/8/421.full.
6. Ho, A.D., Reich, J., Nesterko, S., Seaton, D. T., Mullaney, T., Waldo, J. and Chuang, I. HarvardX and MITx: The First Year of Open Online Courses, Fall 2012–Summer 2013; http://ssrn.com/abstract=2381263
7. Holdren, J.P. Increasing access to the results of federally funded scientific research; http://www.whitehouse.gov/sites/default/files/microsites/ostp/ostp_public_access_memo_2013.pdf.
8. Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B. and Stamper, J. A data repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining*. C. Romero, S. Ventura, M. Pechenizkiy, R.SJ.d Baker, eds. CRC Press, Boca Raton, FL. 2010.
9. Machanavajjhala, A., Gehrke, J., Kifer, D. and Venkitasubramaniam, M. l-diversity: Privacy beyond *k*-anonymity. *ACM Trans. Knowledge Discovery from Data 1*,1 (2007), 3.
10. MITx and HarvardX. HarvardX-MITx person-course academic year 2013 de-identified dataset, version 2.0. http://dx.doi.org/10.7910/DVN/26147.
11. MOOCs @ Illinois. FAQ for Faculty, Feb. 7., 2013; http://mooc.illinois.edu/resources/faqfaculty/
12. National Institutes of Health. Final NIH statement on sharing research data, 2003; http://grants.nih.gov/grants/guide/notice-files/NOT-OD-03-032.html.
13. Newman, J. and Oh, S. 8 things you should know about MOOCs. *The Chronicle of Higher Education* (June 13, 2014); http://chronicle.com/article/8-Things-You-Should-Know-About/146901/.
14. President's Council of Advisors on Science and Technology. Big data and privacy: A technological perspective, 2014; http://www.whitehouse.gov/sites/default/files/microsites/ostp/PCAST/pcast_big_data_and_privacy_-_may_2014.pdf.
15. Privacy Technical Assistance Center. Frequently asked questions—disclosure avoidance, Oct. 2012; http://ptac.ed.gov/sites/default/files/FAQs_disclosure_avoidance.pdf.
16. Siemens, G. Supporting and promoting learning analytics research. *Journal of Learning Analytics 1*, 1 (2014), 3–5; http://epress.lib.uts.edu.au/journals/index.php/JLA/article/view/3908/4010.
17. Skopek, J.M. Anonymity, the production of goods, and institutional design. *Fordham Law Review 82*, 4 (2014), 1751–1809; http://ir.lawnet.fordham.edu/flr/vol82/iss4/4/.
18. Sweeney, L. Datafly: A system for providing anonymity in medical data. Database Security, XI: Status and Prospects. T. Lin and S. Qian, eds. Elsevier Science, Amsterdam. 1998.
19. Sweeney, L. Simple demographics often identify people uniquely. *Health 671* (2000), 1–34, San Francisco, CA.
20. Sweeney, L. *k*-anonymity: a model for protecting privacy. *Intern. J. on Uncertainty, Fuzziness and Knowledge-based Systems 10*, 5 (2002), 557–570.
21. Sweeney, L. Achieving *k*-anonymity privacy protection using generalization and suppression. *Intern. J. on Uncertainty, Fuzziness and Knowledge-based Systems 10*, 5, (2002), 571–588.
22. U.S. Department of Education. Family Educational Rights and Privacy (Federal Register Vol. 73, No. 237). U.S. Government Printing Office, Washington, D.C., http://www.gpo.gov/fdsys/pkg/FR-2008-12-09/pdf/E8-28864.pdf

This article was prepared by a group of researchers and administrators from MIT and Harvard, who have been working with the data, and policies related to the data, from the MITx and HarvardX MOOCs on the edX platform.

Jon P. Daries, MIT; **Justin Reich**, **Jim Waldo**, **Elise M. Young**, **Jonathan Whittinghill**, and **Andrew Dean Ho** of Harvard University, and **Daniel Thomas Seaton** and **Isaac Chuang**, of MIT.

DOI:10.1145/2654847

Defense begins by identifying the targets likely to yield the greatest reward for an attacker's investment.

BY CORMAC HERLEY

Security, Cybercrime, and Scale

A TRADITIONAL THREAT model has been with us since before the dawn of the Internet (see Figure 1). Alice seeks to protect her resources from Mallory, who has a suite of attacks, $k = 0; 1, \ldots ,Q - 1$; now assume, for the moment, (unrealistically) that Q is finite and all attacks are known to both parties. What must Alice do to prevent Mallory gaining access? Clearly, it is sufficient for Alice to block all Q possible attacks. If she does, there is no risk. Further, assuming Mallory will keep trying until he exhausts his attacks (or succeeds), it is also necessary; that is, against a sufficiently motivated attacker, it is both necessary and sufficient that Alice defend against all possible attacks. For many, this is a starting point; for example, Schneider[14] says, "A secure system must defend against all possible attacks, including those unknown to the defender." A popular textbook[13] calls it the "principle of easiest penetration" whereby "An intruder must be expected to use any available means of penetration." An often-repeated quip from Schneier, "The only secure computer in the world is unplugged, encased in concrete, and buried underground," reinforces the view.

How did Mallory meet Alice? How does this scale? That is, how does this model fare if we use it for an Internet-scale population, where, instead of a single Alice, there are many? We might be tempted to say, by extension, that unless each Alice blocks all Q attacks, then some attacker would gain access. However, a moment's reflection shows this cannot always be true. If there are two billion users, it is numerically impossible that each would face the "sufficiently motivated" persistent attacker—our starting assumption; there simply are not two billion attackers or anything close to it. Indeed, if there were two million rather than two billion attackers (making cybercriminals approximately one-third as plentiful as software developers worldwide) users would still outnumber attackers 1,000 to one. Clearly, the threat model in Figure 1 does not scale.

Sufficient ≠ necessary-and-sufficient. The threat model applies to some users and targets but cannot apply to all. When we try to apply it to all we confuse sufficient and "necessary and sufficient." This might appear a quibble, but the logical difference is enormous and leads to absurdities and contradictions when applied at scale.

First, if defending against all attacks is necessary and sufficient, then failure

» key insights

- **A financially motivated attacker faces a severe constraint unrelated to his technical abilities; the average gain minus average cost of an attack must be positive.**
- **Without a cost-effective way of telling profitable targets from unprofitable targets, an attack is no use to a financially motivated attacker.**
- **The difficulty of finding profitable targets is extreme when density is small; a 10x reduction in the density of profitable targets generally results in much more than 10x reduction in economic opportunity for the attacker.**

ILLUSTRATION BY DANIEL HERTZBERG

to do everything is equivalent to doing nothing. The marginal benefit of almost all security measures is thus zero. Lampson[11] expressed it succinctly: "There's no resting place on the road to perfection."

Second, in a regime where everything is necessary, trade-offs are not possible. We have no firm basis on which to make sensible claims (such as keylogging is a bigger threat than shoulder surfing). Those who adhere to a binary model of security are unable to participate constructively in trade-off decisions.

Third, the assumption that there is only a finite number of known attacks is clearly favorable to the defender. In general, it is not possible to enumerate all possible attacks, as the number is growing constantly, and there are very likely to be attacks unknown to the defenders. If failing to do everything is the same as doing nothing (and Alice cannot possibly do everything) the situation appears hopeless.

Finally, the logical inconsistencies are joined by observations that clearly contradict what the model says is necessary. The fact that most users ignore most security precautions and yet escape regular harm is irreconcilable with the threat model of Figure 1. If the model applies to everyone, it is difficult to explain why everyone is not hacked every day.

Modifying the threat model. The threat model of Figure 1 might appear to be a strawman. After all, nobody seriously believes that all effort short of perfection is wasted. It is doubtful that anyone (especially the researchers quoted earlier) adheres to a strictly binary view of security. Rather than insist that the threat model always applies, many use it as a starting point appropriate for some situations but overkill for others. Some modification is generally offered; for example, the popular textbook mentioned earlier[13] codifies this as the "principle of adequate protection," saying "[computer items] must be protected to a degree consistent with their value." A more realistic view is that we start with some variant of the traditional threat model (such as "it is necessary and sufficient to defend against all attacks") but then modify it in some way (such as "the defense effort should be appropriate to the assets").

However, while the first statement is absolute, and involves a clear call to action, the qualifier is vague and imprecise. Of course we cannot defend against everything, but on what basis should we decide what to neglect? It helps little to say the traditional threat model does not always apply unless we specify when it does, and what should be used in its place when it does not. A qualifier that is just a partial and imprecise walkback of the original claim clarifies nothing. Our problem is not that anyone insists on rigid adherence to the traditional threat model, so much as we lack clarity as to when to abandon it and what to take up in its place when we do. Failure to be clear on this point is an unhandled exception in our logic.

This matters. A main reason for elevated interest in computer security is the scale of the population with security needs. A main question for that population is how to get best protection for least effort. It is of the first importance to understand accurately the threats two billion users face and how they should respond. All models may, as it is said, be wrong, but failure to scale, demands of unbounded effort, and inability to handle trade-offs are not tolerable flaws if one seeks to address the question of Internet threats. The rest of this article explores modifications of the traditional threat model.

Figure 1. In a traditional threat model, a single user faces a single attacker; given a sufficiently motivated attacker it is necessary and sufficient to block all attacks.

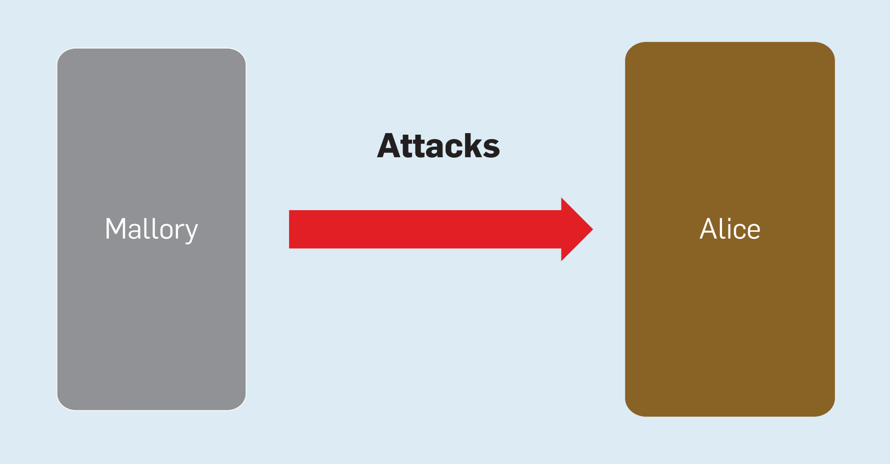
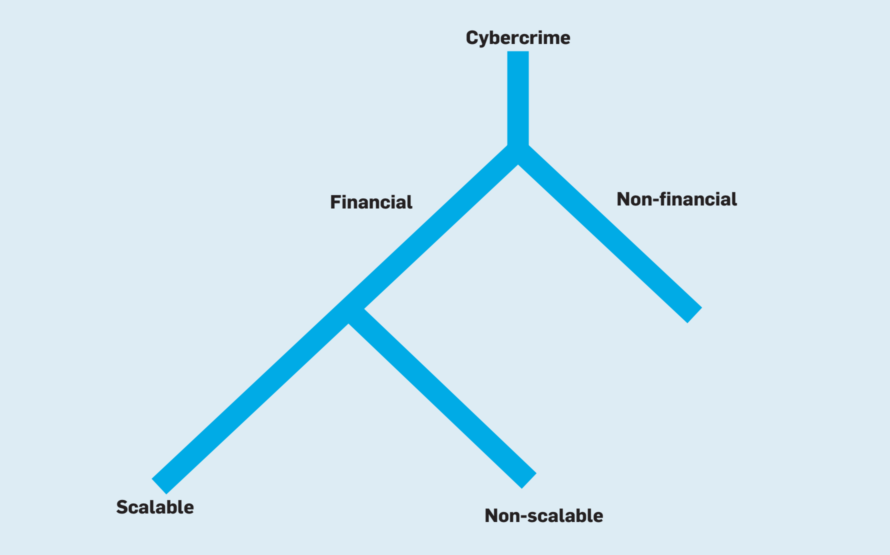


Figure 2. Dividing attacks as financial and nonfinancial; here, financial attacks are further divided into scalable and non-scalable.



Financially Motivated Cybercrime

The threat model of Figure 1 tried to abstract all context away. There is no reference to the value of the resource, the cost of the attack, or how

Mallory came to focus his attention on Alice. The model does not distinguish between finite and infinite gain or between zero and non-zero cost. Abstraction like this is useful. It is far more powerful if we can solve the general problem without resorting to specifics. Unfortunately, the attempt breaks down at scale; the binary view of security must be qualified.

When money is the goal, it seems reasonable to assume Mallory is "sufficiently motivated" when the expected gain from an attack exceeds the cost. I now examine whether focusing on the sub-problem of financially motivated cybercrime will allow progress on the questions of exactly when and how to deviate from the binary model. I propose a bifurcation of attacks (see Figure 2) into those that are financially motivated and those that are not.

Profit at scale. A main reason for concern with cybercrime is the scale of the problem. We might be less concerned if it were a series of one-off or isolated attacks rather than an ongoing problem. Only when the one-time costs can be amortized over many attacks does it become a sustained phenomenon that affects the large online population. To be sustainable there must first be a supply of profitable targets and a way to find them. Hence, the attacker must then do three things: decide who and what to attack, successfully attack, or get access to a resource, and monetize that access.

A particular target is clearly not worthwhile if gain minus cost is not positive: $G - C > 0$. Thus, when attacks are financially motivated, the average gain for each attacker, $E\{G\}$, must be greater than the cost, C:

$$E\{G\} - C > 0. \qquad (1)$$

C must include all costs, including that of finding viable victims and of monetizing access to whatever resources Mallory targets. The gain must be averaged across all attacks, not only the successful ones. If either $E\{G\} \to \infty$ or $C = 0$, then equation (1) represents no constraint at all. When this happens we can revert to the traditional threat model with no need to limit its scope; Alice can neglect no defense if the asset is infinitely valuable or attacks have no cost.

That gain is never infinite needs no demonstration. While it should be equally clear that cost is never precisely zero, it is common to treat cybercrime costs as small enough to neglect. Against this view I present the following arguments: First, if any attack has zero cost, then all targets should be attacked continuously, and all profitable opportunities should be exhausted as soon as they appear. Instead of "Why is there so much spam?," we would ask "Why is there so little?," as it would overwhelm all other traffic. Second, while a script may deliver victims at very low cost, the setup and infrastructure are not free. Even if we grant that a script finds dozens of victims in one day (the Internet is big after all) why should the same script find dozens more the next day, and again the day after? Why should it do so at a sustained rate? Finally, as discussed later, while scripts might achieve access to resources at low cost, the task of monetizing access is generally very difficult. Thus, I argue that not only is attacker cost greater than zero, it is the principal brake on attacker effort.

Attacks that scale. While I have argued that $C > 0$, it is clear that the majority of users are regularly attacked by attacks that involve very low cost per attacked user. I find it useful to segment attacks by how their costs grow. Scalable attacks are one-to-many attacks with the property that cost (per attacked user) grows slower than linearly; for example, doubling the number of users attacked increases the cost very little:[8]

$$C(2N) \ll 2 \cdot C(N). \qquad (2)$$

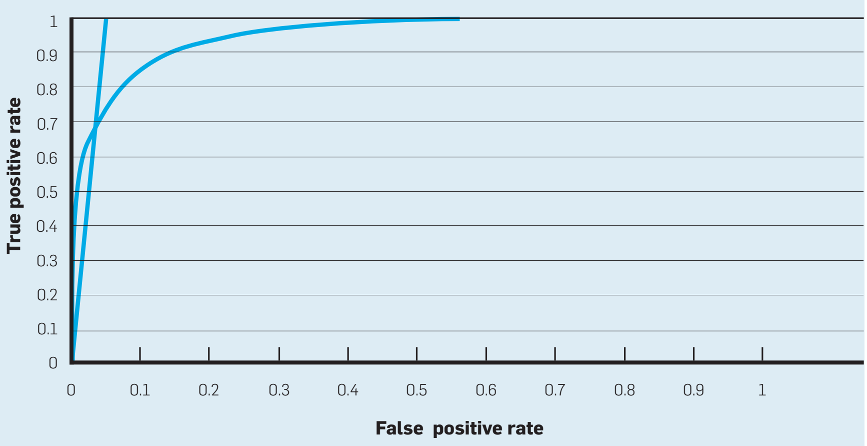
Many of the attacks most commonly seen on the Internet are of this type. Phishing and all attacks for which spam is the spread vector are obvious examples. Viruses and worms that spread wherever they find opportunity are others. Drive-by download attacks (where webpage visitors are attacked through browser vulnerabilities) are yet more. Non-scalable attacks are everything else. In contrast to equation (2), they have costs that are proportional to the number attacked: $C(N) \propto N$. I add the bifurcation, into scalable and non-scalable attacks, to Figure 2.

Constraints on Financially Motivated Attackers

A financially motivated attacker must decide who and what to attack, attack successfully, then monetize access. The better these activities can be scaled the greater the threat he represents to the online population. I now examine some of the difficulties and constraints in scalable answers to these questions.

Scalable attacks (attack everybody). An alternative to solving the problem of deciding whom to attack is to attack everyone. Scalable attacks have inherent advantages over non-scalable attacks. They reach large masses at very low cost, and techniques can be propagated easily—advantages that come with severe constraints, however. Scalable attacks are highly visible; in reaching millions going unnoticed is

Figure 3. Example ROC curve with line of slope *T*/*d* = 20. Only operating points to the left of this line satisfy equation (5) and yield profit. As *T*/*d* increases, the true positive rate falls, and fewer viable targets are attacked; for example, with this classifier, when *T*/*d* = 10⁴, less than 1% of viable targets will be attacked.



difficult. Their broadcast nature gives an alert, to both defenders and other would-be attackers. This attracts competition and increases defense efforts.

Scalable attacks are a minority of attack types. It is the exception rather than the rule that costs have only weak dependence on the number attacked. Anything that cannot be automated completely or involves per-target effort is thus non-scalable, as this cost violates the constraint defined in equation (2). Physical side-channel attacks (requiring proximity) are out, as getting close to one million users costs a lot more than getting close to one. Labor-intensive social engineering attacks (such as those described by Mitnick[12]) and the "stuck in London" scam are non-scalable. After an initial scalable spam campaign, the Nigerian 419 scam (and variants) devolves into a non-scalable effort in manipulation. Equally, spear-phishing attacks that make use of information about the target are non-scalable. While the success rate on well-researched spear-phishing attacks may be much higher than the scatter-shot (such as "Dear Paypal customer") approaches, they are non-scalable. Attacks that involve knowledge of the target are usually non-scalable; for example, guessing passwords based on knowledge of the user's dog's name, favorite sports team, or cartoon character involves significant non-scalable effort. Equally, attacks on backup authentication questions that involve researching where a user went to high school are non-scalable.

While Internet users see evidence of scalable attacks every day, it is actually a minority of attack types that are scalable.

Finding viable targets. Non-scalable attacks resemble the one-on-one attacks of the traditional threat model. However, rather than an attacker who is sufficiently motivated to persist, no matter what, we have one who obeys a profit constraint, as in equation (1). The problem (for Mallory) is that profitability is not directly observable. It is not obvious who will succumb to most attacks and who will prove profitable. Since $C > 0$, the cost of false positives (unprofitable targets) can consume the gain from true positives. When this happens, attacks that are perfectly feasible from a technical standpoint become impossible to run profitably. The cost and difficulty of deciding whom to attack is almost unstudied in the security literature; however, no audit of Mallory's accounts can be complete without it. Unless he has a cost-effective way to identify targets in a large population, non-scalable attacks are of little use to Mallory.

Assume Mallory can estimate a probability, or likelihood, of profit, given everything he observes about a potential target. This is the probability that the target succumbs and access can be monetized (for greater than average cost). Call this $P\{\text{viable}|\text{obs.}\}$. The observables might be address, ZIP code, occupation, and any other factor likely to indicate profitability. Without loss of generality, they can be wrapped into a single one-dimensional sufficient statistic.[15] We assume the cost of gathering the observables is small relative to the cost of the attack. This makes the problem a binary classification,[9] so receiver operator characteristic (ROC) curves are the natural analysis tool; the ROC curve is the graph of true positive rate, t_p, vs. false positive rate, f_p (an example is shown in Figure 3).

Let us now examine how the binary classification constrains Mallory. Suppose, in a population of size N, a fraction $P\{\text{viable}\} = d$ of targets are viable. From Bayes's theorem (when d is small):

$$P\{\text{viable}|\text{obs.}\} = \frac{d}{d + \dfrac{P\{\text{obs.}|\text{ non-viable}\}}{P\{\text{obs.}|\text{viable}\}}\cdot(1-d)}$$

$$\approx d\cdot\frac{P\{\text{obs.}|\text{viable}\}}{P\{\text{obs.}|\text{non-viable}\}}. \qquad (3)$$

$P\{\text{viable}|\text{obs.}\}$ is proportional to density; so the difficulty of finding a viable target gets worse as d falls. A set of observables that gives a 90% chance of finding a viable target when $d = 0.01$ gives only a 0.09% chance when $d = 10–5$. So observables that promise a near "sure thing" at one density offer a worse than 1,000-to-1 long shot at another.

Mallory presumably decides to attack depending on whether or not $P\{\text{viable}|\text{obs.}\}$ is above or below some threshold, T. The threshold T will generally be set by a budget if, say, an attacker needs one attack in every $1/T$ (such as 1-in-20 and 1-in-100) to be profitable. Then, from equation (3) he must have:

$$P\{\text{obs.}|\text{viable}\} \geq \left(\frac{T}{d}\right)\cdot P\{\text{obs.}|\text{ non-viable}\}. \qquad (4)$$

This constraint says the observables must be a factor of T/d more common among viable targets than non-viable. If 1-in-10,000 is viable, and Mallory needs one attack in 20 to succeed, he must identify observable features that are T/d = 500x more common in the viable population than in the non-viable.

The ROC curve gives a geometric interpretation. Mallory finds dt_pN viable targets in $dt_pN + (1 - d)f_pN$ attacks. To satisfy the budget constraint, the ratio of successes to attacks must be greater than T, so (when d is small) we get:

$$\frac{t_p}{f_p} \geq \frac{T}{d}. \qquad (5)$$

Thus, only points (f_p, t_p) on the ROC curve to the left of a line with slope T/d will satisfy Mallory's profit constraint. To illustrate, a line of slope 20 is shown in Figure 3.

