

COMMUNICATIONS

CACM.ACM.ORG OF THE ACM 04/2018 VOL.61 NO.04



Building a Smart City: Lessons from Barcelona

DevOps Delivers

Bridgeware

Realizing the Potential
of Data Science

Go BIG!

Open Access and ACM

Association for
Computing Machinery





10th ACM International Conference on

Automotive User Interfaces and Interactive Vehicular Applications

September 23-25, 2018, Toronto, Canada
with Doctoral Colloquium on September 22

CONFERENCE CHAIR

Birsen Donmez
University of Toronto, Canada
chair2018@auto-ui.org

PROGRAM CHAIRS

Bruce Walker
Georgia Tech, USA
papers2018@auto-ui.org

Peter Froehlich
Austrian Institute of Technology
papers2018@auto-ui.org

Welcome to AutomotiveUI

AutomotiveUI is the premier forum for UI research in the automotive domain. Sponsored by ACM SIGCHI, it is the only international conference focusing on in-vehicle interaction technologies.

The conference highlights novel vehicle technologies through models and concepts for enhancing the driver experience, performance, and behavior, the development of semi and fully autonomous driving, and the needs of different user groups, including passengers and pedestrians.

The goal is to support the development of automotive user interfaces that are safe, easy to use, useful, and desired by users.

Over 200 attendees, both from academia and industry, come together from across the world within a forward looking perspective necessary to continue supporting the road users of today and the future. Automotive UI is a great place to exchange ideas with researchers and practitioners, as well as to meet leading experts in the field, researchers and students who will form the next generation of experts in Automotive UI.

Please join us in Toronto for AutomotiveUI!

Important Dates



Full Papers: April 26, 2018

Workshop & Tutorial Proposals: June 4, 2018

Work In Progress: July 11, 2018

Interactive Demos: July 11, 2018

Videos: July 11, 2018

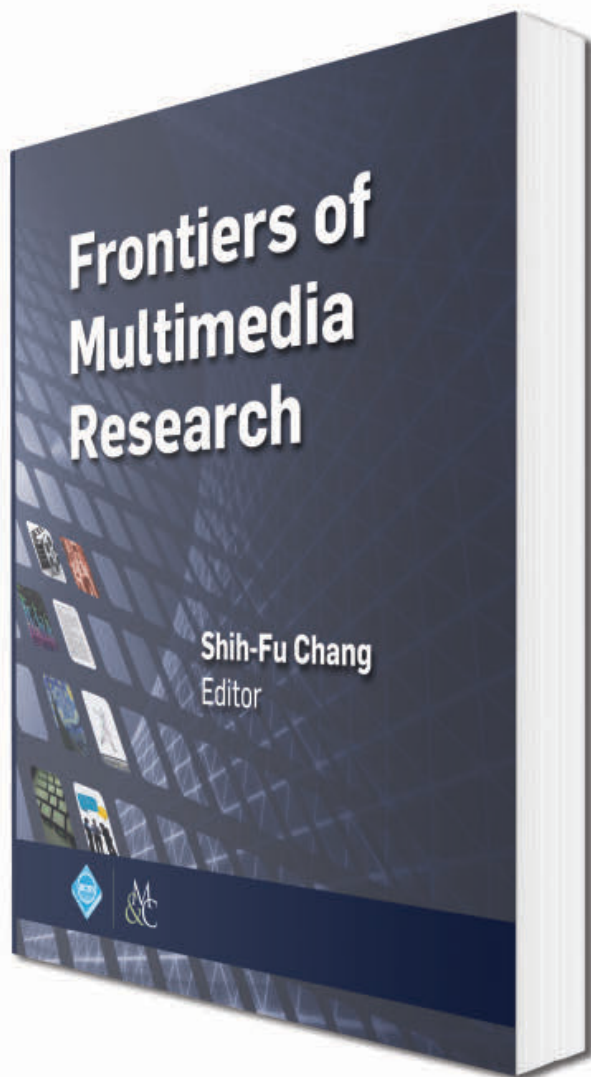
Doctoral Colloquium: July 11, 2018

All deadlines are 11:59pm AoE (anywhere on earth) on the date shown

auto-ui.org

  #AutoUI





12 rising-stars from different subfields of multimedia research discuss the challenges and state-of-the-art developments of their prospective research areas in a general manner to the broad community.