Since the slope of the ROC curve is monotonic,[15] as we retreat to the left, t_p/f_p thus increases; equation (5) can almost always be satisfied for some points no matter how good or bad the classifier. However, as we retreat leftward, t_p decreases, so a smaller and smaller fraction of the true positives, or viable targets, are attacked; for example, for the classifier in Figure 3, when $T = 1/10$ (or Mallory needs one attack in 10 to succeed) and $d = 10−5$ (or one in 100,000 is viable), Mallory requires $t_p/f_p \geq 104$, which happens only for values $t_p < 0.01$, meaning less than 1% of the viable population is observably profitable. As d decreases, Mallory ends up with a shrinking fraction of a pool that is itself shrinking.[9] Without a very good classifier (with t_p high while keeping f_p low), most viable victims escape harm.

It is easy to underestimate the difficulty of building good classifiers. Real-world examples from other domains illustrate that this is non-trivial; for example, the false positive rate for mammograms is $t_p \approx 0{:}94$ at $f_p \approx 0{:}065$ (so $t_p/f_p \approx 14.5$).[4] For appendectomies it is $t_p \approx 0.814$ at $f_p \approx 0{:}105$ (so $t_p/f_p \approx 7.8$).[7] Even with the benefits of decades of effort and millions of examples of both true and false positives, building a classifier is often extremely difficult. This is especially true when the base-rate of sought items is low. When *d* is small, Mallory faces a seemingly intractable Catch-22; he must find victims in order to figure out how they can be found. Determining how viable and non-viable can be distinguished requires a large collection of viable targets.

Monetization: Access ≠ dollars. In many forms of non-financial cybercrime the attacker succeeds once he gains access. Often getting the celebrity's password, control of the Web server, or the file of customer records is the end; once he is in he is done. A few screenshots, a decorated webpage, or extruded files suffice if the attacker merely wants acknowledgment. However, for financially motivated attacks, things are different. The attacker is not after passwords or files or access to secure servers as ends in themselves. He wants money and is interested in these things only to the degree they lead to money. Turning access into money is much more difficult than it looks.

For concreteness, consider the assets the Internet's two billion users are trying to protect. Consider bank passwords first. It might seem that once an attacker gets a bank password that money follows quickly. However, several factors indicate this is not the case: First, most transactions in the banking system are reversible; when fraud is discovered they are rolled back.[5] It is for this reason that bank fraud often requires money mules, who (often unwittingly) accept reversible transfers from a compromised account and send on irreversible transfers (such as by Western Union). A money mule can be used no more than once or twice before transactions begin to bounce. While stealing passwords may be easy, and scalable, the limiting factor in the password-stealing business is thus mule recruitment.[5] This view also explains anecdotal accounts that the asking price for stolen credentials in underground markets is fractions of a penny on the dollar.

The situation with other account types is typically worse. Attempts to monetize access to social-networking passwords generally involve the well-known, labor-intensive "stuck in London" scam. Email accounts often receive password reset links for other accounts. However, even when a bank password can be reset, this is simply an indirect path to a resource we already found problematic.

When resources are finite, the question is not whether trade-offs will be made, but how.

Other consumer assets also seem challenging. It may be possible to compromise users' machines by getting them to click on a malicious link. However, even with arbitrary code running on the machine, monetization is far from simple. All passwords on a machine can be harvested, but we have seen that only a minority of stolen bank passwords can be monetized and most nonbank passwords are worthless. The machine can be used to send spam, but the return on spam-based advertising campaigns is low.[10] A botnet responsible for one-third of the world's spam in 2010 apparently earned its owners $2.7 million.[1] A machine can be used to host malicious content. However, as an argument for monetization, this logic is circular, suggesting how yet more machines can be infected, rather than how the original or subsequent machines can be monetized. Scareware, or fake anti-virus software, appears to be one of the better prospects. Successfully compromised boxes can be sold; a pay-per-install market reportedly pays on the order of $100 to $180 per thousand machines in developed markets.[2] Ransomware offers another possible strategy but works best against those who do not practice good backup regimes. For a financially motivated attacker, bank passwords seem to be the best of the consumer-controlled assets, though that best is not very good.

Popular accounts often paint a picture of easy billions to be made from cybercrime, though a growing body of work contradicts this view. Widely circulated estimates of cybercrime losses turn out to be based on bad statistics and off by orders of magnitude.[6] The most detailed examination of spam puts the global revenue earned by all spammers at tens of millions of dollars per year.[10] In a meta-analysis of available data, Anderson et al.[1] estimated global revenue from the stranded traveler and fake anti-virus scams at $10 million and $97 million respectively. The scarcity of monetization strategies is illustrated by the fact that porn-dialers (which incur high long-distance charges), popular in the days of dial-up-modem access, have resurfaced in mobile-phone malware. It would be wrong to conclude there is no money in cybercrime. It appears to be a profitable endeavor for some, but the pool of money to be shared seems much smaller than is often assumed. It is likely that for those who specialize in infrastructure, selling services to those downstream, capture much of the value.

The difficulty of monetization appears to be not clearly understood. The idea that attacks resulting in non-financial harm might have been worse is quite common. The journalist Mat Honan of *Wired* magazine, whose digital life was erased but who suffered no direct financial loss, said, "Yet still I was actually quite fortunate. They could have used my email accounts to gain access to my online banking, or financial services." This is almost certainly wrong. His attackers, after several hours of effort, gained access to a Twitter account and an iTunes account and wiped several devices. While exceedingly inconvenient for the victim, anyone attempting to monetize these accomplishments would likely be disappointed.

Discussion

Scalability is not a "nice to have" feature. Today's widespread interest in computer security seems a result of scale. Scale offers several things that work in the attacker's favor. A potential victim pool that could not be imagined by criminals in 1990 is now available. Further, a huge online population means that even attacks with very low success rates will have significant pools of victims; if only one in a million believes an offer of easy money from a Nigerian prince, there are still 2,000 in the online population.

However, a large pool helps only if there is some way to attack it. Scalable attacks can reach vast populations but fall into only a few limited categories. Non-scalable attacks face a different problem. While the number of viable victims in even a niche opportunity may be large, the difficulty of finding them is related to their relative frequency, not their absolute number. In this case, while the attack itself is non-scalable, Mallory still needs a low-cost way to accurately identify the good prospects in a vast population.

Trade-offs are not optional. When resources are finite, the question is not whether trade-offs will be made but how. For defenders, a main problem with the traditional threat model is it offers no guidance whatsoever as to how it can be done. Most acknowledge that defending against everything is neither possible nor appropriate. Yet without a way to decide which attacks to neglect, defensive effort will be assigned haphazardly.

Most assets escape exploitation not because they are impregnable but because they are not targeted.

We are unlikely to be able to defeat unconstrained attackers, who, according to Pfleeger and Pfleeger,[13] "can (and will) use any means they can" with bounded effort. Recall, however, that most assets escape exploitation not because they are impregnable but because they are not targeted. This happens not at random but predictably when the expected monetization value is less than the cost of the attack. We propose understanding target selection and monetization constraints is necessary if we are to make the unavoidable trade-offs in a systematic way.

Which attacks can be neglected? As before, I concentrate on attacks that are financially motivated, where expected gain is greater than cost. Scalable attacks represent an easy case. Their ability to reach vast populations means no one is unaffected. They leave a large footprint so are not difficult to detect, and there is seldom much mystery as to whether or not an attack is scalable. In the question of trade-offs it is difficult to make the case that scalable attacks are good candidates to be ignored. Fortunately, they fall into a small number of types and have serious restrictions, as we saw earlier. Everyone needs to defend against them.

Non-scalable attacks present our opportunity; it is here we must look for candidates to ignore. Cybercriminals probably do most damage with attacks they can repeat and for which they can reliably find and monetize targets. I suggest probable harm to the population as a basis for prioritizing attacks. First, attacks where viable and non-viable targets cannot be distinguished pose the least economic threat. If viability is entirely unobservable, then Mallory can do no better than attack at random. Second, when the density of viable victims is small T/d becomes very large, and the fraction of the viable population that is attacked shrinks to nothing, or $t_p \rightarrow 0$. This suggests non-scalable attacks with low densities are smaller threats than those where it is high. Finally, the more difficult an attack is to monetize the smaller the threat it poses.

Examples of attacks with low densities might be physical side-channel attacks that allow an attacker in close proximity to the target to shoulder surf and spy on the output on a screen or printer or the input to a keyboard. The viable target density would be the fraction of all LCD screens, printers, and keyboards whose output (or input) can be successfully attacked and monetized for greater reward than the cost of the attack. It seems safe to say this fraction should be very small, perhaps $d = 10^{-5}$ or so. It is also unclear how they might be identified. Hence, an attacker who needs one success in every 20 attacks must operate to the left of a line with slope $T/d = 5{,}000$ on the ROC curve. Those who can accomplish this might consider abandoning cybercrime and trying information retrieval and machine learning. Examples of resources that are difficult to monetize are low-value assets (such as email messages and social networking accounts); while these occasionally lead to gain, the average value appears quite low.

Analysis of the observability, density, and monetization of attacks will never be perfect. To some degree, judgments must be retroactive. That errors will be made seems unavoidable; however, since we are unable to defend against everything, attacks for which the evidence of success is clear must take priority over those for which it is not. When categories of targets (such as small businesses in the U.S.) or categories of attacks (such as spear phishing email messages) are clearly being profitably exploited, additional countermeasures are warranted.

What should we do differently? There are also possible directions for research. The hardness of the binary classification problem suggests unexplored defense mechanisms. Any linear cost component makes it impossible to satisfy equation (2). Imposing a small charge has been suggested as a means of combatting spam,[3] and it is worth considering whether it might be applicable to other scalable attacks. Address space layout randomization similarly converts scalable attacks to non-scalable. Relatively unexplored is the question of how to make the classification problem even more difficult. That is, Mallory has a great sensitivity to the density of viable targets. By creating phantom targets that look plausibly viable but which in fact are not, we make his problem even more difficult; for example, phantom online banking accounts that do nothing but consume attacker effort might reduce the profitability of brute-forcing. When non-viable targets reply to scam email messages it reduces return and makes it more difficult to make a profit.[9]

I have repeatedly stressed that an attacker must choose targets, successfully attack, and then monetize the success. The second of these problems has dominated the research effort. However, if the admonition "Think like an attacker" is not empty, we should pay equal attention to how attackers can select targets and monetize resources. I have pointed out the theoretical difficulty of the binary-classification problem represented by target selection. Yet for a profit-seeking attacker the problem is not abstract. It is not enough to hopefully suggest that some ZIP codes, employers, or professions might be indicative of greater viability than others. The attacker needs concrete observable features to estimate viability. If he does not get it right often enough, or does not satisfy equation (4), he makes a loss. What is observable to attackers about the population is also observable to us. The problem of how viable niches for a particular attack can be identified is worth serious research. If they can be identified, members of these niches (rather than the whole population) are those who must invest extra in the defense. If they cannot, it is difficult to justify spending on defense. I reiterate that I have focused on financially motivated attacks. An interesting research question would be which types of target are most at risk of non-financial attacks.

Conclusion

When we ignore attacker constraints, we make things more difficult than they need to be for defenders. This is a luxury we cannot afford. The view of the world that says every target must block every attack is clearly wasteful, and most of us understand it is neither possible nor necessary. Yet acknowledging this fact is helpful only if we are clear about which attacks can be neglected. The contradiction between the traditional model, which says trade-offs are not possible, and reality, which says they are necessary, must be resolved. I propose the difficulties of profitably finding targets and monetizing them are underutilized tools in the effort to help users avoid harm.

Acknowledgments

I thank Rainer Böhme, Joe Bonneau, Shuo Chen, Alain Forget, Rob Reeder, Adam Shostack, and the anonymous reviewers for comments that greatly improved this article. C

References

1. Anderson, R., Barton, C., Böhme, R., Clayton, R., van Eeten, M. J.G., Levi, M, Moore, T., and Savage, S. Measuring the cost of cybercrime. In *Proceedings of the 11th Annual Workshop on the Economics of Information Security* (Berlin, June 25–26, 2012).
2. Caballero, J., Grier, C., Kreibich, C., and Paxson, V. Measuring pay-per-install: The commoditization of malware distribution. In *Proceedings of the USENIX Security Symposium*. USENIX Association, Berkeley, CA, 2011.
3. Dwork, C. and Naor, M. Pricing via processing or combatting junk mail. In *Proceedings of Crypto 1992*.
4. Elmore, J.G., Barton, M.B., Moceri, V.M., Polk, S., Arena, P.J., and Fletcher, S.W. Ten-year risk of false positive screening mammograms and clinical breast examinations. *New England Journal of Medicine 338*, 16 (1998), 1089–1096.
5. Florêncio, D. and Herley, C. Is everything we know about password-stealing wrong? *IEEE Security & Privacy Magazine* (Nov. 2012).
6. Florêncio, D. and Herley, C. Sex, lies and cyber-crime surveys. In *Proceedings of the 10th Workshop on Economics of Information Security* (Fairfax, VA, June 14–15, 2011).
7. Graff, L., Russell, J., Seashore, J., Tate, J., Elwell, A., Prete, M., Werdmann, M., Maag, R., Krivenko, C., and Radford, M. False-negative and false-positive errors in abdominal pain evaluation failure to diagnose acute appendicitis and unnecessary surgery. *Academic Emergency Medicine 7*, 11 (2000), 1244–1255.
8. Herley, C. The plight of the targeted attacker in a world of scale. In *Proceedings of the Ninth Workshop on the Economics of Information Security* (Boston, June 7–8, 2010).
9. Herley, C. Why do Nigerian scammers say they are from Nigeria? In *Proceedings of the 11th Annual Workshop on the Economics of Information Security* (Berlin, June 25–26, 2012).
10. Kanich, C., Weaver, N., McCoy, D., Halvorson, T., Kreibich, C., Levchenko, K., Paxson, V., Voelker, G.M., and Savage, S. Show me the money: Characterizing spam-advertised revenue. In *Proceedings of the 20th USENIX Security Symposium* (San Francisco, Aug. 8–12). USENIX Association, Berkeley, CA, 2011.
11. Lampson, B. Usable security: How to get it. *Commun. ACM 52*, 11 (Nov. 2009), 25–27.
12. Mitnick, K. and Simon, W.L. *The Art of Deception: Controlling the Human Element of Security.* John Wiley & Sons, Inc., New York, 2003.
13. Pfleeger, C.P. and Pfleeger, S.L. *Security In Computing.* Prentice Hall Professional, 2003.
14. Schneider, F. Blueprint for a science of cybersecurity. *The Next Wave 19*, 2 (2012), 47–57.
15. van Trees, H.L. *Detection, Estimation and Modulation Theory: Part I.* John Wiley & Sons, Inc., New York, 1968.

Cormac Herley (cormac@microsoft.com) is a principal researcher in the Machine Learning Department of Microsoft Research, Redmond, WA.

DOI:10.1145/2629612

The unknown and the invisible exploit the unwary and the uninformed for illicit financial gain and reputation damage.

BY MICHAIL TSIKERDEKIS AND SHERALI ZEADALLY

Online Deception in Social Media

PROLIFERATION OF WEB-BASED technologies has revolutionized the way content is generated and exchanged through the Internet, leading to proliferation of social-media applications and services. Social media enable creation and exchange of user-generated content and design of a range of Internet-based applications. This growth is fueled not only by more services but also by the rate of their adoption by users. From 2005 to 2013, users and developers alike saw a 64% increase in the number of people using social media;[1] for instance, Twitter use increased 10% from 2010 to 2013, and 1.2 billion users connected in 2013 through Facebook and Twitter accounts.[24] However, the ease of getting an account also makes it easy for individuals to deceive one another. Previous work on deception found that people in general lie routinely, and several efforts have sought to detect and understand deception.[20] Deception has been used in various contexts throughout human history (such as in World War II and the Trojan War) to enhance attackers' tactics. Social media provide new environments and technologies for potential deceivers. There are many examples of people being deceived through social media, with some suffering devastating consequences to their personal lives.

Here, we consider deception as a deliberate act intended to mislead others, while targets are not aware or do not expect such acts might be taking place and where the deceiver aims to transfer a false belief to the deceived.[2,9] This view is particularly relevant when examining social media services where the boundary between protecting one's privacy and deceiving others is not morally clear. Moreover, such false beliefs are communicated verbally and non-verbally,[14] with deception identifiable through cues, including verbal (such as audio and text), non-verbal (such as body movement), and physiological (such as heartbeat).

Training and raising awareness (such as might be taught to security personnel[17]) could help protect users of social media. However, people trained to detect deception sometimes perform worse in detection accuracy than people who are not trained,[17] and evidence of a "privacy paradox" points to individuals sharing detailed information, even though they are aware of privacy concerns,[26] making them more vulnerable to attack. Making things worse, social media, as a set of Internet-based applications, can be broadly defined as including multiple virtual environments.[15,16]

» key insights

- **In social media, deception can involve content, sender, and communication channel or all three together.**
- **The nature of a social medium can influence the likelihood of deception and its success for each deception technique.**
- **Deception detection and prevention are complicated by lack of standard online deception detection, of a computationally efficient method for detecting deception in large online communities, and of social media developers looking to prevent deception.**

PHOTOGRAPH BY NOMAD SOUL

Exploring deception in social media, we focus on motivations and techniques used and their effect on potential targets, as well as on some of the challenges that need to be addressed to help potential targets detect deception. While detecting and preventing deception are important aspects of social awareness relating to deception, understanding online deception and classifying techniques used in social media is the first step toward sharpening one's defenses.

Online Deception

Nature often favors deception as a mechanism for gaining a strategic advantage in all kinds of biological relationships; for example, viceroy butterflies deceive birds by looking like monarch butterflies (which have a bitter taste), ensuring their survival as long as there are not too many in a particular area.[8] Similarly, humans have long used deception against fellow humans.[3] In warfare, Chinese military strategist and philosopher Sun Tzu[29] famously said, "All warfare is based on deception."

Social media services are generally classified based on social presence/media richness and self-representation/self-disclosure.[16] Social presence can also be influenced by the intimacy and immediacy of the medium in which communication takes place; media richness describes the amount of information that can be transmitted at a given moment. Self-representation determines the control users have representing themselves, whereas self-disclosure defines whether one reveals information, willingly or unwillingly. Using these characteristics, Kaplan and Haenlein[16] developed a table including multiple aspects of social media: blogs, collaborative projects (such as Wikipedia), social networking sites (such as Facebook), content communities (such as YouTube), virtual social worlds (such as Second Life), and virtual game worlds (such as World of Warcraft). Table 1 outlines an expanded classification of social media that also includes microblogging (such as Twitter) and social news sites (such as Reddit). We categorize microblogging between blogs and social networking sites[15] and social news sites above microblogging, given their similarity to microblogging in terms of social presence/media richness (limited content communicated through the medium and average immediacy as news comes in) and their low self-presentation/self-disclosure due to their nature as content-oriented communities.

Social media provide an environment in which assessment signals are neither required nor the norm, making deception easy; for instance, gender switching online may require only a name change.

Social media that give users freedom to define themselves are in the second row of Table 1, and social media that force users to adapt to certain roles or have no option for disclosing parts of their identities are in the first row. Moreover, along with increased media richness and social presence, we note a transition from social media using just text for communication to rich media simulating the real world through verbal and non-verbal signals, as well as greater immediacy in virtual game worlds and virtual social communication. The differences between these types of social media affect how deception is implemented and its usefulness in deceiving fellow users.

In most social media platforms, communication is generally text-based and asynchronous, giving deceivers an advantage for altering content—an inexpensive way to deceive others. Zahavi[31] identified the difference between assessment signals that are reliable and difficult to fake and conventional signals that are easier to fake; for example, in the real world, if older people want to pass as younger, they might dress differently or dye their hair to produce conventional signals. However, it would be much more difficult to fake a driver's license or other authentic documentation. But social media provide an environment in which assessment signals are neither required nor the norm, making deception easy; for instance, gender switching online may require only a name change.

Difficulty Perpetrating Online Deception

The level of difficulty perpetrating online deception is determined by several factors associated with the deceiver, the social media service, the deceptive act, and the potential victim. Significant difficulty could deter potential deceivers, and lack of difficulty may be seen as an opportunity to deceive others (see Figure 1).

The deceiver. Several factors associated with deceivers determine the difficulty of trying to perpetrate online deception, including expectations, goals, motivations, relationship with the target, and the target's degree of suspicion.[2] Expectation is a factor that determines the likelihood of success in deception. More complex messages have a greater likelihood of being communicated.[20] Goals and motivations also determine the difficulty of perpetrating a deception. Goals are broader and longer term, and motivations consist of specific short-term objectives that directly influence the choice and type of deception. A taxonomy developed by Buller and Burgoon[2] described three motivators for deception: "instrumental," where the would-be deceiver can identify goal-oriented deception (such as lying about one's résumé on a social medium to increase the likelihood of more job offers); "relational," or social capital (such as aiming to preserve social relationships typical in online social networks);[26] and "identity" (such as preserving one's reputation from shameful events in an online profile). These motivators in turn determine the cost or level of difficulty to deceivers in trying to deceive; for example, deceivers motivated to fake their identity must exert more effort offline due to the presence of signals much more difficult to fake than online where many identity-based clues (such as gender and age) may take the form of conventional signals (such as adding information to one's profile page without verification). Difficulty perpetrating a deception is also determined by the deceiver's relationship to a target. Familiarity with a target and the target's close social network make it easier to gain trust and reduce the difficulty of perpetrating deception. Many users assume enhanced security comes with technology so are more likely to trust others online.[4] Moreover, the level of trust individuals afford a deceiver also reduces their suspicion toward the deceiver, thereby increasing the likelihood of being deceived.

Moral cost also increases the difficulty of perpetrating deception.[26] Moral values and feelings can influence what deceivers view as immoral in withholding information or even lying. In the real world, the immediacy of interaction may make it much more difficult to deceive for some individuals. In contrast, in the online world, distance and anonymity[28] contribute to a loss of inhibition; the moral cost is thus lower for deceivers.

Social media. Social media require potential targets and would-be deceivers alike to expand their perspective on how interactions are viewed between receiver and sender during deception; for instance, "interpersonal deception theory"[2] says the interaction between a sender and a receiver is a game of iterative scanning and adjustment to ensure deception success.

Donath[8] suggested that if deception is prevalent in a system (such as Facebook) then the likelihood of successful deception is reduced. It makes sense that the prevalence of deception in an online community is a factor that also determines difficulty perpetrating deception. Social media services that encounter too much deception will inevitably yield communities that are more suspicious. Such community suspicion will increase the number of failed attempts at deception. Moreover, increasing a potential target's suspicion will likewise increase the difficulty, thereby deterring deceivers from entering the community in the first place, though some equilibrium may eventually be reached. However, this rationale suggests communities without much deception are likely more vulnerable to attacks since suspicion by potential victims is low. Determining the prevalence of deception in a community is a challenge.