Shih-Fu Chang, Editor
Columbia University

The field of multimedia is unique in offering a rich and dynamic forum for researchers from “traditional” fields to collaborate and develop new solutions and knowledge that transcend the boundaries of individual disciplines. Despite the prolific research activities and outcomes, however, few efforts have been made to develop books that serve as an introduction to the rich spectrum of topics covered by this broad field. A few books are available that either focus on specific subfields or basic background in multimedia. Tutorial-style materials covering the active topics being pursued by the leading researchers at frontiers of the field are currently lacking...until now.

Each chapter discusses the problems, technical challenges, state-of-the-art approaches and performances, open issues, and promising direction for future work. Collectively, the chapters provide an excellent sampling of major topics addressed by the community as a whole. This book, capturing some of the outcomes of such efforts, is well positioned to fill the aforementioned needs in providing tutorial-style reference materials for frontier topics in multimedia.



ISBN: 978-1-970001-044 DOI: 10.1145/3122865
<http://books.acm.org>
<http://www.morganclaypoolpublishers.com/chang>

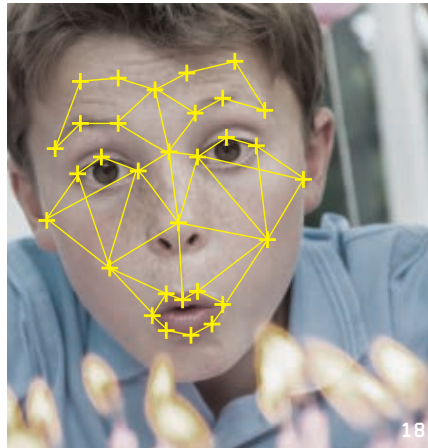
Departments

- 5 **Editor's Letter**
Go BIG!
By Andrew A. Chien
-
- 6 **Cerf's Up**
The Sound of Programming
By Vinton G. Cerf
-
- 7 **Vardi's Insights**
Open Access and ACM
By Moshe Y. Vardi
-
- 8 **Letters to the Editor**
Predicting Failure of the University
-
- 10 **BLOG@CACM**
Fostering Inclusion, Keeping the Net Neutral
ACM-W chair Jodi Tims offers ways everyone can promote inclusiveness, while Daniel A. Reed assesses the debate over Net neutrality.
-
- 29 **Calendar**
-
- 94 **Careers**

Last Byte

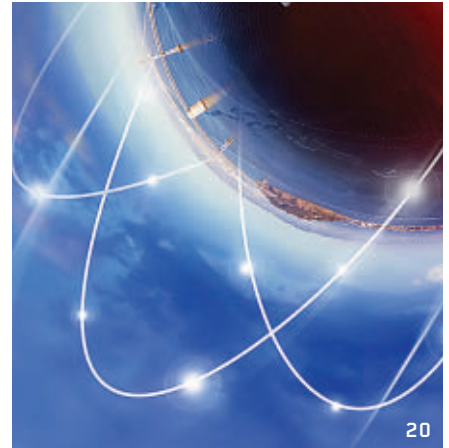
- 96 **Upstart Puzzles**
Finding October
By Dennis Shasha

News



- 12 **Always Out of Balance**
Computational theorists prove there is no easy algorithm to find Nash equilibria, so game theory will have to look in new directions.
By Neil Savage
-
- 15 **Chips for Artificial Intelligence**
Companies are racing to develop hardware that more directly empowers deep learning.
By Don Monroe
-
- 18 **Artificial (Emotional) Intelligence**
Enabled by advances in computing power and neural networks, machines are getting better at recognizing and dealing with human emotions.
By Marina Krakovsky

Viewpoints



- 20 **Technology Strategy and Management**
Business Ecosystems: How Do They Matter for Innovation?
Considering the significant interrelationship of innovation, corporate strategy, and public policy for business ecosystems.
By Mari Sako
-
- 23 **Kode Vicious**
Popping Kernels
Choosing between programming in the kernel or in user space.
By George V. Neville-Neil
-
- 25 **Viewpoint**
Push Versus Pull
Flipping the publishing business model.
By Sheldon H. Jacobson
-
- 28 **Viewpoint**
Smartphones, Contents of the Mind, and the Fifth Amendment
Exploring the connection qualities between smartphones and their users.
By Stephen B. Wicker



Practice



44

32 **DevOps Delivers**
By Nicole Forsgren

34 **Continuous Delivery Sounds Great, but Will It Work Here?**
It's not magic, it just requires continuous, daily improvement at all levels.
By Jez Humble

40 **Containers Will Not Fix Your Broken Culture (and Other Hard Truths)**
Complex socio-technical systems are hard; film at 11.
By Bridget Kromhout

44 **DevOps Metrics**
Your biggest mistake might be collecting the wrong data.
By Nicole Forsgren and Mik Kersten



Articles' development led by acmqueue.queue.acm.org

Contributed Articles

50 **Building a Smart City: Lessons from Barcelona**
Smart Internet-based infrastructure is one thing but will be ignored without the public's continuing engagement.

By Mila Gascó-Hernandez



Watch the author discuss her work in this exclusive *Communications* video.
<https://cacm.acm.org/videos/building-a-smart-city>

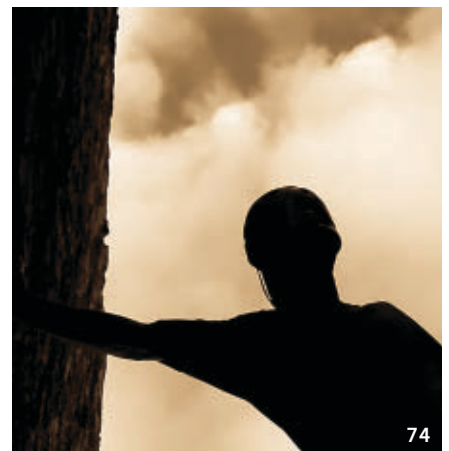
58 **Lessons from Building Static Analysis Tools at Google**
For a static analysis project to succeed, developers must feel they benefit from and enjoy using it.
By Caitlin Sadowski, Edward Aftandilian, Alex Eagle, Liam Miller-Cushon, and Ciera Jaspán

67 **Realizing the Potential of Data Science**
Data science promises new insights, helping transform information into knowledge that can drive science and industry.
By Francine Berman, Rob Rutenbar, Brent Hailpern, Henrik Christensen, Susan Davidson, Deborah Estrin, Michael Franklin, Margaret Martonosi, Padma Raghavan, Victoria Stodden, and Alexander S. Szalay



Watch the authors discuss their work in this exclusive *Communications* video.
<https://cacm.acm.org/videos/realizing-the-potential-of-data-science>

Review Articles



74

74 **Bridgewater: The Air-Gap Malware**
The challenge of combatting malware designed to breach air-gap isolation in order to leak data.
By Mordechai Guri and Yuval Elovici

Research Highlights

84 **Technical Perspective**
Expressive Probabilistic Models and Scalable Method of Moments
By David M. Blei

85 **Learning Topic Models — Provably and Efficiently**
By Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu



About the Cover:
A smart bus stop on a quiet street in Barcelona is just one of many ways this city, considered one of the top smart cities of the world, has employed technologies to foster economic growth and to benefit its citizens. This month's cover story (p. 50) examines what makes Barcelona so smart. Cover photo by Gaudi Lab.



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.