Similarly, the underlying software design of social media can also affect the degree of suspicion; the level of perceived security by potential victims increases the likelihood of success for would-be deceivers.[11] Software design can cause users to make several assumptions about the level of security being provided. Some aspects of the design can make them more relaxed and less aware of the potential signs of being deceived; for example, potential

Table 1. Social media classifications.

		Social presence/Media richness			
		Low		High	
Self-presentation/Self-disclosure	Low	Collaborative projects	Social news sites	Content communities	Virtual game worlds
	High	Blogs	Microblogging	Social networking sites	Virtual social worlds

Figure 1. Entities and participants involved in online deception.

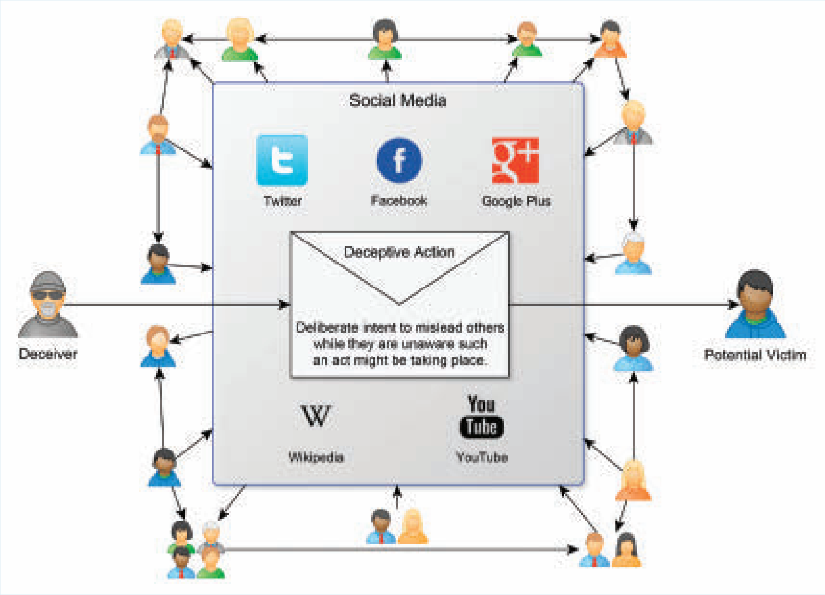
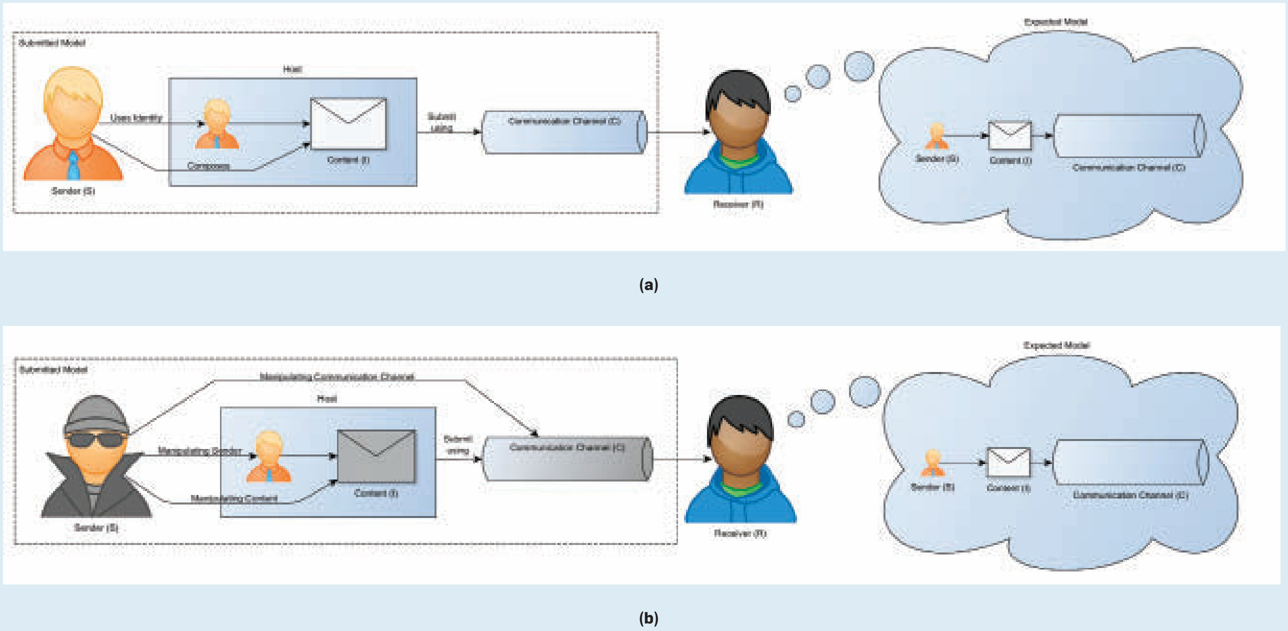


Figure 2. Interaction without and with deception.



(a)

(b)

targets may falsely assume that faking profile information on a social networking site is difficult due to multiple verification methods (such as email confirmation). Moreover, a system's assurance and trust mechanisms determine the level of trust between sender and receiver.[11] Assurance mechanisms can either reduce the probability of successful deception or increase the penalty for deceivers.[11] A tough penalty means increased difficulty for deceivers, especially when the chances of being caught are high. Assurance mechanisms are considered effective in certain contexts where the need for trust may be completely diminished. In social media, assurance mechanisms are much more difficult to implement, penalties and the chances of being caught may be or seem to be lower than those in offline settings, and the cost of deception is much lower. Media richness is another factor determining difficulty perpetrating deception. In this context, Galanxhi and Nah[10] found deceivers in cyberspace feel more stress when communicating with their victims through text rather than through avatar-supported chat.

Deceptive acts. Time constraints and the number of targets also help determine the difficulty perpetrating online deception. The time available and the time required for a successful attack are important, especially in social media services involving asynchronous communication. Moreover, the time required for deception to be detected also determines the effectiveness of the deception method being used. For instances where deception must never be discovered, the cost of implementing a deception method may outweigh any potential benefit, especially when the penalty is high. The social space in which deception is applied and the number of online user targets who are to be deceived help determine the level of difficulty implementing a deception method; for example, in the case of politicians trying to deceive through their online social media profiles, all potential voters face a more difficult challenge deciding how to vote compared to deceivers targeting just a single voter. Type of deception is another important factor. Complex deceptive acts motivated by multiple objectives (such as faking an identity to manipulate targets into actions that serve the deceiver's goals) are more difficult to perpetrate.

Potential victim. In real-world offline settings, the potential target's ability to detect deception may be a factor determining the difficulty perpetrating deception; for example, in a 2000 study of Internet fraud using page-jacking techniques, even experienced users of social media failed to detect inconsistencies, except for a select few who did detect it, thus showing detection is not impossible.[11] In social media, the potential targets' ability to detect deception also depends to some extent on their literacy in information communication technology. Deceivers must therefore evaluate the technology literacy of their potential victims. Users with high technology literacy have a significant advantage over casual Internet users, so the cost to a deceiver as calculated through a cost-benefit analysis for a social engineering attack may be higher.

Deception Techniques

Various techniques are reported in the literature for deceiving others in social media environments, including bluffs, mimicry (such as mimicking a website), fakery (such as establishing a fake website), white lies, evasions, exaggeration, webpage redirections (such as misleading someone to a false profile page), and concealment (such as withholding information from one's profile).[21] We use the communication model proposed by Madhusudan[20] to classify deception techniques for social media and evaluate their effectiveness in achieving deception.

Deception model. The model (see Figure 2) consists of a sender (S), the content or message (I), the channel through which communication takes place (C), and the receiver (R). If a receiver's expected model (the so-called SIC triangle) is different from the received model (any or all SIC elements have been altered) then deception has occurred. This is also in line with Ekman's definition[9] of deception, saying a receiver cannot anticipate deception for deception to be considered deception. Deception is perpetrated by manipulating any of the SIC elements or any combination thereof. We present in the following paragraphs an overview of social media and identify factors and social-media types where deception can be perpetrated with minimal effort at low cost, resulting in a fairly high deception success rate (see Table 2). We identified these factors from the literature.

Content deception. Manipulating content, as in falsifying information, is presumably the most common way to deceive others. Social media that focus primarily on content (such as blogs, microblogs, content communities, and social news sites) are highly susceptible to such deception. Technology allows anyone with access privileges (legitimate and illegitimate) to manipulate multimedia files to an extraordinary degree. Tampering with images[23] is an effective way to fake content (such as representing that one traveled around the world through one's photos, altering them and sharing them through social media). Such a scheme may help deceivers elevate their social status and win a victim's trust to obtain further information. In addition to videos and images, the ease of manipulating content that is at times based on text alone yields low-cost deception and high probability of success due to the targets' low information literacy and lack of expectation for verifiability and even accountability. In addition, social media (such as social network sites and virtual social worlds) offering profile management for users are also susceptible, especially when advertising emphasizes the promise of new relationships. Competent deceivers may thus have a substantial advantage.

Collaborative projects (such as Wikipedia) are less likely to be affected by deception, or manipulating (I). The difficulty in perpetrating deception may seem low, but the likelihood of success (at least over the long term) is also low. This trade-off is due to the software design of these types of social media, where many-to-many communication enables many people to see the content. We see examples of content deception in Wikipedia, where not only vandals (people altering content with intent to deceive others) are eventually detected but other people assume a role in fighting them.[25] Furthermore, assurance mechanisms (such as a requirement for content validity, tracing content back to its source) are built into the system to ensure content deception is more apparent. Another example of content deception in social media involves open source software managed by multiple users where it is much more difficult to add malicious content and perpetrate a deception because multiple individuals evaluate the code before it is released. Virtual game worlds also have low probability for deception due to strongly narrated elements (such as being assigned specific roles that force players to follow a specific course of action).

Sender deception. Sender deception is achieved by manipulating the sender's identity information (S). Impersonation is a common example, resulting in identity deception, or identity theft.[30] Deceivers may gain access to an identity and use it to obtain additional information from their peers (such as home address, date of birth, and cellphone number). Failure to authenticate the sender's credentials yields deception. Social media's designs with built-in high self-presentation and self-disclosure enable low-cost sender deception. Blogs and microblogging can lead to stolen identities, as no control mechanisms are in place to verify new users or their associated names. However, the damage caused by deception with these types of social media is also likely to remain fairly light, and long-term deceiver success is probably not guaranteed. Authentic-identity owners may become aware of the theft, and other individuals familiar with that identity may start identifying behavioral cues that do not match it. In the case of social network sites and virtual social worlds, the cost of deception increases because users must behave and communicate in ways that are appropriate to the identity they impersonate. The benefits are indeed much greater in a social medium because access to a user's personal social network can lead to enhanced ability to win other people's trust within the network and obtain information from them. The target in these cases may not necessarily be the individual whose identity is stolen but others within that person's social network. With no control mechanisms in place for identifying a source, unregistered individuals without an account may be more exposed than registered users.

Social media (such as collaborative projects and virtual game worlds) with limited self-presentation and self-disclosure are likely to be more protected in terms of identity theft, due, in part, to their intended function. Collaborative projects, content communities, and virtual game worlds are heavily task-based, or contain a fixed narrative from which social behavior is not allowed to

Table 2. Manipulation of sender's identity information (S), content (I), and communication channel (C) with low difficulty and high deception success results.

Social media	Low difficulty	High deception success
Blogs	S, I	S, I
Collaborative projects	I	—
Microblogging	S, I	S, I
Social news sites	S, I	S, I
Social networking sites	S, I, C	S, I, C
Content communities	I	I
Virtual social worlds	S, I, C	S, I, C
Virtual game worlds	I, C	C

deviate. Users who want to gain access to the impersonated identity's social network must perform just as well as the identity being impersonated and "act the part." The cost to a deceiver is likely to be great, and the success of the deception low and short term.

Middle ground between content deception and sender deception involves manipulating information associated with an identity. Such attacks can be categorized as "identity concealment," where part of the information for an original identity is concealed or altered, and identity forgery, where a new identity is formed;[30] for example, would-be deceivers may try to fake some of the information in their profiles to win trust or represent themselves in a different way. In customer social network sites, would-be deceivers may try to conceal information to gain advantage when negotiating to buy or trade something.[5]

Communication-channel deception. Manipulating a communication channel requires greater technical skill, thus increasing the cost of deception. Such manipulation includes modifying in-transit messages, rerouting traffic, and eavesdropping. Jamming communications have been used in virtual game worlds. Podhradsky et al.[22] found multiplayer games in consoles can be hacked to provide access to a user's IP address. Would-be deceivers who gain access to the host can kick the player out and proceed with identity-theft deception. The deceiver's goal may not be to obtain information but to damage the victim's reputation. Worth pointing out is there is a fine line between an unintentional disconnection and an intentional departure of a player in a video game. This line is blurred when the player is on the losing side and leaves suddenly. As a result, the player's reliability and reputation are damaged by the invisible, anonymous deceiver. One advantage of communication-channel deception is the implicit assumption social media users make that digital technology is imperfect and things may not work as well as they do in the real world. However, nonverbal behavior[14] (such as body movement and speech patterns) can expose deceivers through social media by, say, introducing jitter or delays in their video or audio to conceal their deception, effectively increasing the likelihood of success. Victims at the other end of the connection find it difficult to differentiate an unreliable or slow connection from a deceptive act.

Since channel deception generally involves technology, all social media services may be susceptible to attack, especially those using similar technologies or architectures. Services that rely more on their client applications are more prone to attack, while those that rely on server applications are probably safer. Services with high media richness (such as virtual social worlds and virtual game worlds) tend to rely on client software. By exploiting communication channels, deception is common in such services.[13] Server-side applications (such as social networking sites and content communities) are less prone to channel deception because exploits rely on vulnerabilities of Web browsers and Web servers that are generally more secure. The cost of this deception is high, though the likelihood of success is also high, especially for a well-orchestrated attack.

Hybrid deception techniques. Hybrid deception techniques involve manipulation of multiple elements in the SIC model outlined earlier and can be more effective in launching deception attacks. The relationships among S, I, and C, as described by Madhusudan,[20] produce a consistent view for a potential victim. If one element of the SIC model shows a slightly different behavior, it may give clues about an inconsistent relationship between two elements (such as S and I); for example, a message received and signed by a target's relative may lose its credibility if the source information of the message does not match that of the relative.

Various hybrid deception techniques that manipulate a sender's information have been reported in the literature, including forgery,[20] phishing, identity forgery, Web forgery,[11] and email fraud. They are highly effective in social media (such as social-networking sites, virtual social worlds, microblogging, and blogs) that highlight user identity and provide one-to-one or one-to-many communications. These online deception attacks are not only effective but their consequences can lead to disaster, including loss of life. A service initially designed for people who want to initiate new relationships and the lack of verification can lead to a devastating outcome involving psychological or even physical damage to a victim. Online deception can also have financial consequences, as in Web forgery (such as creating websites representing fake businesses), manipulating the content of the sender's communication. Web forgery is relevant for social-media services due to the popularity of including user-developed applications or widgets. Even after internal review mechanisms that detect malicious software, vulnerabilities may still be present unexpectedly in such applications.

Challenges

The costs of deception in social media environments open several technical challenges that developers of social networks, as well as users, must address: lack of a standard, unified theory and methods for online deception detection; lack of a universal or context-specific, computationally efficient method for deception detection in large online communities; and lack of effort by social media developers in deception prevention.

Lack of a standard theory and methods. Several theories concerning online (such as phishing email) and offline environments (such as employment interviews) have been proposed for detecting deception, including "management obfuscation hypothesis," "information manipulation theory," "interpersonal deception theory," "four factor theory," and "leakage theory."[14] All focus on detecting leakage cues deceivers might give away or strategic decisions deceivers make that could reveal deceptive intent. Their main drawback is they rely on a set of verbal and nonverbal cues that may not all apply to the online world; for example, nonverbal cues in some social media communities require deception researchers and site developers to rethink what indicators can be used to recognize them, as they are not likely to exist online in the forms they take in the physical world.

New site-developer focus is required. Steps in that direction are being made with, for example, video blob analysis of hands and movement for detecting movement that is too quick for detection by the human eye (100% multiple state classification accuracy but with a limited sample of only five interviews);[19] detection of image manipulation through inconsistencies in compression artifacts (30%–100%, depending on type of image, compression, and tampering method);[23] machine learning detection using audio and transcribed text to identify patterns that signal deception due to deviations from a baseline (66.4% accuracy, when baseline is at 60.2%);[12] and computerized voice stress analysis to identify variations in an individual's speech patterns (56.8%–92.8% accuracy, depending on context).[6]

One notably promising aspect in social media is that most verbal cues are based on text. Verbal deception detection has been used to identify identity deception (such as through similarity analysis of profile information (80.4%–98.6% accuracy);[30] similarity analysis with natural language processing to identify identity deception through writing patterns (68.8% accuracy);[25] cross-referencing information between a social network and anonymized social networks containing the nodes in the first network to evaluate the trustworthiness of social network profile attributes (40%–80% recall, depending on metric and technique when baseline recall is 20%);[5] and natural language processing to identify text features that betray deceptive email messages (75.4% accuracy).[27] These techniques show options are available for addressing online deception.

However, these techniques do not address all types of online deception for all types of social media; for one thing, there is much variation among social media in terms of design and type and amount of information allowed to be exchanged between users, and it is difficult to determine the context in which optimum accuracy will be achieved for each solution. The field lacks a cohesive framework that captures the interdependencies and interactions among different detection methods, types of deception, and types of social media.

The level of trust individuals afford a deceiver also reduces their suspicion toward the deceiver, thereby increasing the likelihood of being deceived.

Computational efficiency. The techniques being used for deception detection are highly context-specific, and many cannot be applied to the online social media environment. The most popular deception-detection methods dealing with verbal communication include "content-based criteria analysis," "scientific content analysis," and "reality monitoring."[14] Their applicability to social media is unclear. Methods dealing with verbal cues (such as video analysis) may be computationally inefficient.[19] Likewise, methods that aim to detect sender deception (identity deception) and use similarity analyses to match identities may be feasible for small datasets, but a comparison of all records results in a computational time complexity $O(N^2)$. In some contexts where profile information is available and text comparison is possible for features in a profile, the time complexity can be reduced to $O(w'N)$ through an adaptive sorted neighborhood method[30] that sorts a list of records based on profile features, then moves through the records using a window (w) comparing just the records within that window in order to find duplicates. The adaptive method shortens the window (w') by finding the first (if any) duplicate record in a window, then ignores all further comparisons within the window ($w' < w$), drastically increasing the efficiency of the algorithm (1.3 million records parsed in 6.5 minutes).

Similarity analyses are most likely to involve the greatest overhead, especially in social media where datasets tend to be large; scalability is a computational expense for large datasets so require more efficient approaches. For such cases, techniques (such as the "expectancy violations theory," which looks for deviations from a baseline[19]) may be an efficient way to filter suspect cases for further examination. This is a computationally cheaper alternative that can be applied to both sender and content deception; for example, comparing deviations from a normal user baseline requires parsing a database just once, leading to a complexity of $O(N)$.

Finally, methods used in deception detection in social media must account for features of social context (such as friends and family of an individual) that have been found to increase the accuracy of detection of deception.[18] The downside is social network analyses (SNAs) tend to be dramatically more expensive as networks grow. Simple SNA metrics (such as "betweeness centrality") become overwhelmingly difficult to compute as networks grow ($O(N3)$) where N is the number of nodes and more advanced statistical methods (such as exponential random graph models using Markov chain Monte Carlo algorithms) are costly to compute. However, the potential for this newly available social data is apparent, and computational efficiency must be addressed in large social networks. On a positive note, one online trend is formation of small social networking sites[5] and communities for which deception-detection methods may be more computationally feasible.

Deception prevention. Social media application designers must address deception in social media environments; for example, Wikipedia's editing policy requires information added to articles to be cited back to its source and has exposed many baseless arguments to readers. Other social media services must address identity verification; for example, individuals who do not have a Facebook account are paradoxically more likely to fall victim to identity theft (for sensitive information), along with their real-life friends. Friends and other users become wary in the presence of duplicate accounts, especially when a social media account has been active by the original owner of an identity. On the other hand, when a deceiver registers an identity that did not previously exist in a social media service, users are more likely to assume the genuine owner only recently joined the service. In an attempt to increase their user base, social media services, using easy registration and access features, expose unsuspecting users to online deception. An effort to standardize user registration and credential verification must be investigated by government agencies and technical organizations, as elements of everyday life shift to an all-online presence.

Conclusion

Social media keep being extended through a diverse set of tools and technologies available to deceivers. While the physical distance separating a deceiver and a potential target may seem large, the damage that could be done could be enormous. Individuals, organizations, and governments are at risk. Understanding how online deception works through social media is a challenge. To address it, the social media industry must design applications with rules and norms lacking in traditional physical space. Vast numbers of users' desire for innovation and personal connection, as well as romance, has resulted in online designs not yet fully understood, with vulnerabilities exploited by attackers, including those engaging in deception attacks. Researchers, developers, and communities must address how to design social interaction in social-media environments to safeguard and protect users from the consequences of online deception.

Acknowledgments

We thank Elisa Bertino and Moshe Y. Vardi for their encouragement and support throughout the preparation of this article and for the opportunity they gave us to revise and strengthen it. We also thank the anonymous reviewers for their valuable feedback and comments that helped us improve it.

References

1. Brenner, J. and Smith, A. *72% of Online Adults are Social Networking Site Users.* Pew Internet & American Life Project, Washington, D.C., Aug. 5, 2013; http://pewinternet.org/Reports/2013/social-networking-sites.aspx
2. Buller, D.B. and Burgoon, J.K. Interpersonal deception theory. *Communication Theory 6*, 3 (Aug. 1996), 203–242.
3. Burgoon, J., Adkins, M., Kruse, J., Jensen, M.L., Meservy, T., Twitchell, D.P., Deokar, A., Nunamaker, J.F., Lu, S., Tsechpenakis, G., Metaxas, D.N., and Younger, R.E. An approach for intent identification by building on deception detection. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (Big Island, HI, Jan. 3–6). IEEE, New York, 2005.
4. Castelfranchi, C. and Tan, Y-H. The role of trust and deception in virtual societies. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences* (Maui, HI, Jan. 3-6). IEEE, New York, 2001.
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 Networking, Applications and Worksharing* (Pittsburgh, PA, Oct. 14-17). IEEE, New York, 2012, 97–106.
6. Damphousse, K.R., Pointon, L., Upchurch, D., and Moore, R.K. *Assessing the Validity of Voice Stress Analysis Tools in a Jail Setting: Final Report to the U.S. Department of Justice.* Washington, D.C., 2007; http://www.ncjrs.gov/pdffiles1/nij/grants/219031.pdf
7. Dando, C.J. and Bull, R. Maximising opportunities to detect verbal deception: Training police officers to interview tactically. *Journal of Investigative Psychology and Offender Profiling 8*, 2 (July 2011), 189–202.
8. Donath, J.S. Identity and deception in the virtual community. In *Communities in Cyberspace*, M.A. Smith and P. Kollock, Eds. Routledge, New York, 1999, 29–59.
9. Ekman P. Deception, lying, and demeanor. In *States of Mind: American and Post-Soviet Perspectives on Contemporary Issues in Psychology*, D.F. Halpern and A.E. Voiskounsky, Eds. Oxford University Press, New York, 1997, 93–105.
10. Galanxhi, H. and Nah, F.F.-H. Deception in cyberspace: A comparison of text-only vs. avatar-supported medium. *International Journal of Human-Computer Studies 65*, 9 (Sept. 2007), 770–783.
11. Grazioli, S. and Jarvenpaa, S.L. Perils of Internet fraud: An empirical investigation of deception and trust with experienced Internet consumers. *IEEE Transactions on Systems, Man and Cybernetics 30*, 4 (July 2000), 395–410.
12. Hirschberg, J., Benus, S., Brenier, J.M. et al. Distinguishing deceptive from non-deceptive speech. In *Proceedings of the Ninth European Conference on Speech Communication and Technology* (Lisbon, Portugal, Sept. 4–8, 2005), 1833–1836.
13. Hoglund, G. and McGraw, G. *Exploiting Online Games: Cheating Massively Distributed Systems.* Addison-Wesley Professional, Boston, 2007.
14. Humpherys, S.L., Moffitt, K.C., Burns, M.B., Burgoon, J.K., and Felix, W.F. Identification of fraudulent financial statements using linguistic credibility analysis. *Decision Support Systems 50*, 3 (Feb. 2011), 585–594.
15. Kaplan, A.M. and Haenlein, M. The early bird catches the news: Nine things you should know about microblogging. *Business Horizons 54*, 2 (Mar.–Apr. 2011), 105–113.
16. Kaplan, A.M. and Haenlein, M. Users of the world, unite! The challenges and opportunities of social media. *Business Horizons 53*, 1 (Jan.–Feb. 2010), 59–68.
17. Kassin, S. and Fong, C. 'I'm innocent!': Effects of training on judgments of truth and deception in the interrogation room. *Law and Human Behavior 23*, 5 (Oct. 1999), 499–516.
18. Li, J., Wang, G.A., and Chen, H. PRM-based identity matching using social context. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics* (Taipei, June 17–20). IEEE, New York, 2008, 150–155.
19. Lu, S., Tsechpenakis, G., Metaxas, D.N., Jensen, M.L., and Kruse, J. Blob analysis of the head and hands: A method for deception detection. In *Proceedings of the 38th Annual Hawaii International Conference on Systems Sciences* (Big Island, HI, Jan. 3–6). IEEE, New York, 2005.
20. Madhusudan, T. On a text-processing approach to facilitating autonomous deception detection. In *Proceedings of the 36th Annual Hawaii International Conference on Systems Sciences* (Big Island, HI, Jan. 6–9). IEEE, New York, 2003.
21. Nunamaker Jr., J.F. Detection of deception: Collaboration systems and technology. In *Proceedings of the 37th Annual Hawaii International Conference on Systems Sciences* (Big Island, HI, Jan. 5–8). IEEE, New York, 2004.
22. Podhradsky, A., D'Ovidio, R., Engebretson, P., and Casey, C. Xbox 360 hoaxes, social engineering, and gamertag exploits. In *Proceedings of the 46th Annual Hawaii International Conference on Systems Sciences* (Maui, HI, Jan. 7–10). IEEE, New York, 2013, 3239–3250.
23. Popescu, A.C. and Farid, H. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing 53*, 2 (Feb. 2005), 758–767.
24. Shen, X. Security and privacy in mobile social network. *IEEE Network 27*, 5 (Sept.–Oct. 2013), 2–3.
25. Solorio, T., Hasan, R., and Mizan, M. A case study of sockpuppet detection in Wikipedia. In *Proceedings of the Workshop on Language Analysis in Social Media*, A. Farzindar, M. Gamon, M. Nagarajan, D. Inkpen, and C. Danescu-Niculescu-Mizil, Eds. (Atlanta, June 3). Association for Computational Linguistics, Stroudsburg, PA, 2013, 59–68.
26. Squicciarini, A.C. and Griffin, C. An informed model of personal information release in social networking sites. In *Proceedings of the 2012 International Conference on Social Computing* (Amsterdam, Sept. 3–5). IEEE, New York, 2012, 636–645.
27. Stone, A. Natural-language processing for intrusion detection. *Computer 40*, 12 (Dec. 2007), 103–105.
28. Suler, J. The online disinhibition effect. *CyberPsychology & Behavior 7*, 3 (June 2004), 321–326.
29. Tzu, S. *The Art of War* (translated by Samuel B. Griffith). Oxford University Press, New York, 1963.
30. Wang, G.A., Chen, H., Xu, J.J., and Atabakhsh, H. Automatically detecting criminal identity deception: An adaptive detection algorithm. *IEEE Transactions on Systems, Man, and Cybernetics 36*, 5 (Sept. 2006), 988–999.
31. Zahavi, A. The fallacy of conventional signalling. *Philosophical Transactions of the Royal Society of London 340*, 1292 (May 1993), 227–230.

Michail Tsikerdekis (tsikerdekis@uky.edu) is an assistant professor in the College of Communication and Information of the University of Kentucky, Lexington, KY.

Sherali Zeadally (szeadally@uky.edu) is an associate professor in the College of Communication and Information at the University of Kentucky, Lexington, KY.

Previous A.M. Turing Award Recipients

1966	A.J. Perlis
1967	Maurice Wilkes
1968	R.W. Hamming
1969	Marvin Minsky
1970	J.H. Wilkinson
1971	John McCarthy
1972	E.W. Dijkstra
1973	Charles Bachman
1974	Donald Knuth
1975	Allen Newell
1975	Herbert Simon
1976	Michael Rabin
1976	Dana Scott
1977	John Backus
1978	Robert Floyd
1979	Kenneth Iverson
1980	C.A.R Hoare
1981	Edgar Codd
1982	Stephen Cook
1983	Ken Thompson
1983	Dennis Ritchie
1984	Niklaus Wirth
1985	Richard Karp
1986	John Hopcroft
1986	Robert Tarjan
1987	John Cocke
1988	Ivan Sutherland
1989	William Kahan
1990	Fernando Corbató
1991	Robin Milner
1992	Butler Lampson
1993	Juris Hartmanis
1993	Richard Stearns
1994	Edward Feigenbaum
1994	Raj Reddy
1995	Manuel Blum
1996	Amir Pnueli
1997	Douglas Engelbart
1998	James Gray
1999	Frederick Brooks
2000	Andrew Yao
2001	Ole-Johan Dahl
2001	Kristen Nygaard
2002	Leonard Adleman
2002	Ronald Rivest
2002	Adi Shamir
2003	Alan Kay
2004	Vinton Cerf
2004	Robert Kahn
2005	Peter Naur
2006	Frances E. Allen
2007	Edmund Clarke
2007	E. Allen Emerson
2007	Joseph Sifakis
2008	Barbara Liskov
2009	Charles P. Thacker
2010	Leslie G. Valiant
2011	Judea Pearl
2012	Shafi Goldwasser
2012	Silvio Micali
2013	Leslie Lamport

ACM A.M. TURING AWARD NOMINATIONS SOLICITED

Nominations are invited for the 2014 ACM A.M. Turing Award. This, ACM's oldest and most prestigious award, is presented for contributions of a technical nature to the computing community. Although the long-term influences of the nominee's work are taken into consideration, there should be a particular outstanding and trendsetting technical achievement that constitutes the principal claim to the award. The recipient presents an address at an ACM event that will be published in an ACM journal.

Nominations should include:

1) A curriculum vitae, listing publications, patents, honors, and other awards.

2) A letter from the principal nominator, which describes the work of the nominee, and draws particular attention to the contribution which is seen as meriting the award.

3) Supporting letters from at least three endorsers. The letters should not all be from colleagues or co-workers who are closely associated with the nominee, and preferably should come from individuals at more than one organization. Successful Turing Award nominations should include substantive letters of support from prominent individuals broadly representative of the candidate's field or related field impacted by the candidate's contribution.

For additional information on ACM's award program please visit: www.acm.org/awards/

Additional information on the past recipients of the A.M. Turing Award is available on: http://amturing.acm.org/byyear.cfm.

Nominations should be sent electronically by November 30, 2014 to: Barbara Liskov, MIT CSAIL c/o mcguinness@acm.org

acm Association for Computing Machinery

DOI:10.1145/2629535

Exploring the distinction between an optimal robot motion and a robot motion resulting from the application of optimization techniques.

BY JEAN-PAUL LAUMOND, NICOLAS MANSARD, AND JEAN-BERNARD LASSERRE

Optimality in Robot Motion: Optimal versus Optimized Motion

THE FIRST BOOK dedicated to robot motion was published in 1982 with the subtitle "Planning and Control."[5] The distinction between motion planning and motion control has mainly historical roots. Sometimes motion planning refers to geometric path planning, sometimes it refers to open loop control; sometimes motion control refers to open loop control, sometimes it refers to close loop control and stabilization; sometimes planning is considered as an offline process whereas control is real time. From a historical perspective, robot motion planning arose from the ambition to provide robots with motion autonomy: the domain was born in the computer science and artificial intelligence communities.[22] Motion planning is about deciding on the existence of a motion to reach a given goal and computing one if this one exists. Robot motion control arose from manufacturing and the control of manipulators[30] with rapid effective applications in the automotive industry. Motion control aims at transforming a task defined in the robot workspace into a set of control functions defined in the robot motor space: a typical instance of the problem is to find a way for the end-effector of a welding robot to follow a predefined welding line.

What kind of optimality is about in robot motion? Many facets of the question are treated independently in different communities ranging from control and computer science, to numerical analysis and differential geometry, with a large and diverse corpus of methods including, for example, the maximum principle, the applications of Hamilton-Jacobi-Bellman equation, quadratic programming, neural networks, simulated annealing, genetic algorithms, or Bayesian inference. The ultimate goal of these methods is to compute a so-called *optimal* solution whatever the problem is. The objective of this article is not to overview this entire corpus that follows its own routes independently from robotics, but

» key insights

- **Computing an optimal robot motion is a challenging issue illustrated by more than 20 years of research on wheeled mobile robots.**
- **Geometric control theory and numerical analysis highlight two complementary perspectives on optimal robot motion.**
- **Most of the time, robot algorithms aimed at computing an optimal motion provide an *optimized* motion, which is not optimal at all, but is the output of a given optimization method.**
- **When optimal motions exist, numerical algorithms mostly fail in accounting for their combinatorial structure. Moreover, optimization algorithms bypass (not overcome) the question of the existence of optimal motions.**

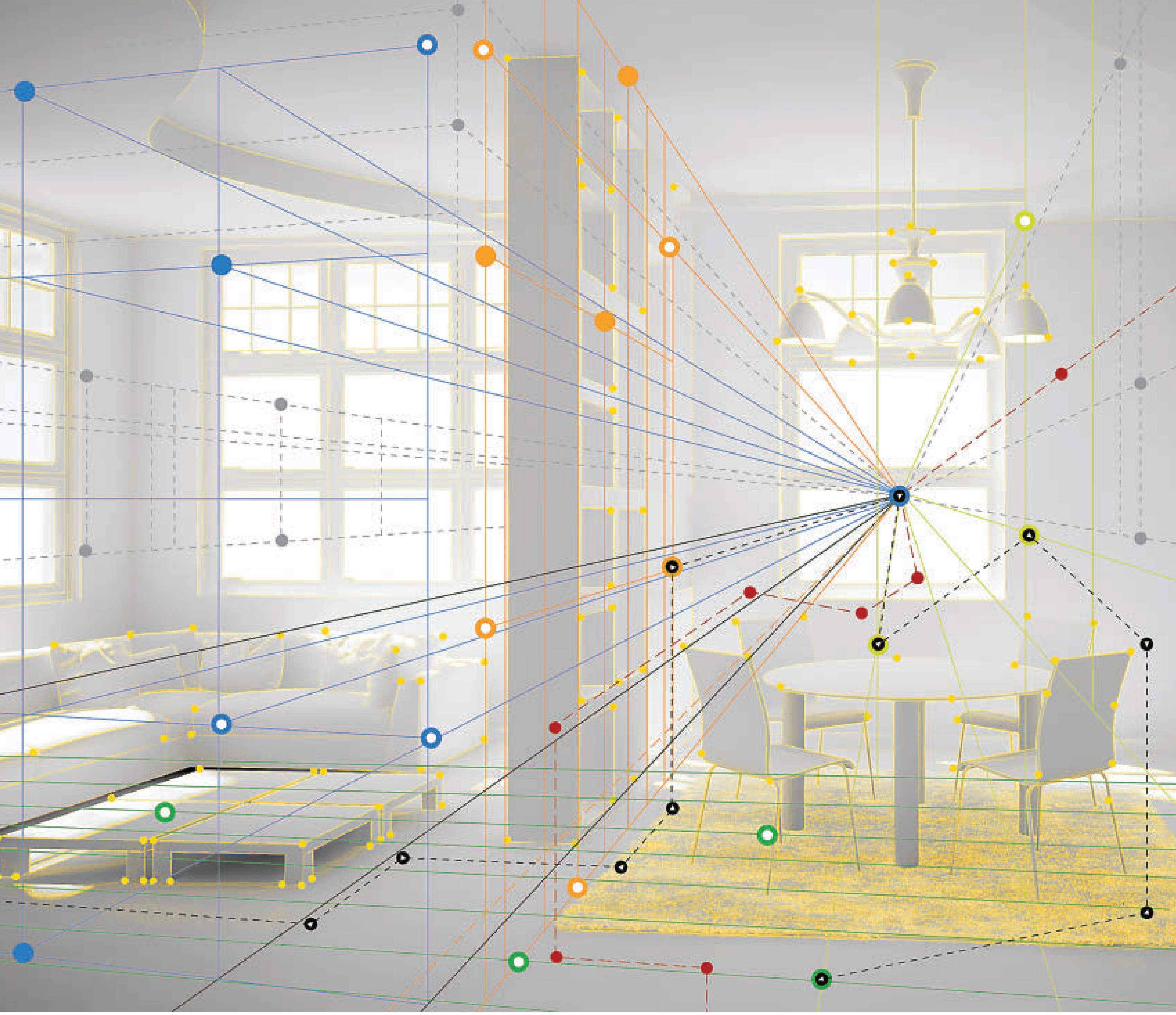


ILLUSTRATION BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

rather to emphasize the distinction between "optimal motion" and "optimized motion." Most of the time, robot algorithms aiming at computing an optimal motion provide in fact an optimized motion that is not optimal at all, but is the output of a given optimization method. Computing an optimal motion is mostly a challenging issue as it can be illustrated by more than 20 years of research on wheeled mobile robots (as we discuss later).

Note that the notion of optimality in robot motion as it is addressed in this article is far from covering all the dimensions of robot motion.[7] It does not account for low-level dynamical control, nor for sensory-motor control, nor for high level cognitive approaches to motion generation (for example, as developed in the context of robot soccer or in task planning).

What Is Optimal in Robot Motion Planning and Control?

Motion planning explores the computational foundations of robot motion by facing the question of the existence of admissible motions for robots moving in an environment populated with obstacles: how to transform the continuous problem into a combinatorial one?

This research topic[22,26] evolved in three main stages. In the early 1980s, Lozano-Perez first transformed the problem of moving bodies in the physical space into a problem of moving a point in some so-called configuration space.[28] In doing so, he initiated a well-defined mathematical problem: planning a robot motion is equivalent to searching for connected components in a configuration space. Schwartz and Sharir then showed the problem is decidable as soon as we can prove that the connected components are semi-algebraic sets.[35] Even if a series of papers from computational geometry explored various instances of the problem, the general "piano mover" problem remains intractable.[14] Finally by relaxing the completeness exigence for the benefit of probabilistic completeness, Barraquand and Latombe introduced a new algorithmic paradigm[3] in the early 1990s that gave rise to the popular probabilistic roadmap[20] and rapid random trees[31] algorithms.

Motion planning solves a point-to-point problem in the configuration space. Whereas the problem is a difficult computational challenge that is well understood, optimal motion planning is a much more difficult challenge. In addition to finding a solution to the planning problem (that is, a path that accounts for collision-avoidance and kinematic constraints if any), optimal motion planning refers to finding a solution that optimizes some criterion. These can be the length, the time or the energy (which are equivalent criteria under some assumption), or more sophisticated ones, as the number of maneuvers to park a car.

In such a context many issues are concerned with optimization:

- For a given system, what are the motions optimizing some criteria? Do such motions exist? The existence of optimal motion may depend either on the presence of obstacles or on the criterion to be optimized.
- When optimal motions exist, are they computable? If so, how complex is their computation? How to relax exactness constraints to compute approximated solutions? We will address the combinatorial structure of the configuration space induced by the presence of obstacles and by the metric to be optimized. Time criterion is also discussed, as are practical approaches to optimize time along a predefined path.
- Apart from finding a feasible solution to a given problem, motion planning also wants to optimize this solution once it has been found. The question is particularly critical for the motions provided by probabilistic algorithms that introduce random detours. The challenge here is to optimize no more in the configuration space of the system, but in the motion space.

In this article, optimal motion planning is understood with the underlying hypothesis that the entire robot environment is known and the optimization criterion is given: the quest is to find a global optimum without considering any practical issue such as model uncertainties or local sensory feedback.

Optimal Motion Existence

Before trying to compute an optimal motion, the first question to ask is about its existence. To give some intuition about the importance of this issue, consider a mobile robot moving among obstacles. For some rightful security reason, the robot cannot touch the obstacles. In mathematical language, the robot must move in an open domain of the configuration space. Yet, an optimal motion to go from one place to another one located behind some obstacle will necessarily touch the obstacle. So this optimal motion is not a valid one. It appears as an ideal motion that cannot be reached. The best we can do is to get a collision-free motion whose length approaches the length of this ideal shortest (but non-admissible) motion. In other words, there is no optimal solution to the corresponding motion planning problem. The question here is of topological nature: combinatorial data structures (for example, visibility graphs) may allow us to compute solutions that are optimal in the closure of the free space, and that are not solutions at all in the open free space.

Even without obstacle, the existence of an optimal motion is far from being guaranteed. In deterministic continuous-time optimal control problems we usually search for a time-dependent control function that optimizes some integral functional over some time interval. Addressing the issue of existence requires us to resort to geometric control theory;[18] for instance, Fillipov's theorem proves the existence of minimum-time trajectories,[a] whereas Prontryagin Maximum Principle (PMP) or Boltyanskii's conditions give respectively necessary and sufficient conditions for a trajectory to be optimal. However it is usually difficult to extract useful information from these tools. If PMP may help to characterize optimal trajectories locally, it generally fails to give their global structure. Later, we show how subtle the question may be in various instances of wheeled mobile robots.

The class of optimal control problems for which the existence of an optimal solution is guaranteed, is limited. The minimum time problems for controllable linear systems with bounded controls belong to this class: optimal solutions exist and optimal controls are of bang-bang type. How-

Figure 1. A modern view of the "piano mover" problem: two characters have to move a shared piano while avoiding surrounding obstacles.

[a] Here and in the following, we treat trajectory and motion as synonyms.

ever, the so-called Fuller problem may arise: it makes the optimal solution not practical at all as it is of bang-bang type with infinitely many switches. Other examples include the famous linear-quadratic-Gaussian problem (the cost is quadratic and the dynamics is linear in both control and state variables), and systems with a bounded input and with a dynamics that is affine in the control variables. In the former a closed loop optimal solution can be computed by solving algebraic Riccati equations, whereas in the latter the existence of an optimal control trajectory is guaranteed under some appropriate assumptions.

In more general cases, we can only hope to approximate as closely as desired the optimal value via a sequence of control trajectories. There is indeed no optimal solution in the too restricted space of considered control functions. This has already been realized since the 1960s. The limit of such a sequence can be given a precise meaning as soon as we enlarge the space of functions under consideration. For instance, in the class of problems in which the control is affine and the integral functional is the L1-norm, the optimal control is a finite series of impulses and not a function of time (for example, see Neustadt[29]). In some problems such as the control of satellites, such a solution makes sense as it can approximately be implemented by gas jets. However, in general, it cannot be implemented because of the physical limitations of the actuators.

Changing the mathematical formulation of the problem (for example, considering a larger space of control candidates) may allow the existence of an optimal solution. In the former case of satellite control, the initial formulation is coherent as an "ideal" impulse solution can be practically approximated by gas jets. However, in other cases the initial problem formulation may be incorrect as an ideal impulse solution is not implementable. Indeed, if we "feel" a smooth optimal solution should exist in the initial function space considered and if in fact it does not exist, then either the dynamics and/or the constraints do not reflect appropriately the physical limitations of the system or the cost functional is not appropriate to guarantee the existence of an optimal solution in that function space. To the best of our knowledge, this issue is rarely discussed in textbooks or courses in optimal control.

Even without an obstacle, the existence of an optimal motion is far from being guaranteed.

Optimal Path Planning

Considering a motion is a continuous function of time in the configuration (or working) space, the image of a motion is a path in that space. The "piano mover" problem refers to the path-planning problem, that is, the geometric instance of robot motion planning (see Figure 1). The constraint of obstacle avoidance is taken into account. In that context, optimality deals with the length of the path without considering time and control. The issue is to find the shortest path between two points.

Depending on the metric that equips the configuration space, a shortest path may be unique (for example, for the Euclidean metric) or not unique (for example, for the Manhattan metric). All configuration space metrics are equivalent from a topological point of view (that is, if there is a sequence of Euclidean paths linking two points, then there is also a sequence of Manhattan paths linking these two points). However, different metrics induce different combinatorial properties in the configuration space. For instance, for a same obstacle arrangement, two points may be linked by a Manhattan collision-free path, while they cannot by a collision-free straight-line segment: both points are mutually visible in a Manhattan metric, while they are not in the Euclidean one. So, according to a given metric, there may or may not exist a finite number of points that "watch" the entire space.[25] These combinatorial issues are particularly critical to devise sampling-based motion planning algorithms.