Acting Executive Director
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
 Scott E. Delman

ACM COUNCIL

President
 Vicki L. Hanson
Vice-President
 Cherri M. Pancake
Secretary/Treasurer
 Elizabeth Churchill
Past President
 Alexander L. Wolf
Chair, SGB Board
 Jeanna Matthews
Co-Chairs, Publications Board
 Jack Davidson and Joseph Konstan
Members-at-Large
 Gabriele Anderst-Kotis; Susan Dumais;
 Elizabeth D. Mynatt; Pamela Samuelson;
 Eugene H. Spafford
SGB Council Representatives
 Paul Beame; Jenna Neefe Matthews;
 Barbara Boucher Owens

BOARD CHAIRS

Education Board
 Mehran Sahami and Jane Chu Prey
Practitioners Board
 Terry Coatta and Stephen Ibaraki

REGIONAL COUNCIL CHAIRS

ACM Europe Council
 Chris Hankin
ACM India Council
 Madhavan Mukund
ACM China Council
 Yunhao Liu

PUBLICATIONS BOARD

Co-Chairs
 Jack Davidson; Joseph Konstan
Board Members
 Phoebe Ayers; Anne Condon; Nikil Dutt;
 Roch Guerrin; Chris Hankin;
 Yannis Ioannidis; XiangYang Li;
 Sue Moon; Michael L. Nelson;
 Sharon Oviatt; Eugene H. Spafford;
 Stephen N. Spencer; Alex Wade;
 Julie R. Williamson

ACM U.S. Public Policy Office

Adam Eisgrau,
 Director of Global Policy and Public Affairs
 1701 Pennsylvania Ave NW, Suite 300,
 Washington, DC 20006 USA
 T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association
 Jake Baskin
 Executive Director

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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

STAFF

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

Executive Editor
 Diane Crawford
Managing Editor
 Thomas E. Lambert
Senior Editor
 Andrew Rosenbloom
Senior Editor/News
 Lawrence M. Fisher
Web Editor
 David Roman
Rights and Permissions
 Deborah Cotton
Editorial Assistant
 Jade Morris

Art Director
 Andrij Borys
Associate Art Director
 Margaret Gray
Assistant Art Director
 Mia Angelica Balaquiot
Production Manager
 Bernadette Shade
Advertising Sales Account Manager
 Iliia Rodriguez

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@hq.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/about-communications/author-center>

ACM ADVERTISING DEPARTMENT

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

Advertising Sales Account Manager
 Iliia Rodriguez
 ilia.rodriguez@hq.acm.org

Media Kit acmm mediasales@acm.org

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

EDITORIAL BOARD

EDITOR-IN-CHIEF
 Andrew A. Chien
 eic@cacm.acm.org

Deputy to the Editor-in-Chief
 Lihan Chen
 cacm.deputy.to.eic@gmail.com

SENIOR EDITOR

Moshe Y. Vardi

NEWS

Co-Chairs
 William Pulletyblank and Marc Snir
Board Members
 Monica Divitini; Mei Kobayashi;
 Michael Mitzenmacher; Rajeev Rastogi;
 François Sillion

VIEWPOINTS

Co-Chairs
 Tim Finin; Susanne E. Hambrusch;
 John Leslie King; Paul Rosenbloom
Board Members
 Stefan Bechtold; Michael L. Best; Judith Bishop;
 Andrew W. Cross; Mark Guzdial; Haym B. Hirsch;
 Richard Ladner; Carl Landwehr; Beng Chin Ooi;
 Francesca Rossi; Loren Terveen;
 Marshall Van Alstyne; Jeannette Wing

PRACTICE

Chair
 Stephen Bourne and Theo Schlossnagle
Board Members
 Eric Allman; Samy Bahra; Peter Bailis;
 Terry Coatta; Stuart Feldman; Nicole Forsgren;
 Camille Fournier; Benjamin Fried;
 Pat Hanrahan; Tom Killalea; Tom Limoncelli;
 Kate Matsudaira; Marshall Kirk McKusick;
 Erik Meijer; George Neville-Neil;
 Jim Waldo; Meredith Whittaker

CONTRIBUTED ARTICLES

Co-Chairs
 James Larus and Gail Murphy
Board Members
 William Aiello; Robert Austin; Kim Bruce;
 Alan Bundy; Peter Buneman; Carl Gutwin;
 Yannis Ioannidis; Gal A. Kaminka;
 Ashish Kapoor; Kristin Lauter; Igor Markov;
 Bernhard Nebel; Lionel M. Ni; Adrian Perrig;
 Marie-Christine Rousset; Krishan Sabnani;
 m.c. schraefel; Ron Shamir; Alex Smola;
 Josep Torrellas; Sebastian Uchitel;
 Hannes Werthner; Reinhard Wilhelm