Now, consider the usual case of a configuration space equipped with a Euclidean metric. Exploring visibility graph data structures easily solves the problem of finding a bound on the length of the shortest path among polygonal obstacles. This is nice, but it is no longer true if we consider three-dimensional spaces populated with polyhedral obstacles. Indeed, finding the shortest path in that case becomes a *NP*-hard problem.[14] So, in general, there is no hope to get an algorithm that computes an optimal path in presence of obstacles, even if the problem of computing an optimal path in the absence of obstacle is solved and even if we allow the piano-robot to touch the obstacles.

As a consequence of such poor results, optimal path planning is usually addressed by means of numerical techniques. Among the most popular ones are the discrete search algorithms operating on bitmap representations of work or configuration spaces.[3] The outputs we only obtain are approximately optimal paths, that is, paths that are "not so far" from a hypothetical (or ideal) estimated optimal path. Another type of methods consists in modeling the obstacles by repulsive potential. In doing so, the goal is expressed by an attractive potential, and the system tends to reach it by following a gradient descent.[21] The solution is only locally optimal. Moreover, the method may get stuck in a local minimum without finding a solution, or that a solution actually exists or not. So it is not complete. Some extensions may be considered. For instance, exploring harmonic potential fields[8] or devising clever navigation functions[34] allow providing globally optimal solutions; unfortunately, these methods require an explicit representation of obstacles in the configuration space, which is information that is generally not available. At this stage, we can see how the presence of obstacles makes optimal path planning a difficult problem.

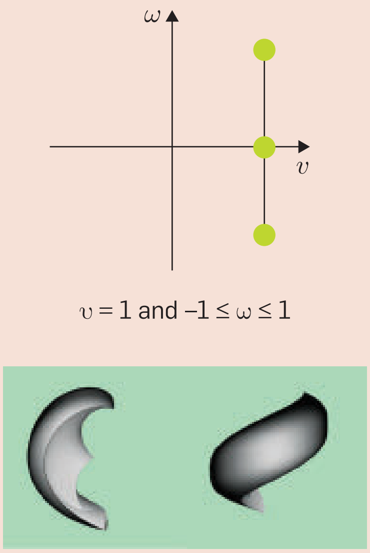
Figure 2. A car (logo of the European Project Esprit 3 PRO-Motion in the 1990s) together with the unicycle model equations.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}cos\theta\\ sin\theta\\ 0\end{bmatrix} v + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix} \omega$$

Optimal Motion Planning

In addition to obstacle avoidance, constraints on robot controls or robot dynamics add another level of difficulties. The goal here is to compute a minimal-time motion that goes from a starting state (configuration and velocity) to a target state while avoiding obstacles and respecting constraints on velocities and acceleration. This is the so-called kinodynamic motion planning problem.[12] The seminal algorithm is based on discretizations of both the state space and the workspace.

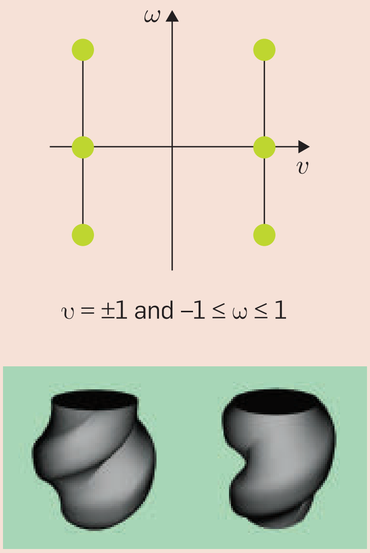
Figure 3. Dubins car.



$\upsilon = 1$ and $-1 \leq \omega \leq 1$

Reachable set in the (x, y, θ)-configuration space

Figure 4. Reeds-Shepp car.



$\upsilon = \pm 1$ and $-1 \leq \omega \leq 1$

Reachable set in the (x, y, θ)-configuration space

It gave rise to many variants including nonuniform discretization, randomized techniques, and extensions of A* algorithms (see LaValle[26]). Today, they are the best algorithms to compute approximately optimal motions.

Less popular in the robot motion planning community are numerical approaches to optimal robot control.[11] Numerical methods to solve optimal control problems fall into three main classes. Dynamic programming implements the Bellman optimality principle saying that any sub-motion of an optimal motion is optimal. This leads to a partial differential equation (the so-called Hamilton-Jacobi-Bellman equation in continuous time) whose solutions may sometimes be computed numerically. However, dynamic programming suffers from the well-known curse of the dimensionality bottleneck. Direct methods constitute a second class. They discretize in time both control and state trajectories so that the initial optimal control problem becomes a standard static non-linear programming (optimization) problem of potentially large size, for which a large variety of methods can be applied. However, local optimality is generally the best one can hope for. Moreover, potential chattering effects may appear hidden in the obtained optimal solution when there is no optimal solution in the initial function space. Finally, in the third category are indirect methods based on optimality conditions provided by the PMP and for which, ultimately, the resulting two-point boundary value problem to solve (for example, by shooting techniques) may be extremely difficult. In addition, the presence of singular arcs requires specialized treatments. So direct methods are usually simpler than indirect ones even though the resulting problems to solve may be very large. Indeed, their structural inherent sparsity can be taken into account efficiently.

At this stage, we can conclude that exact solutions for optimal motion planning remain out of reach. Only numerical approximate solutions are conceivable.

Optimal Motion Planning Along a Path

A pragmatic way to bypass (not overcome) the intrinsic complexity of the kinodynamic and numerical approaches is to introduce a decoupled approach that solves the problem in two stages: first, an (optimal) path planning generates a collision-free-path; then a time-optimal trajectory along the path is computed while taking into account robot dynamics and control constraints. The resulting trajectory is of course not time-optimal in a global sense; it is just the best trajectory for the predefined path. From a computational point of view, the problem is much simpler than the original global one because the search space (named phase plane) is reduced to two dimensions: the curvilinear abscissa along the path and its time-derivative. Many methods have been developed since the introduction of dynamic programming approaches by Shin and McKay[36] in configuration space and simultaneously by Bobrow et al.[4] in the Cartesian space. Many variants have been considered including the improvement by Pfeiffer and Johanni[31] that combines forward and backward integrations, and the recent work by Verscheure et al.[39] who transform the problem into a convex optimization one.

Optimization in Motion Space

Extensions of path-tracking methods may be considered as soon as we allow the deformations of the supporting paths. Here, we assume some motion planner provides a first path (or trajectory). Depending on the motion planner, the path may be far from being optimal. For instance, probabilistic motion planners introduce many useless detours. This is the price to pay for their effectiveness. So, the initial path must be reshaped, that is, optimized with respect to certain criteria. Geometric paths require to be shortened according to a given metric. The simplest technique consists in picking pairs of points on the path and linking them by a shortest path: if the shortest path is collision-free, it replaces the corresponding portion of the initial path. Doing so iteratively, the path becomes shorter and shorter. The iterative process stops as soon as it does not significantly improve the quality of the path. The technique gives good results in practice.

Beside this simple technique, several variational methods operating in the trajectory space have been introduced. Among the very first ones, Barraquand and Ferbach[2] propose to replace a constrained problem by a convergent series of less constrained subproblems increasingly penalizing motions that do not satisfy the constraints. Each sub-problem is then solved using a standard motion planner. This principle has been successfully extended recently to humanoid robot motion planning.[9]

Another method introduced by Quinlan and Khatib consists in modeling the motion as a mass-spring system.[32] The motion then appears as an elastic band that is reshaped according to the application of an energy function optimizer. The method applies for nonholonomic systems as soon as the nonholonomic metric is known[16] as well as for real-time obstacle avoidance in dynamic environments.[6]

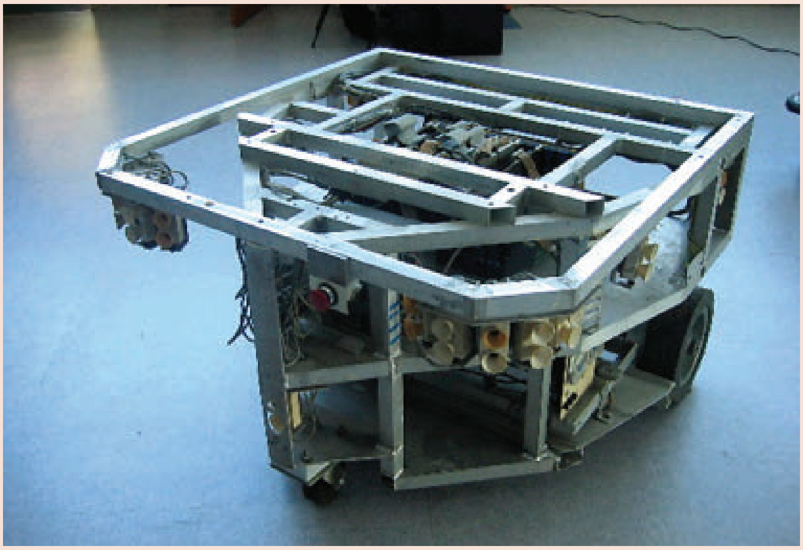
Recently, successful improvements have been introduced by following the same basic principle of optimizing an initial guess in motion space. Zucker et al. take advantage of a simple functional expressing a combination of smoothness and clearance to obstacles to apply gradient descent in the trajectory space.[40] A key point of the method is to model a trajectory as a geometric object, invariant to parametrization. In the same framework, Kalakrishman et al. propose to replace the gradient descent with a derivative-free stochastic optimization technique allowing us to consider non-smooth costs.[19]

What We Know and What We Do Not Know About Optimal Motion for Wheeled Mobile Robots

Mobile robots constitute a unique class of systems for which the question of optimal motion is best understood. Since the seminal work by Dubins in the 1950s,[13] optimal motion planning and control for mobile robots has attracted a lot of interest. We briefly review how some challenging optimal control problems have been solved and which problems still remain open.

Let us consider four control models of mobile robots based on the model of a car (Figure 2). Two of them are simplified models of a car: the so-called Dubins (Figure 3) and Reeds-Shepp (Figure 4) cars respectively. The

Figure 5. The Hilare robot at LAAS-CNRS in the 1990s.



Dubins car moves only forward. The Reeds-Shepp car can move forward and backward. Both of them have a constant velocity of unitary absolute value. Such models account for a lower bound on the turning radius, that is, the typical constraint of a car. Such a constraint does not exist for a two-wheel differentially driven mobile robot. This robot may turn on the spot while a car cannot. Let us consider two simple control schemes of a two-driving wheel mobile robot:[b] in the first one (Hilare-1), the controls are the linear velocities of the wheels; in the second one (Hilare-2), the controls are the accelerations (that is, the second system is a dynamic extension of the first).

Figure 6. Hilare-1: A different drive mobile robot. First model: The controls are the velocities of the wheels. The optimal controls are bang-bang. Optimal trajectories are made of pure rotations and of straight-line segments.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}1/2cos\theta\\ 1/2sin\theta\\ 1\end{bmatrix} v_1 + \begin{bmatrix}1/2cos\theta\\ 1/2sin\theta\\ -1\end{bmatrix} v_2$$

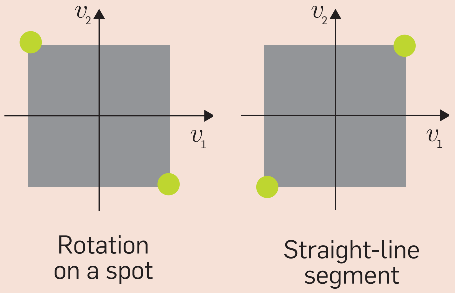
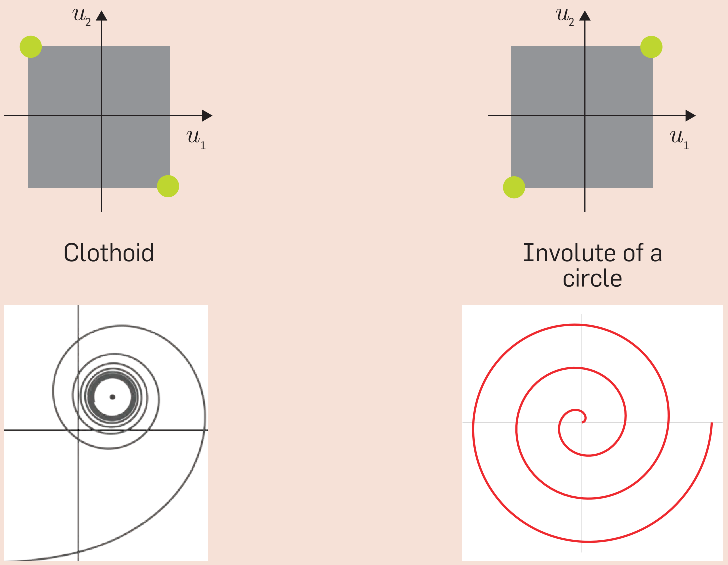


Figure 7. Hilare-2: A different drive mobile robot. Second model: The controls are the acceleration of the wheels. The optimal controls are bang-bang. Optimal trajectories are made of arcs of clothoids and of arcs of involute of a circle.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{v}_1\\ \dot{v}_2\end{bmatrix} = \begin{bmatrix}1/2(v_1+v_2)cos\theta\\ 1/2(v_1+v_2)sin\theta\\ v_1-v_2\\ 0\\ 0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} u_1 + \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix} u_2$$



Time-optimal trajectories. The car-like robots of figures 3 and 4 represent two examples of non-linear systems for which we know exactly the structure of the optimal trajectories. Note that in both examples the norm of the linear velocity is assumed to be constant. In those cases, time-optimal trajectories are supported by the corresponding shortest paths. Dubins solved the problem for the car moving only forward.[13] More than 30 years later, Reeds and Shepp[33] solved the problem for the car moving both forward and backward. The problem has been completely revisited with the modern perspective of geometric techniques in optimal control theory:[37,38] the application of PMP shows that optimal trajectories are made of arcs of a circle of minimum turning radius (bang-bang controls) and of straight-line segments (singular trajectories). The complete structure is then derived from geometric arguments that characterize the switching functions. The Dubins and Reeds-Shepp cars are among the few examples of nonlinear systems for which optimal control is fully understood. The same methodology applies for velocity-based controlled differential drive vehicles (Hilare-1 in Figure 6). In that case, optimal trajectories are bang-bang, that is, made of pure rotations and straight-line segments. The switching functions are also fully characterized.[1] This is not the case for the dynamic extension of the system, that is, for acceleration-based controlled differential drive vehicles (Hilare-2 as shown in Figure 7). Only partial results are known: optimal trajectories are bang-bang (that is, no singular trajectory appears) and are made of arcs of clothoid and involutes of a circle.[15] However, the switching functions are unknown. The synthesis of optimal control for the Hilare-2 system the remain an open problem.

While the existence of optimal trajectories is proven for the four systems shown here, a last result is worth mentioning. If we consider the Reeds-Shepp car optimal control problem in presence of an obstacle, even if we allow the car touching the obstacles, it has been proven that a shortest path may not exist.[10]

Motion planning. These results are very useful for motion planning in the presence of obstacles. In figures 3 and 4 we display the reachable domain

[b] The distance between the wheels is supposed to be 2.

for both the Dubins and Reeds-Shepp cars. While the reachable set of the Reeds-Shepp car is a neighborhood of the origin, it is not the case for the Dubins car. Stated differently, the Reeds-Shepp car is small-time controllable, while the Dubins car is only controllable. The consequence in terms of motion planning is important. In the case of the Reeds-Shepp car, any collision-free—not necessarily feasible—path can be approximated by a sequence of collision-free feasible paths. Optimal paths allow building the approximation, giving rise to an efficient motion-planning algorithm.[24] Not only does such an algorithm not apply for the Dubins car, we still do not know whether the motion-planning problem for Dubins car is decidable or not.

In Laumond et al.,[24] we prove the number of maneuvers to park a car varies as the inverse of the square of the clearance. This result is a direct consequence of the shape of the reachable sets. So, the combinatorial complexity of (nonholonomic) motion planning problems is strongly related to optimal control and the shape of the reachable sets in the underlying (sub-Riemannian) geometry.[17]

Conclusion

When optimal solutions cannot be obtained for theoretical reasons (for example, nonexistence) or for practical ones (for example, untractability), we have seen how the problem can be reformulated either by considering a discrete representation of space and/or time, or by slightly changing the optimization criterion, or by resorting to numerical optimization algorithms. In all these cases, the resulting solutions are only approximated solutions of the original problem.

In conclusion, it appears the existence of optimal robot motions is rarely guaranteed. When it is, finding a solution has never been proven to be a decidable problem as is the motion-planning problem. So, "optimal motion" is most often an expression that should be understood as "optimized motion," that is, the output of an optimization numerical algorithm. However, motion optimization techniques follow progress in numerical optimization with effective practical results on real robotic platforms, if not with new theoretical results.

The distinction between optimal and optimized motions as it is addressed in this article does not cover all facets of optimality in robot motion. In a companion article,[23] we consider the issue of motion optimal as an action selection principle and we discuss its links with machine learning and recent approaches to inverse optimal control.

Acknowledgments

The article benefits from comments by Quang Cuong Pham, from a careful reading by Joel Chavas, and above all, from the quality of the reviews. The work has been partly supported by ERC Grant 340050 Actanthrope, by a grant of the Gaspar Monge Program for Optimization and Operations Research of the Fédération Mathématique Jacques Hadamard (FMJH) and by the grant ANR 13-CORD-002-01 Entracte.

References

1. Balkcom, D. and Mason, M. Time optimal trajectories for differential drive vehicles. *Int. J. Robotics Research 21*, 3 (2002), 199–217.
2. Barraquand, J. and Ferbach, P. A method of progressive constraints for manipulation planning. *IEEE Trans. on Robotics and Automation 13*, 4 (1997) 473–485.
3. Barraquand, J. and Lacombe, J.-C. Robot motion planning: A distributed representation approach. *Intern. J. of Robotics Research 10*, 6 (1991), 628–649.
4. Bobrow, J., Dubowsky, S. and Gibson, J. Time-optimal control of robotic manipulators along specified paths. *Int. J. of Robotics Research 4*, 3 (1985), 3–17.
5. Brady, M., Hollerbach, J., Johnson, T., Lozano-Pérez, T. and Masson, M.T. *Robot Motion: Planning and Control.* MIT Press, 1983.
6. Brock, O. and Khatib, O. Elastic strips: A framework for motion generation in human environments. *Int. J. of Robotics Research 21*, 12 (2002), 1031–1052.
7. Choset, H., Lynch, K.M., Hutchinson, S., Kantor, A., Burgard, W., Kavraki, L.E. and Thrun, S. *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, Cambridge, MA, June 2005.
8. Connolly, C. and Grupen, R. Applications of harmonic functions to robotics. *J. of Robotic Systems 10*, 7 (1992), 931–946.
9. Dalibard, S., Nakhaei, A., Lamiraux, F. and Laumond, J.-P. Whole-body task planning for a humanoid robot: A way to integrate collision avoidance. In *Proceedings of IEEE-RAS Int. Conference on Humanoid Robots*, 2009.
10. Desaulniers, G. On shortest paths for a car-like robot maneuvering around obstacles. *Robotics and Autonomous Systems 17* (1996), 139–148.
11. Diehl, M. and Mombaur, K., eds. Fast Motions in Biomechanics and Robotics. *Lecture Notes in Control and Information Sciences*, vol 340. Springer, 2006.
12. Donald, B., Xavier, P., Canny, J. and Reif, J. Kinodynamic motion planning. *JACM 40*, 5 (1993), 1048–1066.
13. Dubins, L. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Amer. J. of Mathematics 79* (1957), 497–516.
14. Hopcroft, J. Schwartz, J. and Sharir, M. *Planning, Geometry, and Complexity of Robot Motion.* Ablex, 1987.
15. Jacobs, P., Laumond, J.-P. and Rege, A. Nonholonomic motion planning for HILARE-like mobile robots. *Intelligent Robotics.* M. Vidyasagar and M.Trivedi, eds. McGraw Hill, 1991.
16. Jaouni, H., Khatib, M. and Laumond, J.-P. Elastic bands for nonholonomic car-like robots: algorithms and combinatorial issues. In *Robotics: The Algorithmic Perspective.* P. Agarwal, L. Kavraki, and M. Mason, eds. A.K. Peters, 1998.
17. Jean, F. Complexity of nonholonomic motion planning. *Int. J. of Control 74*, 8 (2001), 776–782.
18. Jurjdevic, V. *Geometric Control Theory.* Cambridge University Press, 1996.
19. Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P. and Schaal, S. Stomp: Stochastic trajectory optimization for motion planning. In *Proceedings of the IEEE Int. Conf. on Robotics and Automation* (2011).
20. Kavraki, L., Svestka, P., Latombe, J.-C. and Overmars, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics and Automation 12*, 4 (1996), 566–580.
21. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *The Int. J. of Robotics Research 5*, 1 (1986), 90–98, 1986.
22. Latombe, J.-C. *Robot Motion Planning.* Kluwer Academic Press, 1991.
23. Laumond, J., Mansard, N. and Lasserre, J. Robot motion optimization as action selection principle. *Commun. ACM*, (to appear).
24. Laumond, J.-P., Jacobs, P., Taix, M. and Murray, R. A motion planner for nonholonomic mobile robots. *IEEE Trans. on Robotics and Automation, 10*, 5 (1994), 577–593.
25. Laumond, J.-P. and Siméon, T. Notes on visibility roadmaps and path planning. *New Directions in Algorithmic and Computational Robotics.* B. Donald, K. Lynch, and D. Rus, eds. A.K. Peters, 2001.
26. LaValle, S. *Planning Algorithms.* Cambridge University Press, 2006.
27. LaValle, S. and Kuffner, J. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and Computational Robotics: New Directions.* B. Donald, K. Lynch and D. Rus, eds. A.K. Peters, 2001, 293–308.
28. Lozano-Pérez, T. Spatial planning: A configuration space approach. *IEEE Trans. on Computer 32*, 2 (1983), 108–120.
29. Neustadt, L. Optimization, a moment problem and non linear programming. *SIAM J. Control* (1964), 33–53.
30. Paul, R. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Cambridge, MA, 1st edition, 1982.
31. Pfeiffer, F. and Johanni, R. A concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation 3*, 2 (1987).
32. Quinlan, S. and Khatib, O. Elastic bands: Connecting path planning and control. In *Proceedings of the IEEE Int. Conf. on Robotics and Automation* (1993).
33. Reeds, J. and Shepp, L. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics 145*, 2 (1990), 367–393.
34. Rimon, E. and Koditschek. Exact robot navigation using artificial potential fields. *IEEE Trans. on Robotics and Automation 8*, 5 (1992), 501–518.
35. Schwartz, J. and Sharir, M. On the piano movers problem II: General techniques for computing topological properties of real algebraic manifolds. *Advances of Applied Mathematics 4* (1983), 298–351.
36. Shin, K.G. and McKay, N.D. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Trans. on Automatic Control 30*, 6 (1985), 531–541.
37. Souères, P. and Laumond, J.-P. Shortest paths synthesis for a car-like robot. *IEEE Trans. on Automatic Control 41*, 5 (1996), 672–688.
38. Sussmann, H. and Tang, G. Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control. Rutgers Center for Systems and Control Technical Report 91-10, 1991.
39. Verscheure, D., Demeulenaere, B., Swevers, J., Schutter, J.D. and Diehlm M. Time-optimal path tracking for robots: A convex optimization approach. *IEEE Trans. on Automatic Control 54*, 10 (2009), 2318–2327.
40. Zucker, M., Ratliff, N., Dragan, A., Pivtoraiko, M., Klingensmith, M., Dellin, C., Bagnell, J. and Srinivasa, S. Chomp: Covariant Hamiltonian optimization for motion planning. *Int. J. of Robotics Research 32*, 9-10 (2013), 1164–1193.

Jean-Paul Laumond (jpl@laas.fr) is a CNRS director of research at LAAS, Toulouse, France.

Nicolas Mansard (nmansard@laas.fr) is a CNRS researcher at LAAS, Toulouse, France.

Jean-Bernard Lasserre (lasserre@laas.fr) is CNRS director of research at LAAS, Toulouse, France.

Call for Nominations

The ACM Doctoral Dissertation Competition

Rules of the Competition

ACM established the Doctoral Dissertation Award program to recognize and encourage superior research and writing by doctoral candidates in computer science and engineering. These awards are presented annually at the ACM Awards Banquet.

Submissions

Nominations are limited to one per university or college, from any country, unless more than 10 Ph.Ds are granted in one year, in which case two may be nominated.

Eligibility

Please see our website for exact eligibility rules. Only English language versions will be accepted. Please send a copy of the thesis in PDF format to emily.eng@acm.org.

Sponsorship

Each nomination shall be forwarded by the thesis advisor and must include the endorsement of the department head. A one-page summary of the significance of the dissertation written by the advisor must accompany the transmittal.

Deadline

Submissions must be received by **October 31, 2014** to qualify for consideration.

Publication Rights

Each nomination must be accompanied by an assignment to ACM by the author of exclusive publication rights. (Copyright reverts to author if not selected for publication.)

Publication

Winning dissertations will be published by ACM in the ACM Books Program and appear in the ACM Digital Library. Honorable mention dissertations will appear in the ACM Digital Library.

Selection Procedure

Dissertations will be reviewed for technical depth and significance of the research contribution, potential impact on theory and practice, and quality of presentation. A committee of individuals serving staggered five-year terms performs an initial screening to generate a short list, followed by an in-depth evaluation to determine the winning dissertation.

The selection committee will select the winning dissertation in early 2015.

Award

The Doctoral Dissertation Award is accompanied by a prize of $20,000 and the Honorable Mention Award is accompanied by a prize of $10,000. Financial sponsorship of the award is provided by Google.

For Submission Procedure

See http://awards.acm.org/doctoral_dissertation/

acm

Association for
Computing Machinery

research highlights

DOI:10.1145/2647748

To view the accompanying paper, visit doi.acm.org/10.1145/2647750

Technical Perspective
Portraiture in the Age of Big Data

By Alexei A. Efros

"I have never *been aware before how many faces there are.*

There are quantities of human beings, but there are many more faces, for each person has several."

—*Rainer Maria Rilke*

How many faces does a person possess? That is, how much does a face vary in its appearance over the lifetime of a given individual? Aging, of course, produces the greatest changes in facial structure, as anyone who has ever tried to pick out an adult friend from his first-grade class photo can attest. This is why many official ID documents require their holders to update their photograph every 5–10 years. But even at shorter temporal scales (days or weeks) there could be significant variations due, for instance, to the changes in hairstyle, eyewear, facial hair, or makeup. Add to that the changing pose (full face, profile, 3/4 view) and the constant parade of momentary changes in facial expression: happy, amused, content, angry, pensive...there are literally hundreds of words for describing the nuances of the human face.

This, of course, poses a great problem for portraitists, for how can a single portrait, even the most masterful one, ever come close to capturing the full *gestalt* of a living face? Indeed, many great artists have been obsessively preoccupied with this very question. Rembrandt painted over 60 self-portraits over his lifetime, a monumental study of his own face. Da Vinci, a master of visual illusion, purposefully blurred the corners of Mona Lisa's mouth and eyes, perhaps in an effort to transcend the immediacy of the moment and let the observer mentally "paint in" the missing details. The cubists argued that to truly seize the essence of a person requires forgoing the traditional single-perspective 2D pictorial space and instead capture the subject from several viewpoints simultaneously, fusing them into a single image. Cinema, of course, has helped redefine portraiture as something beyond a single still image—the wonderful "film portraits" of the likes of Charlie Chaplin or Julia Andrews capture so much more than the still-life versions. Yet, even the cinema places strict limits on the temporal dimension since filming a movie rarely takes longer than a few months, which is only a small fraction of a person's life.

The following paper is, in some sense, part of this grand tradition—the perpetual quest to capture the perfect portrait. Its principal contribution is in adapting this age-old problem to our post-modern, big data world. The central argument is that there already exist thousands of photographs of any given individual, so there is no need to capture more. Rather, the challenge is in *organizing* and *presenting* the vast amount of visual data that is already there. But how does one display thousands of disparate portraits in a human-interpretable form? Show them all on a large grid, *à la* Warhol? Each will be too small to see. Play them one after another in a giant slideshow? The visual discontinuities will soon make the viewer nauseated.

The solution presented by these authors is wonderfully simple: first, they represent all photos as nodes in a vast graph with edges connecting portraits that have a high degree of visual similarity (in pose, facial expression, among others); then, they compute a smooth path through the graph and make it into a slideshow. The surprising outcome is that, typically, there are enough photographs available for a given individual that the resulting slideshow appears remarkably smooth, with each photo becoming but a frame in a continuous movie, making these "moving portraits" beautifully mesmerizing.

The following paper could be thought of as a type of Visual Memex specialized for faces.

This type of graph representation betrays another intellectual lineage that goes back to Vannevar Bush and his article "As We May Think" (*The Atlantic*, 1945). Bush proposed the concept of the Memex (Memory Extender), a device that would organize information not by categories, but via direct associations between instances, using a vast graph. This idea has been influential in the development of hypertext, but the original Memex concept is actually much broader, encompassing data types beyond text (for example, photographs, sketches, video, audio), and describing paths through the instance graph (Bush called them "associative trails"). So, the following paper could be thought of as a type of Visual Memex specialized for faces.

In many ways this work signifies the coming of age of computer vision as a practical discipline. The work is one of the first instances when a fairly complex computer vision system (itself utilizing several nontrivial components such as face detection and face alignment) has become a "button" in a mainstream software product (Google Picasa) used by millions of people. So, read the paper, try the software—I do not think you will be disappointed.

Alexei A. Efros (efros@eecs.berkeley.edu) is an associate professor of electrical engineering and computer science at the University of California, Berkeley.

DOI:10.1145/2647750

Moving Portraits

By Ira Kemelmacher-Shlizerman, Eli Shechtman, Rahul Garg, and Steven M. Seitz

Abstract

We present an approach for generating face animations from large image collections of the same person. Such collections, which we call *photobios*, are remarkable in that they summarize a person's life in photos; the photos sample the appearance of a person over changes in age, pose, facial expression, hairstyle, and other variations. Yet, browsing and exploring photobios is infeasible due to their large volume. By optimizing the quantity and order in which photos are displayed and cross dissolving between them, we can render smooth transitions between face pose (e.g., from frowning to smiling), and create *moving portraits* from collections of still photos. Used in this context, the *cross dissolve* produces a very strong motion effect; a key contribution of the paper is to explain this effect and analyze its operating range. We demonstrate results on a variety of datasets including time-lapse photography, personal photo collections, and images of celebrities downloaded from the Internet. Our approach is completely automatic and has been widely deployed as the "Face Movies" feature in Google's Picasa.

1. INTRODUCTION

People are photographed thousands of times over their lifetimes. Taken together, the photos of each person form his or her visual record. Such a visual record, which we call a *photobio*, samples the appearance space of that individual over time, capturing variations in facial expression, pose, hairstyle, and so forth. While acquiring photobios used to be a tedious process, the advent of photo sharing tools like Facebook coupled with face recognition technology and image search is making it easier to amass huge numbers of photos of friends, family, and celebrities. As this trend increases, we will have access to increasingly complete photobios. The large volume of such collections, however, makes them very difficult to manage, and better tools are needed for browsing, exploring, and rendering them.

If we could capture *every* expression that a person makes, from every pose and viewing/lighting condition, and at every point in their life, we could describe the *complete* appearance space of that individual. Given such a representation, we could render any view of that person on demand, in a similar manner to how a lightfield[18] enables visualizing a static scene. However, key challenges are (1) the face appearance space is extremely high dimensional, (2) we generally have access to only a sparse sampling of this space, and (3) the mapping of each image to pose, expression, and other parameters is not generally known a priori. In this paper, we take a step toward addressing these problems to create animated viewing experiences from a person's photobio.

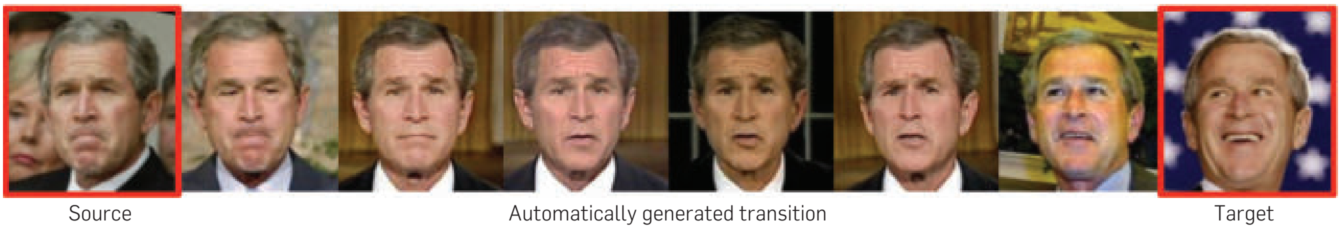
A key insight in our work is that *cross dissolving* well-aligned images produces a very strong motion sensation. While the cross dissolve (also known as cross fade, or linear intensity blend) is prevalent in morphing and image-based-rendering techniques (e.g., Chen and Williams,[7] Levoy and Hanrahan,[18] Seitz and Dyer[22]), it is usually used in tandem with a geometric warp, the latter requiring accurate pixel correspondence (i.e., optical flow) between the source images. Surprisingly, the cross dissolve *by itself* (without correspondence/flow estimation) can produce a very strong sensation of movement, particularly when the input images are well aligned. We explain this effect and prove some remarkable properties of the cross dissolve. In particular, given two images of a scene with small motion between them, a cross dissolve produces a sequence in which the edges move smoothly, with *nonlinear ease-in, ease-out dynamics*. Furthermore, the cross dissolve can also synthesize physical illumination changes, in which the light source direction moves during the transition.

Our photobios approach takes as input an *unorganized* collection of photos of a person, and produces animations of the person moving continuously (Figure 1). The method operates best when the photo collection is very large (several hundred or thousand photos), but produces reasonable results for smaller collections. As a special case of interest, we first show results on time-lapse sequences, where the same person is photographed every day or week over a period of years. We then apply the technique to more standard image collections of the same person taken over many years, and also to images of celebrities downloaded from the Internet.

Our approach is based on representing the set of images in a photobio as nodes in a graph, solving for optimal paths, and rendering a stabilized transition from the resulting image sequence. The key issues therefore are (1) defining the edge weights in the graph, and (2) creating a compelling, stabilized output sequence. Our approach leverages automatic face detection, pose estimation, and robust image comparison techniques to both define the graph and create the final transitions. The pipeline is almost entirely automatic—the only step that requires manual assistance is to identify the subject of interest when an image contains multiple people. Automating this step is also likely possible using face recognition techniques.[4]

The original version of this paper is entitled "Exploring Photobios" and was published in *ACM Transactions on Graphics (SIGGRAPH)*, 2011.

Figure 1. Automatically generated transition between George W. Bush frowning (left) and smiling (right).



We note that graph-based techniques have been used for other animation tasks,[2,6,10,16,25] but ours is the first to operate on unstructured photos collections and propose the moving portraits idea. Our ability to operate on unstructured photo collections is a direct result of the maturity of computer-vision-based face analysis techniques that are now in widespread use in the research community and in digital cameras, Google Streetview, Apple's iPhoto, etc. These techniques include face detection, face tagging, and recognition.

The *Face Movies* feature of Picasa's 3.8 release[21] provides an implementation of moving portraits, targeted to personal photos. This feature leverages Picasa's built-in face recognition capabilities, and enables creating a face movie of a person with minimal effort (a single click). Deploying the approach at scale (with photo collections numbering in the tens of thousands) required a number of modifications to our basic algorithm, which we describe.

2. THE FACE GRAPH

We represent the photobio as a graph (Figure 2), where each photo is a node, and edge weights encode distance (inverse similarity) between photos. Photo similarity is measured as a function of similarity in facial appearance (expression, glasses, hair, etc.), 3D pose (yaw, pitch), and time (how many hours/days apart). That is, a distance between faces *i* and *j* is defined as $D(i, j)$:

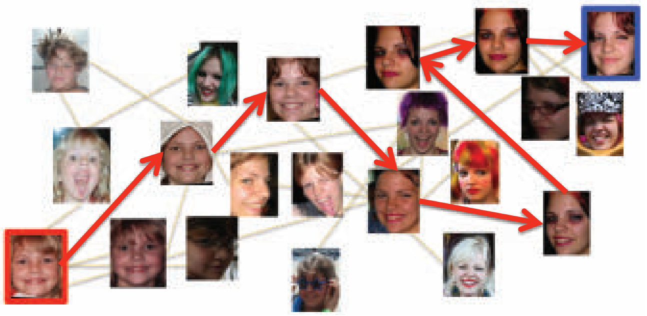
$$D(i, j) = 1 - \prod_{s \in \{\text{app, yaw, pitch, time}\}} (1 - D_s(i, j)), \qquad (1)$$

where "app" stands for facial appearance distance, and 3D head pose distance is represented by yaw and pitch angles. Next, we define each of these similarity terms and how they are computed.

2.1. Head pose estimation and alignment

To compute face pose (yaw, pitch) distances between photos, we estimate 3D head pose as follows (illustrated in Figure 3). Each photo is preprocessed automatically by first running a face detector[5] followed by a fiducial points detector[9] that finds the left and right corners of each eye, the two nostrils, the tip of the nose, and the left and right corners of the mouth. We ignore photos with low detection confidence (less than 0.5 in face detection and less than −3 in detection of the fiducial points). Throughout our experiments, despite variations in lighting, pose, scale, facial expression, and identity, this combination of methods was extremely robust for near-frontal faces with displacement errors gradually increasing as the face turns to profile.

Figure 2. The face graph. Face photos are represented by nodes, and edges encode distance (inverse similarity) between faces. Distance is a function of facial appearance (expression, hairstyle, etc.), 3D head pose, and difference in the date/time of capture. Once constructed, we can compute smooth transitions (red arrows) between any pair of photos (e.g., red and blue bordered photos in this example) via shortest path computations.



The next step is to detect pose, which is achieved by geometrically aligning each detected face region to a 3D template model of a face. We use a neutral face model from the publicly available space-time faces[25] dataset for the template. We estimate a linear transformation that transforms the located fiducial points to prelabeled fiducials on the template model, and use RQ decomposition to find rotation and scale. We then estimate the yaw, pitch, and roll angles from the rotation matrix. Given the estimated pose, we transform the template shape to the orientation of the face in the image and warp the image to a frontal pose using point-set z-buffering[13] to account for occlusions. This results in a roughly frontal version of the given photo.

The head pose distance between two faces is measured separately for yaw and pitch angles and each of these is normalized using a robust logistic function:

$$D_{\text{yaw}}(i, j) = L(|Y_i - Y_j|); \quad D_{\text{pitch}}(i, j) = L(|P_i - P_j|), \qquad (2)$$

where the logistic function $L(d)$ is defined as $L(d) = 1/[1 + \exp(-\gamma(d - T)/\lambda)]$ with $\gamma = \ln(99)$. It normalizes the distances d to the range [0, 1] such that the value $d = T$ is mapped to 0.5 and the values $d = T \pm \lambda$ map to 0.99 and 0.01, respectively.

2.2. Facial appearance similarity

Face appearance, measured in photo pixel values, can vary dramatically between photos of the same person, due to changes in lighting, pose, color balance, expression, hair, etc. Better invariance can be achieved by use of various descriptors instead of raw pixel colors. In particular, Local Binary Pattern (LBP) histograms have proven effective for face recognition and retrieval tasks.[1, 14] LBP operates by replacing each pixel with a binary code that represents the relative brightness of each of that pixel's immediate neighbors. Each neighbor is assigned a 1 if it is brighter, or 0 if it is darker than the center pixel. This pattern of 1's and 0's for the neighborhood defines a per pixel binary code. Unlike raw RGB values, these codes are invariant to camera bias, gain, and other common intensity transformations. Additionally, some invariance to small spatial shifts is obtained by dividing the image into a grid of super-pixel cells, and assigning each cell a *descriptor* corresponding to the histogram of binary patterns for pixels within that cell.[1]

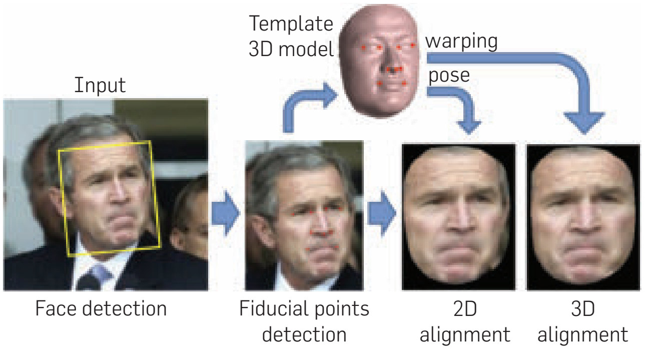
We calculate LBP descriptors on aligned faces and estimate a separate set of descriptors for the eyes, mouth, and hair regions, where a descriptor for a region is a concatenation of participating cells' descriptors. The regions and example matching results are shown in Figure 4. The distance between two face images i and j, denoted d_{ij}, is then defined by χ^2-distance between the corresponding descriptors. The combined appearance distance function is defined as

$$D_{\text{app}}(i, j) = 1 - \left(1 - \lambda^m d^m_{ij}\right)\left(1 - \lambda^e d^e_{ij}\right)\left(1 - \lambda^h d^h_{ij}\right), \qquad (3)$$

where $d^{m,e,h}$ are the LBP histogram distances restricted to the mouth, eyes, and hair regions, respectively, and $\lambda^{m,e,h}$ are the corresponding weights for these regions. For example, assigning $\lambda^m = 1$ and $\lambda^e = \lambda^h = 0$ will result in only the mouth region being considered in the comparison. In our experiments, we used $\lambda^m = 0.8$ and $\lambda^e = \lambda^h = 0.1$.

When time or date stamps are available, we augment Equation (3) with an additional term measuring L_2 difference in time.

Figure 3. Automatic alignment and pose estimation. We first localize the face and estimate fiducial points (e.g., eyes, nose, mouth). Then a 3D template model is used to estimate pose and to warp the image to a frontal view.



Each distance is normalized using a robust logistic function $L(d)$.

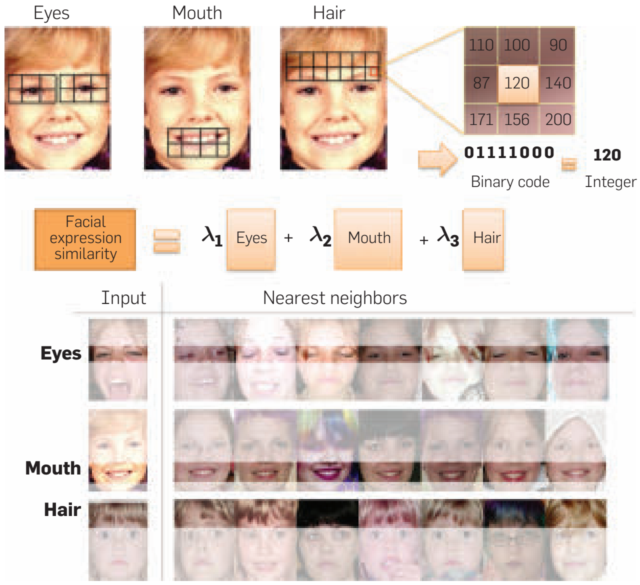
3. PHOTO PATHS

By constructing a face graph we can now traverse paths on the graph and find smooth, continuous transitions from the still photos contained in a photobio. We find such paths either via traditional shortest path or by greedy walk algorithms on the face graph.

Given any two photos, we can find the smoothest path between them by solving for the shortest path in the face graph. We are interested in finding a path with the minimal cost (sum of distances), which is readily solved using Dijkstra's algorithm. The number of in-between images is controlled by raising the distance to some power: $D(i, j)^\alpha$. The exponent α is used to nonlinearly scale the distances, and provides additional control of step size in the path planning process.

Given any starting point, we can also produce a smooth path of arbitrary length by taking walks on the graph. Stepping to an adjacent node with minimal edge distance generally results in continuous transitions. There are a number of possible ways to avoid repetitions, for example, by injecting randomness. We obtained good results simply by deleting previously visited nodes from the graph (and all of their incident edges). For collections with time/date information, we encourage chronological transitions by preferentially choosing steps that go forward in time.

Figure 4. Appearance similarity is calculated separately for eyes, mouth and hair. For each region we show the query image and its nearest neighbors from a dataset of photos of the same person across many ages and expressions. Note how a closed eye input retrieves other faces with mostly closed eyes, and similarly for open mouth and hairstyle.



4. REAL-TIME PERFORMANCE

An important requirement for Picasa's Face Movies implementation is real-time performance, that is, the face movie should start playing almost immediately when the feature is selected. We achieved this goal through a number of optimizations.