RESEARCH HIGHLIGHTS

Co-Chairs
 Azer Bestavros and Shriram Krishnamurthi
Board Members
 Martin Abadi; Amr El Abbadi; Sanjeev Arora;
 Michael Backes; Maria-Florina Balcan;
 Andrei Broder; David Brooks; Doug Burger;
 Stuart K. Card; Jeff Chase; Jon Crowcroft;
 Alexei Efros; Alon Halevy; Gernot Heiser;
 Sven Koenig; Steve Marschner;
 Greg Morrisett; Tim Roughgarden;
 Guy Steele, Jr.; Robert Williamson;
 Margaret H. Wright; Nicolai Zeldovich;
 Andreas Zeller

WEB

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

ACM Copyright Notice

Copyright © 2018 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from permissions@hq.acm.org or fax (212) 869-0481.

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

Subscriptions

An annual subscription cost is included in ACM member dues of \$99 (\$40 of which is allocated to a subscription to *Communications*); for students, cost is included in \$42 dues (\$20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is \$269.

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 USA.



Association for
 Computing Machinery





Andrew A. Chien

DOI:10.1145/3192027

Go BIG!

I WAS FORTUNATE to enter computing in the era of site funding by the Defense Advanced Research Projects Administration (DARPA). “DARPA sites” from 1960s through the mid-1990s had sustained investment of \$10M/year (inflation adjusted). The talent and vision combined with sustained funding at scale enabled the undertaking of bold transformative ideas, such as timesharing, entire new operating systems, novel computing system architectures, new models of networking, and a raft of exciting artificial intelligence technologies—in robotics, self-driving cars, computer vision, and more. For example, I joined Arvind’s Tagged-token Dataflow Computing project as an MIT graduate student. This single project, which involved a dozen graduate and undergraduates plus staff and faculty, garnered ~\$13.5M support with a run rate exceeding \$4M/year.

Why is large-scale funding important? It enables examination of larger questions that cross problem spaces, systems, abstractions, even fields. Yes, we have open source software. Yes, we can build on large-scale frameworks and software systems. Yes, we can compose services and leverage the cloud. But it is important to recognize that leverage of such infrastructures often means that rethinking or reimagining them is beyond the scope of inquiry. I believe the need for such large investments has never been greater. We are seeing:

► Transformative change in architecture, operating systems, programming languages, and applications. New large-scale geographic and distributed structures—datacenters, edge, home, undersea, orbital systems and soon outer space.

► New applications from computing, data, intelligence, and trusted algorithmic execution (for example, blockchain) melding with a variety of fields—law, business, in new ways with accelerating benefit to society.

► Information technology in the body politic, journalism, and social fabric writ large—“fake news,” “social network addiction and depression,” and “societal manipulation.”

In the past decade, we have seen extraordinary new capabilities. GPU computing has enabled new levels of energy efficient, high performance, but only with radical software change (and of course architecture). The re-emergence of neuromorphic computing (including deep learning) delivered new artificial intelligence capabilities to a broad array of applications. And visionaries suggest many more eruptions are coming.¹

Numerous National Academy and committee reports have called for research “big bets” based on need and opportunity. And in late 2007, during my service on the NSF CISE Advisory Committee, then-CISE AD Jeannette Wing initiated the NSF CISE Expeditions program to create “big bets.”² A bright spot! But with a much smaller investment than prior programs, funding only 1.5 projects per year (each ~\$10M/five years). Expeditions are terrific projects, but typically split over several sites, diluting infrastructure, capability, and perspective. For example, a 2018 Expedition Award, “EpiQC: Enabling Practical-Scale Quantum Computation,” led by my colleague Fred Chong at the University of Chicago, includes five institutions.