First, we simplified the similarity descriptors. Specifically, we observed that the mouth is highly correlated with the eyes and other regions of the face in forming expressions. Hence, we found it sufficient to match only the mouth region. More surprisingly, we found that head *orientation* is also well correlated with the appearance of the mouth region (producing mouth foreshortening and rotation), eliminating the need to compute pose explicitly. Similarly, we found that using 2D affine image alignment rather than 3D warping produces satisfactory results at lower cost. These optimizations, as well as the use of HOG (Histogram of Oriented Gradients) features[8] in place of LBP, significantly reduced matching costs.

By default, Face Movies creates and renders a greedy walk rather than an optimized path, as the former is faster to compute. We accomplish this using a multithreaded approach where one thread computes the image graph on the fly and selects the next image to step to, while the other thread renders the transition between the previous two images. For computing the greedy sequence, we first sort all the images by time, start at the oldest image, and then consider the next 100 images in the sequence, chronologically. We choose the one with the closest similarity as the next image to step to. This procedure repeats until the time window overlaps the most recent photo, at which point we reverse direction, that is, select photos going monotonically back in time. We then continue to oscillate forward and back until all images have been shown. We give preference to *starred* photos by reducing their edge weights so that they are more likely to be shown toward the beginning of the face movie.

The user can control the length of the movie by specifying the number of photos in the face movie and we find the optimal sequence of desired length via dynamic programming. To achieve reasonable performance (a delay of a few seconds, even for collections of tens of thousands of images), we employed additional optimizations, such as breaking the sequence into separately optimized chunks of 1000 images and sparsifying the graph by considering only 100 neighbors for each image.

5. ANIMATING PHOTO PATHS

Now that we have computed an optimal sequence of still photos, how can we render them as a continuous animation? Although pose-, expression-, and time-aligned, it is still a sequence of independently taken photos; creating a smooth animation requires rendering compelling transitions from one photo to the next. Morphing techniques can produce excellent transitions, but require accurate correspondence between pixels in the images, which is difficult to obtain. A simpler alternative is to use a *cross dissolve*. The *cross dissolve* or *cross fade* transitions between two images (or image sequences) by simultaneously fading one out while fading the other in over a short time interval. Mathematically, the cross dissolve is defined as

$$I_{\text{out}}(t) = (1 - t)I_{\text{in}_1} + tI_{\text{in}_2}, \tag{4}$$

where I_{in_1} and I_{in_2} are the input images and $I_{\text{out}}(t)$ is the output sequence.

This effect is often combined with geometric warps in morphing,[3, 22] and image-based rendering methods,[18] to synthesize motion between photos. More surprisingly, the cross dissolve *by itself* (without correspondence/flow estimation) can produce a very strong sensation of movement, particularly when the input images are well aligned. For example, Figure 5 shows a cross dissolve between two photos of a person's face, in which both the lighting and features appear to move realistically. While it makes sense that warping an image produces a motion sensation, why would motion arise from a simple intensity blend? We explain this effect and prove some remarkable properties of the cross dissolve.

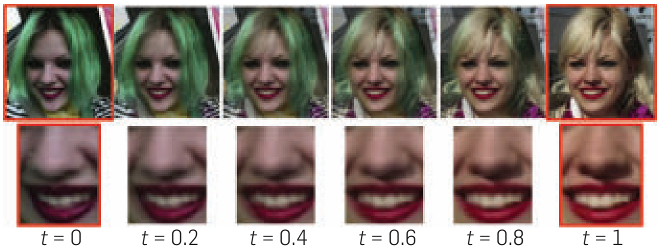
We show that the cross dissolve produces not just the illusion of motion, but true motion; given two images of a scene with small motion between them, a cross dissolve produces a sequence in which image edges *move* smoothly, with *nonlinear ease-in, ease-out dynamics*. Furthermore, the cross dissolve can synthesize physical illumination changes, in which the light source direction moves during the transition. We briefly describe these effects here, for further analysis see our SIGGRAPH paper.[15]

5.1. Image edge motion

Images are composed of edges of different locations, orientations, and frequencies. By modeling the effects of the cross dissolve on edges, we can thereby analyze image motion in general. Image edges arise from rapid spatial changes in intensity in a still photo. Real image edge profiles tend to be smooth rather than discontinuous, due to the optical blurring effects of the imaging process.[19, 20] Indeed, the convolution of a step-edge with a Gaussian[a] blurring kernel is the *erf* function: $erf(x) = \int_0^x e^{-t^2} dt$. This function is very closely approximated as a segment of a

[a] We note that the Gaussian is an imperfect PSF model,[12] but still useful as a first-order approximation.

Figure 5. Cross dissolve synthesizes motion. Notice how the edges of the nose and mouth move realistically, as does the lighting (more clearly seen in video: http://grail.cs.washington.edu/photobios/).



sine curve. We use this sine edge model to prove properties of the cross dissolve and then show the correlation with real image edges.

Consider two sine waves (each represents a different image) where one is a translated (and optionally amplitude-scaled) version of the other. Specifically, we consider $\alpha\sin(mx)$ and $\sin(mx + d)$ so that d is the phase shift (spatial translation) and α is the amplitude scale. Cross dissolving these two sine waves produces a sequence of sine waves given as follows:

$$(1-t)\alpha\sin(mx) + t\sin(mx+d) = c\sin(mx+k), \tag{5}$$

where $t \in [0, 1]$ and

$$k = \arctan\frac{t\sin d}{(1-t)\alpha + t\cos d}, \tag{6}$$

$$c^2 = \alpha^2(1-t)^2 + t^2 + 2(1-t)\alpha t\cos d. \tag{7}$$

Therefore, cross dissolving two sines (image edges) with different phases produces a *motion*, where the phase k is smoothly interpolated. This simple analysis gives rise to a number of remarkable observations (Figure 6):

- The speed of the motion is determined by the phase k. Note that k is not linear, but resembles the *ease-in, ease-out* curves.[17] This type of curve is known to have a major role in producing more believable animations; it is remarkable that it arises naturally in the cross dissolve. Furthermore, different edges move at different rates, and with different ease-in/ease-out parameters, depending on their phase offsets. In particular, large displacements give rise to more exaggerated ease-in/ease-outs.
- The perceived motion is strictly less than a half-period. Hence, low-frequency edges (lower m) can move relatively large distances, whereas high-frequency edges can move only slightly. When the phase offset reaches π (a half-period), the edge disappears entirely at the center frame and becomes a constant function. This phenomenon, in which image content fades away during a transition, is known as *ghosting*.[24]
- There is a gradual decrease in image contrast toward the midpoint of the transition, due to the drop in amplitude of the sine, according to c in Equation (7). For example, the highlights get darker, and the shadows get lighter. This reduction in dynamic range is subtle (except in the most extreme cases), yet serves to hide visual artifacts like ghosting[24] in the frames in which they are most likely to appear.
- This motion effect only works for edges with (approximately) the same frequency. Interpolating sines with different frequencies produces multi-model curves that do not resemble edges (another form of ghosting).

Note that our analysis so far is based on a periodic function (sine); however, most edges are not periodic. Periodicity is not necessary, however, as the analysis applies *locally*. See Kemelmacher-Shlizerman et al.[15] for a more detailed analysis of the nonperiodic case.

5.2. Generalizing to 2D

Our analysis naturally generalizes to translations of 2D image edges. In case of 2D edge translation, we simply define our edge profiles in the direction normal to the edge, thus reducing to the 1D case.

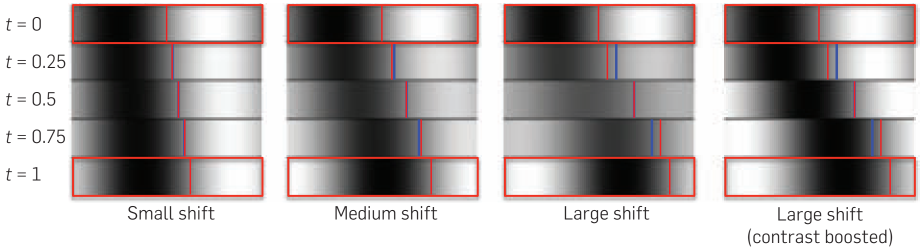
We have observed that cross dissolving two edges with different orientations produces a compelling apparent rotation perception (see video for examples), particularly when the orientation change is small and edge is low frequency. This effect is explained further in Kemelmacher-Shlizerman et al.[15]

5.3. Interpolation of light sources

In addition to edge motion, cross dissolves can also produce very convincing illumination changes in which the light source direction appears to move realistically during the transition. Indeed, we now describe conditions under which a cross dissolve produces physically-correct illumination changes.

An image of a Lambertian object, ignoring shadows, specularities, and inter-reflections, is determined by

Figure 6. Cross dissolve of sine and phased-shifted sine, with small (0.3π), medium (0.6π), and large (0.9π) shifts. We show film strips of the cross dissolve, with rows corresponding to times $t = 0$ (input frame), 0.25, 0.5, 0.75, and 1 (input frame). The location of the edge is marked in red and the location corresponding to a linear motion is marked in blue. The displacement of the red and blue lines for larger shifts demonstrates the nonlinear ease-in, ease-out speed curves (better seen in the video). Also note the decrease in contrast for larger shifts. To better visualize the (nonlinear) edge motion (for a 0.9π shift), we remove the contrast change (far right) by scaling the image by the inverse of c.



$I = \rho l^T n$, where ρ is the albedo, l is the lighting direction vector, and n is the surface normal vector. Cross dissolving two such images: $(1 - t)I_1 + tI_2$, has the effect of interpolating the lighting directions $\rho((1 - t)l_1 + tl_2)^T n$. In particular, we can rewrite the image formation equation as $I = \rho|l| \cos \phi$, where ϕ is the angle between the surface normal at each point on the surface and lighting direction.

A cross dissolve of two images can be then formulated as

$$(1 - t)\, \rho|l_1| \cos \phi + t\rho|l_2| \cos(\phi + d) = c_l \cos(\phi + k_l) \qquad (8)$$

with d being the difference between the surface normal and the two lighting direction angles. Hence, the interpolated pixel is the sum of two shifted cosines, which is also a cosine. In this case, however, the cosine is not in the image plane, but rather defines the variation of the pixel intensity as a function of lighting. The amplitude change c_l results in an effective *dimming* of the light during the transition, with minimum contrast occurring at the midpoint. This dimming effect serves to hide artifacts due to shadows, specular highlights, and other non-Lambertian effects that are not modeled by Equation (8). The cross dissolve thereby hides artifacts in the frames in which they are most likely to appear—a remarkable property!

While we are presenting the specific light trajectory of the cross dissolve (two-image) case for the first time, we emphasize that the basic result that image interpolations produce new directional illuminations of Lambertian objects is well known in the computer vision community, going back to the work of Shashua.[23]

6. RESULTS

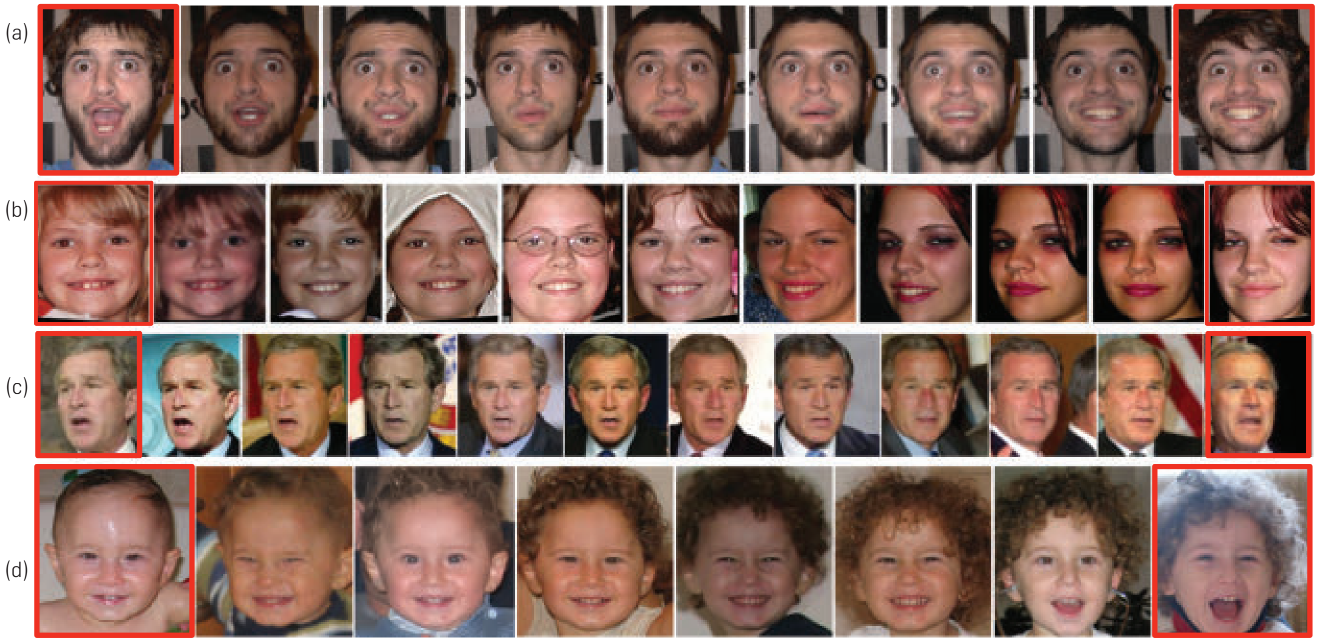
We experimented with datasets downloaded from the Internet, and with personal photo collections. Hundreds of Face Movies can also be found on YouTube, created by users of Picasa.

Most of our results are best viewed in the video (http://grail.cs.washington.edu/photobios/). We present a few example paths in Figure 7. We first experimented with time-lapse photo collections, in which a single person is photographed every week/day over a period of years, and usually include large variations in facial expression, hairstyle, etc. We show an example result on The "Daily Jason" dataset contains 1598 pictures taken almost every day during 5 years. Figure 7(a) shows an optimized path—the end points (marked in red) are chosen by the user in our interface and the intermediate sequence is computed by our method. Note the smooth transitions in mouth expression and eyes.

We have also experimented with personal photo collections: (1) 584 pictures of Amit over 5 years, (2) 1300 pictures of Ariel over 20 years, and (3) 530 photos of George W. Bush taken from the Labeled Faces in the Wild[11] collection. In contrast to the time-lapse datasets, the pictures in these three datasets were taken in arbitrary events, locations, with various illumination, resolution, cameras, etc., and are therefore more challenging. Figure 7(b, c, d) show typical results. Note how in all sequences, in addition to smooth transition in facial expression, the pose changes smoothly.

Examples of Face Movies created by people can be found on YouTube, here are links to a couple of our

Figure 7. Example paths produced with our method using different types of datasets: (a) time-lapse photo collection of Jason (1598 photos), (b) personal photo collection of Ariel over 20 years (1300 photos), (c) George W. Bush photo collection (530 photos), and (d) Amit's photos (584 photos over 5 years). The end points (marked in red) were chosen by the user and all the intermediate pictures were selected automatically by our method. Note the smooth transition in facial expression as well as pose.



favorites: http://www.youtube.com/watch?v=lydaVvF3fWI and http://www.youtube.com/watch?v=q9h7rGmFxJs.

6.1. Rendering style

For the Picasa implementation, we found that people prefer seeing more of the photos beyond just the cropped faces, as the wider field of view provides more context, and instead of showing photos one at a time, we layer the aligned photos over one another as shown in Figure 8. The user interface also provides the ability to output the movie, upload to the Web, or add audio, captions, and customize appearance in several other ways.

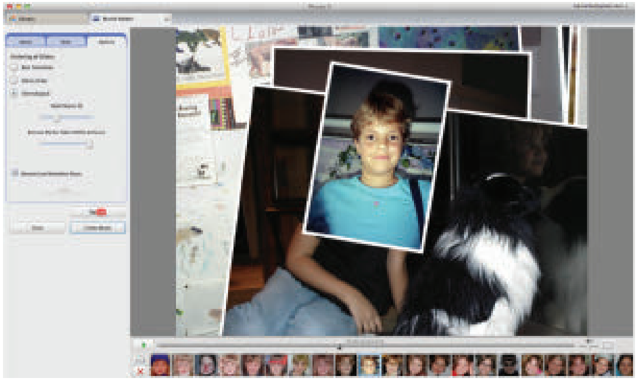
7. CONCLUSION

We presented a new technique for creating animations of real people through time, pose, and expression, from large unstructured photo collections. The approach leverages computer vision techniques to compare, align, and order face images to create pleasing paths, and operates completely automatically. The popular photo browsing tool Picasa has an implementation of this approach, known as "Face Movies," which has seen widespread deployment. Key to the success of this method is the use of the *cross dissolve*, which produces a strong physical motion and illumination change sensation when used to blend well-aligned images. We analyzed this effect and its operating range, and showed that, surprisingly, cross dissolves do indeed synthesize true edge motion and lighting changes under certain conditions.

Acknowledgments

This work was supported in part by National Science Foundation grant IIS-0811878, the University of Washington Animation Research Labs, Adobe, Google, and Microsoft. We thank the following people for the use of their photos: Amit Kemelmakher, Ariel McClendon, David Simons, Jason Fletcher, and George W. Bush. We also thank Todd Bogdan for his help with the Picasa implementation. Pictures of George W. Bush are used with permission by Reuters (photos 2, 4, 8 in Figure 1; Figure 7(c) photos 1, 4, 5, 8, 10, 11, 12) and by AP Photo (photos 1, 3, 5, 6, 7 in Figure 1; Figure 7(c) photos 2, 3, 6, 7, 9).

Figure 8. In Picasa, the images are aligned and displayed by stacking them over one another.



References

1. Ahonen, T., Hadid, A., Pietikäinen, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 12 (2006), 2037–2041.
2. Arikan, O., Forsyth, D.A. Interactive motion generation from examples. *ACM Trans. Graph. 21*, 3 (2002), 483–490.
3. Beier, T., Neely, S. Feature-based image metamorphosis. *ACM Trans. Graph. (SIGGRAPH)* (1992), 35–42.
4. Berg, T.L., Berg, A.C., Edwards, J., Maire, M., White, R., Teh, Y.W., Learned-Miller, E., Forsyth, D.A. Names and faces in the news. In *CVPR* (2004), 848–854.
5. Bourdev, L., Brandt, J. Robust object detection via soft cascade. In *CVPR* (2005).
6. Bregler, C., Covell, M., Slaney, M. Video rewrite: Driving visual speech with audio. *ACM Trans. Graph. (SIGGRAPH)* (1997), 75–84.
7. Chen, S.E., Williams, L. View interpolation for image synthesis. *ACM Trans. Graph. (SIGGRAPH)* (1993), 279–288.
8. Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. In *CVPR* (2005), 886–893.
9. Everingham, M., Sivic, J., Zisserman, A. "Hello! My name is… Buffy"—Automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference* (2006).
10. Goldman, D.B., Gonterman, C., Curless, B., Salesin, D., Seitz, S.M. Video object annotation, navigation, and composition. In *UIST* (2008), 3–12.
11. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49. University of Massachusetts, Amherst, 2007.
12. Joshi, N., Szeliski, R., Kriegman, D.J. PSF estimation using sharp edge prediction. In *CVPR* (2008).
13. Katz, S., Tal, A., Basri, R. Direct visibility of point sets. *ACM Trans. Graph. (SIGGRAPH 2007) 26*, 3 (2007).
14. Kemelmacher-Shlizerman, I., Sankar, A., Shechtman, E., Seitz, S.M. Being John Malkovich. In *ECCV* (2010).
15. Kemelmacher-Shlizerman, I., Shechtman, E., Garg, R., Seitz, S.M. Exploring photobios. *ACM Trans. Graph. 30*, 4 (2011), 61:1–61:10.
16. Kovar, L., Gleicher, M., Pighin, F. Motion graphs. *ACM Trans. Graph. (SIGGRAPH)* (2002), 473–482.
17. Lasseter, J. Principles of traditional animation applied to 3D computer animation. *ACM Trans. Graph. (SIGGRAPH)*(1987), 35–44.
18. Levoy, M., Hanrahan, P. Light field rendering. *ACM Trans. Graph. (SIGGRAPH)* (1996), 31–42.
19. Marr, D., Hildreth, E. Theory of edge detection. *Proc. R. Soc. Lond. B 207* (1980), 187–217.
20. Nalwa, V.S., Binford, T.O. On detecting edges. *IEEE Trans. Pattern Anal. Mach. Intell. 8*, (1986), 699–714.
21. Picasa, 2010. http://googlephotos.blogspot.com/2010/08/picasa-38-face-movies-picnik.html.
22. Seitz, S.M., Dyer, C.R. View morphing. *ACM Trans. Graph. (SIGGRAPH)* (1996), 21–30.
23. Shashua, A. Geometry and photometry in 3D visual recognition. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA (1992).
24. Szeliski, R., Shum, H.Y. Creating full view panoramic image mosaics and environment maps. *ACM Trans. Graph. (SIGGRAPH)* (1997), 251–258.
25. Zhang, L., Snavely, N., Curless, B., Seitz, S.M. Spacetime faces: High resolution capture for modeling and animation. *ACM Trans. Graph. (SIGGRAPH)* (2004), 548–558.

Ira Kemelmacher-Shlizerman (kemelmi@cs.washington.edu), University of Washington.

Eli Shechtman (elishe@adobe.com), Adobe Inc.

Rahul Garg (rahul@cs.washington.edu), Google Inc.

Steven M. Seitz (seitz@cs.washington.edu), University of Washington and Google Inc.

CAREERS

Bentley University

Tenure Track Assistant/Associate Professor, Computer Information Systems

The Department of Computer Information Systems at Bentley University invites applications for a tenure track position at the assistant or associate professor level starting in the fall of 2015. The department is seeking an individual with a PhD in Information Systems, Computer Science, or a related field. A qualified candidate should have an excellent record of accomplishments in both research and teaching and will be expected to contribute to the CIS department's mission of educating technically adept information systems professionals and advancing the state of research at the intersection of business and information and communication technologies.

Located in Waltham, Massachusetts, just minutes from Boston, Bentley University is a dynamic community of leaders, teacher-scholars and creative thinkers, including approximately 4,000 undergraduate and 1,300 graduate students.

To apply, please go to: https://jobs.bentley.edu/. Apply URL: https://jobs.bentley.edu/applicants/Central?quickFind=52435 You will need to upload a curriculum vitae, a cover letter that includes a reference contact list of three people, a research statement and a teaching statement. Applicants are also encouraged to consider positions in the Information and Process Management department, posted at https://jobs.bentley.edu/. Review of applicants will start on September 15, 2014 and will continue until the position is filled. For questions, contact Dr. Monica Garfield (CISRecruiting@bentley.edu).