Given industrial-scale R&D, does computing need large-scale government research investments? Yes! Academic research has fundamental

advantages for society, including education and broad idea and technology dissemination; openness to a wide range of possibilities and directions (not just “our business position”); vetted and publicly examined rigorously from a scientific point of view (for example, bias in algorithms, security architectures, privacy and exploitation); and vetted and publicly examined from a broad societal perspective (for example, social media, addictive technology).

So what would I propose?

Perhaps a new program. Perhaps a doubling, and doubling again of the Expeditions program. Projects of twice the scale (\$20M over five years) with mechanisms to ensure they are concentrated at no more than two institutions. And doubling the number of such efforts to three new starts every year. Too expensive? Such a program would be less than 1/5 of 1% of the NIH’s annual budget, and 1/100 of 1% of the U.S. Department of Defense budget.

Perhaps if we want to increase global economic growth beyond what economists call “structural limits,” we need more disruptive computing advances. *Carpe Diem!*

Andrew A. Chien, EDITOR-IN-CHIEF

Andrew A. Chien is the William Eckhardt Distinguished Service Professor in the Department of Computer Science at the University of Chicago, Director of the CERES Center for Unstoppable Computing, and a Senior Scientist at Argonne National Laboratory.

References

1. Conte, T.M., DeBenedictis, E.P., Gargini, P.A., and Track, E. Rebooting computing: The road ahead. *Computer* 50, 1 (2017), 20–29.
2. NSF Expeditions in Computing Program; www.nsf.gov

Copyright held by author.



Vinton G. Cerf

DOI:10.1145/3190858

The Sound of Programming

In the early days of digital computing, it was not uncommon to find a radio receiver tuned to a particular frequency (I don't recall which one, sigh) so that the RF emitted by the computer

could be picked up and *played* through the radio. You could tell when a program went into a loop and sometimes you could tell roughly where a computation had reached by the sounds coming from the radio monitor. Fast-forward to the 21st century and we are seeking a different kind of sound: the sound of programming.

Bootstrap World^a has developed online courses in programming, among other subjects, but what makes Bootstrap World so memorable for me is that the team has focused heavily on accessibility. The programming environment is extremely friendly to *screen readers* so that a blind programmer can navigate easily through complex programs using keyboard navigation coupled with oral descriptions/renderings of the program text and structure.^b A recent visit from Emmanuel Schanzer, founder of Bootstrap World, reinforced my positive impression of the group and Schanzer's enthusiasm and commitment to education for everyone.

As I watched and listened, Schanzer began *writing* a program using a visual language that is coupled with a concomitant audible description of where the programmer is in the program. The programming language encourages clean programming structure and that makes it possible for audible navigation without becoming hopelessly lost. Among the many

things I learned from this exercise is the ability of blind people to listen to speech that seems three to five times faster than normal speech. It reminded me of the radio commercials that draw you in with various audible gimmicks, pitch you, and then, at the end, a chipmunk voice announces all kinds of terms and conditions for accepting the offer. The rapid-fire rendering is essential for any timely navigation of the program or confirmation of changes to it.

My thoughts were drawn to the inverse situation in which the programmer might be able to narrate the program text and have the computer absorb and orally respond or confirm its understanding of the programmer's intent. At some point, one fantasizes a dialog with an intelligent agent that makes comments about bugs and mistakes ("You screwed up again, Einstein! That's a buffer overflow!"). Whether we ever get to that point or not, the idea that one could get semantic or cognitive help from a sophisticated computer program to help write more sophisticated computer programs is very attractive. The same structuring that makes it easier to navigate audibly might also make the program more easily analyzed for flaws.

What seems more important about the work at Bootstrap World is the potential to provide students closer to STEM learning with tools that move at the same pace at which the students can move. The notion of self-paced learning has a great deal of attraction.

People learn at different speeds and use different methods to reinforce what they are learning. The Bootstrap World program is designed for this kind of adaptability and flexibility. At a time when science and technology need an increasing population of STEM-educated workers, the traditional methods of four-year college and perhaps graduate study may not be optimal. With longer lives and longer careers, it is likely we will all need to return to school or at least to enter a learning phase more than once in our careers. Adaptable and convenient online learning is sure to be a part of 21st-century careers. It is no longer feasible or even sensible to try to pack all one needs for a lifetime of work into a few years at the beginning of our lives.