Carbonmade, LLC.

Data Engineer

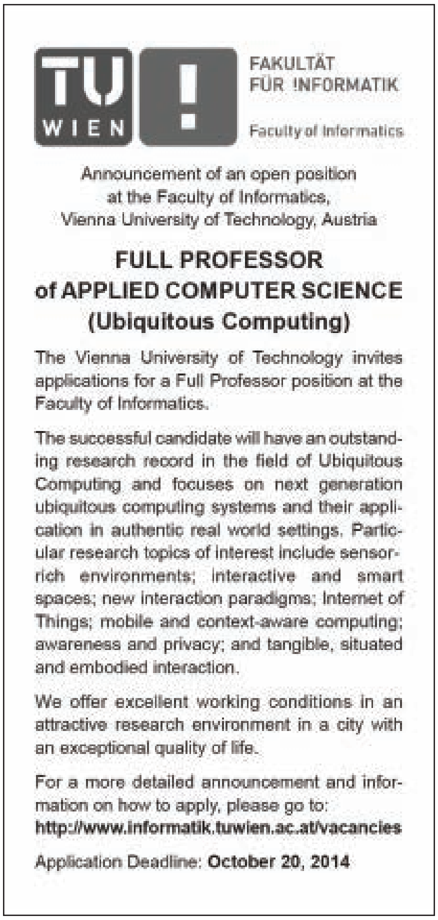
Data Engineer (Carbonmade, LLC, Chicago, IL). Dsgn, dvlp & modify software sys, using sci analysis & math models to predict & measure outcome & consequences of dsgn to optimize operational efficiency & improve data quality, search quality, & predictive capabilities. Write high perf production ready code in C++, Java, C#, or similar lang. MS in CS or Appl'd Math. Ed must incl crswrk in computa'l linear algebra & numerical methods. 2 years exp as Software Eng. Exp must incl: visualizing data & relationships; data anlys; OO program in C++, C#, Java, or similar lang; implementing math methods, algorithms; & statis'l anlys. Email resume to jason@carbonmade.com.

Grinnell College

Assistant Professor, Computer Science

GRINNELL COLLEGE. Tenure-track position in Computer Science starting Fall 2015. Asst Prof (PhD) preferred; Instructor (ABD) or Assoc Prof possible. Area open. Full details and application instructions: https://jobs.grinnell.edu. Candidates will upload letter of application, cv, transcripts (copies), teaching statement, description of scholarly activities, email addresses for three references. Questions: Prof Samuel A. Rebelsky, CSSearch@grinnell.edu, 641-269-3169. Deadline: Nov. 8, 2014.

AA/EOE



TU WIEN

FAKULTÄT FÜR INFORMATIK

Faculty of Informatics

Announcement of an open position
at the Faculty of Informatics,
Vienna University of Technology, Austria

FULL PROFESSOR
of APPLIED COMPUTER SCIENCE
(Ubiquitous Computing)

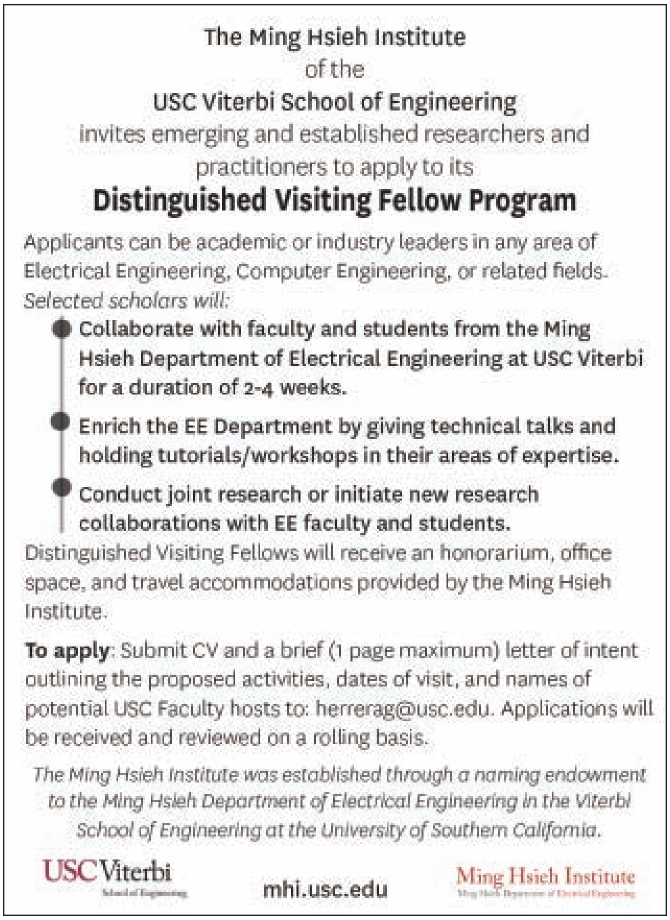
The Vienna University of Technology invites applications for a Full Professor position at the Faculty of Informatics.

The successful candidate will have an outstanding research record in the field of Ubiquitous Computing and focuses on next generation ubiquitous computing systems and their application in authentic real world settings. Particular research topics of interest include sensor-rich environments; interactive and smart spaces; new interaction paradigms; Internet of Things; mobile and context-aware computing; awareness and privacy; and tangible, situated and embodied interaction.

We offer excellent working conditions in an attractive research environment in a city with an exceptional quality of life.

For a more detailed announcement and information on how to apply, please go to:
http://www.informatik.tuwien.ac.at/vacancies

Application Deadline: **October 20, 2014**



The Ming Hsieh Institute
of the
USC Viterbi School of Engineering
invites emerging and established researchers and
practitioners to apply to its

Distinguished Visiting Fellow Program

Applicants can be academic or industry leaders in any area of Electrical Engineering, Computer Engineering, or related fields.
Selected scholars will:

- **Collaborate with faculty and students from the Ming Hsieh Department of Electrical Engineering at USC Viterbi for a duration of 2-4 weeks.**
- **Enrich the EE Department by giving technical talks and holding tutorials/workshops in their areas of expertise.**
- **Conduct joint research or initiate new research collaborations with EE faculty and students.**

Distinguished Visiting Fellows will receive an honorarium, office space, and travel accommodations provided by the Ming Hsieh Institute.

To apply: Submit CV and a brief (1 page maximum) letter of intent outlining the proposed activities, dates of visit, and names of potential USC Faculty hosts to: herrerag@usc.edu. Applications will be received and reviewed on a rolling basis.

The Ming Hsieh Institute was established through a naming endowment to the Ming Hsieh Department of Electrical Engineering in the Viterbi School of Engineering at the University of Southern California.

USC Viterbi School of Engineering

mhi.usc.edu

Ming Hsieh Institute

Portland State University

Faculty Position (Tenure-Track & Fixed-Term)

The Electrical and Computer Engineering (ECE) Department at Portland State University (PSU) seeks outstanding candidates for a tenure-track and non-tenure track fixed-term faculty in design verification/validation.

The ideal candidate for the tenure-track professor position has strong leadership skills, possesses a passion for teaching and research, has a demonstrated ability to attract and execute funded research, and is intimately familiar with current industry standards and needs. The candidate is expected to build and lead a strong and unique research program.

The ideal candidate for the non-tenure track fixed-term position has a strong passion for education along with proven teaching skills, and a desire to develop a compelling curriculum. The candidate must have significant industry experience in verification/validation. Expertise in hardware emulation is preferred.

Located in the heart of one of America's most progressive cities, PSU is Oregon's largest and most diverse public university. The downtown campus is a vibrant center of culture, business, and technology. Portland and the nearby Silicon Forest are hosts to many high-tech companies.

Additional information and requirements for applying are at https://jobs.hrc.pdx.edu . Positions #D93193 and #D93195. (Contact: Dr. Christof Teuscher; Email: teuscher@pdx.edu)

PSU is an Affirmative Action, Equal Opportunity institution and welcomes applications from diverse candidates and candidates who support diversity.

Technological Institute of the Philippines

Full Time Faculty Members with PhD. in Computer Science

Looking for a change and an interesting teaching career?

Visit: www.tip.edu

FULL TIME FACULTY MEMBERS with Ph.D. in Computer Science *who can assist in building world-class Computer Science, Information Systems, and Information Technology programs that will address and manage the rapid developments in Computer Science, Information Systems, and Information Technology* Competitive compensation package and fringe benefits await the qualified candidates

Send resume to milette.dolom@tip.edu.ph

The Technological Institute of the Philippines (TIP) is a private Philippine higher educational institution with campuses in Quezon City and Manila.

The computing programs of TIP in both campuses are accredited by the U.S.-based ABET Computing Accreditation Commission (CAC), the gold standard in computing education accreditation.

TECHNOLOGICAL INSTITUTE OF THE
PHILIPPINES
938 Aurora Blvd., Cubao, Quezon City
363 P. Casal Quiapo
Manila, Philippines

DOI:10.1145/2642744

Peter Winkler

Puzzled
Solutions and Sources

Last month (August 2014), we presented three puzzles concerning the Path Game and the Match Game, each of which can be played on any finite graph. To start, Alice marks a vertex; Bob and Alice then alternate marking vertices until one (the loser) is unable to mark any more. In the Path Game, each vertex thus marked, following the first one, must be adjacent to the most recently marked vertex. In the Match Game, only Bob has this constraint, whereas Alice can mark any vertex.

1. The Path Game. Bob can win the Path Game on an 8×8 checkerboard by imagining the board is covered by 32 dominoes, each occupying two adjacent squares. (There are many ways to do this; can you count them?) When Alice marks a square covered by a domino, Bob simply marks the square at the other end of that domino.

2. The Match Game. The same strategy works for Bob in the Match Game.

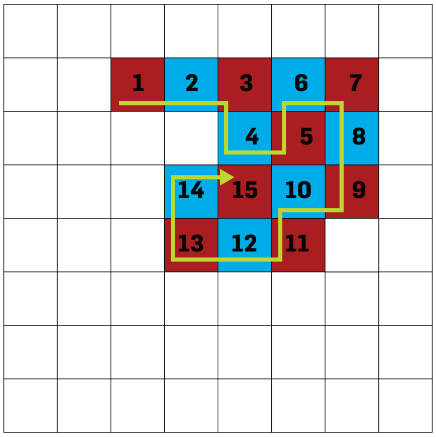
3. General Graphs. Amazingly, whenever Bob has a winning strategy for the Path Game, he can win the apparently more difficult Match Game as well. To win the Match Game, he needs the graphical equivalent of a domino cover—a "perfect matching." A matching in a graph is a collection of pairs of adjacent vertices, such that no vertex shows up in two pairs. A matching is deemed "perfect" if it uses up all the vertices; note this requires the graph to have an even number of vertices to begin with.

If the graph has a perfect matching, Bob easily wins either game by fixing some perfect matching and always marking the vertex paired with the one Alice just marked. The tricky part is to show that if there is no perfect matching, Alice can win the Path Game, and therefore the Match Game as well. To do this she chooses a maximum-size matching, and starts the game by picking a vertex not included in the matching. After that, she always picks the mate to Bob's last move, until poor Bob runs out of moves.

How do we know Alice always has such a move available? Suppose she is OK until she gets stuck after Bob's nth move, she and Bob having constructed the path a_1, b_1, a_2, b_2, . . . , a_n, b_n. The pairs $\{b_1,a_2\}$, $\{b_2,a_3\}$, . . . ,$\{b_n$–1,an$\}$ are then all in the matching, but b_n is unmatched.

However, this situation is not possible.

Why? Because then we could replace the n–1 pairs $\{b_1,a_2\}$, $\{b_2,a_3\}$, . . . ,$\{b_n-1,a_n\}$ with the n pairs $\{b_1,a_1\}$, $\{b_2,a_2\}$, . . . ,$\{b_n,a_n\}$ to get a bigger matching—contradicting the assumption that the matching Alice chose was of maximum size. So Alice always has a legal move, and thus it is Bob who eventually gets stuck and loses.



We conclude that Bob wins both games if the graph has a perfect matching; otherwise Alice wins both.

Want to know more about matchings? Any book on graph theory or computer algorithms will get you started, but if you prefer to tackle a whole book on the subject, see *Matching Theory* by László Lovász and Michael Plummer, North-Holland, Amsterdam, 1986.

All are encouraged to submit prospective puzzles for future columns to puzzled@cacm.acm.org.

Peter Winkler (puzzled@cacm.acm.org) is William Morrill Professor of Mathematics and Computer Science at Dartmouth College, Hanover, NH. .

[CONTINUED FROM P. 104] noticing qualitatively that they were very good, and being amazed by it. But we could never have foreseen the success.

What about in hindsight—what contributed to its success?

I can see a couple things. In our first attempt at developing LDA, we had a complicated algorithm, and after maybe two years of work, we figured out an easier way, which made everything much faster, so we could handle much larger datasets. We were also working on it at a time when probabilistic models from a machine learning perspective were starting to become more mainstream, and LDA became an example that people could latch onto of how someone can use probability models to analyze data. All this coincided with the rise of a lot of unstructured data—and the sudden need to organize all that data.

What are some of your favorite applications of LDA?

I find it amazing where people are using it. It's used in some recommendation systems. I heard from a friend that it's used in detecting credit-card fraud. One of my favorite uses is in the digital humanities, which have picked up on LDA and topic modeling in general. These are historians and English professors and folklore scholars, and they want to find patterns—recurring themes—that they wouldn't have otherwise seen.

So topic modeling isn't just for analyzing big data.

That's right. Imagine you're a historian, and you're studying 500 documents about a period of time—that's a modest number, but that's still a large number of documents to hold in your mind at a time. You can't ask the historian to give you a list of co-occurring words, because to a human it's not obvious, but it is to LDA. That's not to say the algorithm is doing a better job than a person, it's just an alternative view.

You also work on Bayesian nonparametric methods and approximate posterior inference.

They're all related. Let's go back to topic modeling: LDA assumes that there are a fixed number of topics and that documents exhibit multiple topics. We look at those assumptions, get a big document collection, and we ask: under those assumptions, what are the topics that best describe how this collection came about?

> **"Bayesian nonparametrics expands the language of modeling to make it much more flexible."**

What a Bayesian nonparametric model does, it expands what we can do with models. The number of topics is no longer held fixed, but grows in a natural way with the data: if there are a million documents, the Bayesian nonparametric model might find a thousand topics, and if there are more documents, it can add more topics if it needs them.

Bayesian nonparametrics can also be expanded to say that I think my documents have a tree of topics, which starts with general things like sports, and expands to specific things like hockey, baseball, etc. Now the problem of choosing the number of topics is much more difficult. Which tree structure do you choose? You have to try each possible tree structure, and it's an impossible task. But with Bayesian nonparametric methods we can say, "There's a tree, and I don't know what it looks like, but it's somehow responsible for explaining the data, so find me the tree structure as well." Bayesian nonparametrics expands the language of modeling to make it much more flexible.

And approximate posterior inference?

Matt Connelly, a historian at Columbia, studies the history of diplomacy, and he has a dataset of all the cables sent between different diplomatic stations in the '70s. He could use LDA to analyze it, or he could say, "I know something about this," and I can sit down with him and build a topic model based on what he knows about the data and what he wants to uncover in the data. Finding the hidden structure that this data exhibits is a computational problem called inference—the problem of computing the conditional distribution of the hidden variables given the observations. If you use LDA, the way the hidden variables look is different for each dataset—if you use LDA on *The New York Times*, the hidden topics look like "war," and "sports," and "weather," and so on—and if you use LDA on scientific articles, then you get "genetics" and "neuroscience" and "biology."

In the last few years in my group, we've been working on generic and scalable ways of doing inference. If Matt Connelly comes up with a model and it takes me two years to come up with the algorithm, he's going to lose interest. But if there's a procedure that works on lots of different models and quickly gives answers, then we'd be in business. That's approximate posterior inference.

What else are you working on?

We're also working on an interesting set of applied problems, often involving text but also other kinds of data. I'm collaborating with social scientists to study newspaper articles and build models of how perspectives on funding for culture change over time; working with biologists to study genetic data to understand the hidden ancestral populations exhibited in our genome, working with neuroscientists to study fMRI data to understand how text is somehow reflected in measurements of the brain while people are looking at words and texts.

Another big push in my group is user behavior data; we want to take data about which scientific articles people read and use that to understand how the articles are organized.

It's an extremely exciting time for machine learning because of the deluge of data. Statistics and machine learning haven't dealt with data of this size and this complexity, so I'm excited by the potential for doing exploratory data analysis. C

Based in San Francisco, **Marina Krakovsky** is the co-author of *Secrets of the Moneylab: How Behavioral Economics Can Improve Your Business.*

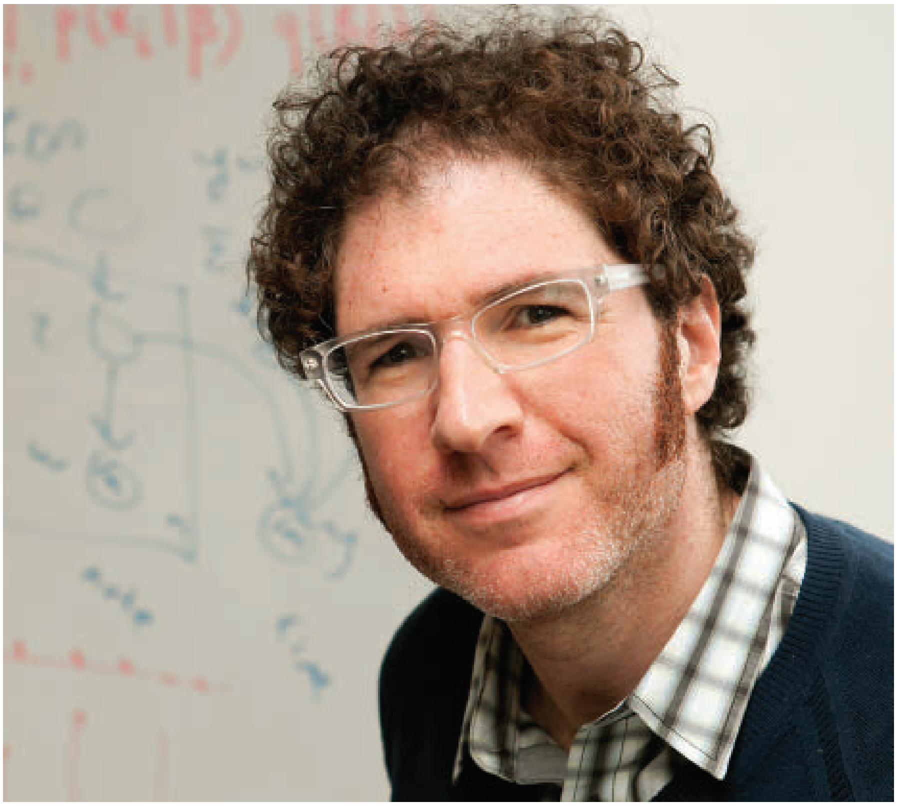
DOI:10.1145/2641223

Marina Krakovsky

Q&A
Finding Themes

ACM-Infosys Foundation Award recipient David Blei recalls the origins of his famous topic model, its extensions, and its uses in areas that continue to amaze him.

IN ANNOUNCING DAVID BLEI as the latest recipient of the ACM-Infosys Foundation Award in the Computing Sciences, ACM president Vint Cerf said Blei's contributions provided a basic framework "for an entire generation of researchers." Blei's seminal 2003 paper on latent Dirichlet allocation (LDA, co-authored with Andrew Ng and Michael Jordan while still a graduate student at the University of California, Berkeley) presented a way to uncover document topics within large bodies of data; it has become one of the most influential in all of computer science, with more Google Scholar citations than, for example, the earlier PageRank paper that launched Google. Blei's approach and its extensions have found wide-ranging uses in fields as varied as e-commerce, legal pretrial discovery, literary studies, history, and more. At Princeton since 2006, Blei begins this fall as a professor at Columbia University, with joint appointments in the two departments his work bridges, computer science and statistics.



PHOTOGRAPH BY DENISE APPLEWHITE, COURTESY OF PRINCETON UNIVERSITY OFFICE OF COMMUNICATIONS

How did the idea for LDA come about?

The idea behind topic modeling is that we can take a big collection of documents and learn that there are topics inside that collection—like sports or health or business—and that some documents exhibit a pattern of words around that topic and others don't. That idea really began in the late '80s and early '90s with latent semantic analysis (LSA), from people like Susan Dumais at Microsoft Research. Then Thomas Hoffman developed probabilistic latent semantic analysis (PLSA), taking the ideas of LSA and embedding them in a probability model.

This story is going to sound anticlimactic: I went to an internship to COMPAQ Research in Boston, and there I was working with Pedro Moreno; we were running speech recognizers on talk-radio programs, and then using PLSA on that noisy text output. I came back to Berkeley, where my officemate was Andrew Ng, and I told him I'd worked with the PLSA-based model. He said, "Something has always bothered me about PLSA." It was a technical thing, but as a consequence we wrote down three models, and LDA was one of them. In going from PLSA to LDA, the main difference is LDA could be used in a larger, more complicated model.

Did you have a sense that this was going to be big?

No. I did have an immediate sense that I enjoyed working on it: I remember fitting a model and seeing the topics pop out, and

[CONTINUED ON P. 103]

BIG IDEAS START SMALL

SIGGRAPH
ASIA 2014
SHENZHEN

CONFERENCE 3 DEC - 6 DEC
EXHIBITION 4 DEC - 6 DEC
SHENZHEN CONVENTION
& EXHIBITION CENTER
SA2014.SIGGRAPH.ORG

REGISTER NOW TO SAVE

At SIGGRAPH Asia, meet the people you want to meet. Get to learn, be enthralled and inspired by quality content and exhibits in ways you will never be at work or school.

From now until **15 October 2014, 15:00 Shenzhen, China time** we offer you early bird discounts of up to **20%** if you register online.

Complete registration details can be found at **sa2014.siggraph.org/registration-travel.**

Sponsored by

acm

In Cooperation with

宽带网数字媒体技术实验室

Supported by

SIAT
中国科学院深圳先进技术研究院
SHENZHEN INSTITUTES OF ADVANCED TECHNOLOGY
CHINESE ACADEMY OF SCIENCES

VR 虚拟现实技术与系统国家重点实验室
STATE KEY LABORATORY OF VIRTUAL REALITY TECHNOLOGY AND SYSTEMS

Tsinghua-Tencent
清华-腾讯联合实验室

VCC