One final note: I am persuaded more than ever that we learn best by trying to do something, perhaps failing, getting some guidance, and trying again. This seems like a form of *just in time* learning not unlike what I experience when I write these columns. I get stuck somewhere (usually more than once) and turn to Google or YouTube to find out *how* or *what* from these media. That sounds like a wave of the future to me. □

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

a www.bootstrapworld.org

b <http://www.bootstrapworld.org/blog/accessibility/User-Interface-REPL.shtml>



Moshe Y. Vardi

DOI:10.1145/3191676

Open Access and ACM

THE BUDAPEST OPEN ACCESS Initiative was issued on Feb. 14, 2002. It argued that “An old tradition and a new technology have converged to make possible an unprecedented public good. ... The public good they make possible is the worldwide electronic distribution of the peer-reviewed journal literature and completely free and unrestricted access to it.” I was immediately captivated by the intellectual elegance of the open access proposition, which aims at removing barriers to scholarly knowledge. In 2005, together with colleagues, I launched a new open access journal, *Logical Methods in Computer Science* (LMCS),^a which to this very day is free to both readers and authors.

But it did not take long for me to realize there is a problem with the business model of “free to both readers and authors”; that is, there are expenses involved in publishing, but “free” means that there is no income! LMCS is run on a shoestring budget, but this model is not scalable. When I became EiC of *Communications* in 2008, I realized how expensive publishing is. According to the latest ACM Annual Report,^b ACM’s total publication expenses in fiscal year 2016 were close to US\$11M! Furthermore, ACM’s publication revenues were close to US\$21M. The surplus of US\$10M helps fund numerous ACM activities—in education, professional development, public policy, and more. Clearly, ACM’s publication business model must be consistent with these realities.

In the standard business model for open access publications, authors of published articles are required to pay article-process charges (APCs), which range from hundreds of dollars to thou-

sands of dollars. The fact that the monetary transaction of publishing is between the publishers and the authors rather than between publishers and institutions gave rise to the phenomenon of *predatory publishing*,^c an ugly, unintended consequence of the open access movement.

And yet, in spite of all the complications of open access publishing, the emerging sense of the scientific community is that science publishing should be done under an open access model, which means that articles should be available to readers without charge. ACM is under significant pressure from its membership to move from a subscription-based publishing model to an open access publishing model. Such a transition is exceedingly challenging. A significant drop in ACM’s publishing revenue, which would threaten ACM’s financial viability, is a risk that must be taken seriously. ACM must engage with its membership to develop and carry out such a transition plan, yet ACM has an obligation to manage such a transition in a way that protects the organization’s financial viability and vibrancy.

In his Viewpoint published in this issue—“Push Versus Pull: Flipping the Publishing Business Model” (p. 25)—Sheldon Jacobson proposes an alternative business model for open access publishing. In the traditional publishing business model, institutions paid publishers reader-subscription fees to enable their employees and students access to published content. In Jacobson’s flipped model, institutions will pay author-subscription fees to enable their employees and students to *submit* articles for publication. This can be viewed as a variation on the APC-based model, but it takes the authors out of the monetary transaction and recasts it again as

a transaction between institutions and publishers. This offers a solution to the predatory-publishing problem, as institutions are less likely to subscribe to publications of low scholarly quality.

The most significant feature of this proposal is that it involves a significant shifting of publishing costs to research-intensive institutions. The ACM Digital Library (DL) has about 2,500 subscribing institutions, but only about 500 of these institutions publish more than 10 papers annually in the DL, so on the average, author-subscription fees would be about five times higher than reader-subscription fees. But such average-case analysis is misleading. As one would expect, a relatively small number of institutions are “heavy publishers.” Jacobson proposes dividing institutions into tiers: Tier I consists of the 70 institutions that publish more than 44 articles per year, while Tier II consists of 850 institutions that publish 6–44 articles per year. His proposal is for Tier I institutions to pay US\$16K for annual author-subscription, and for Tier II institutions to pay US\$10K annually. Other institutions would pay a US\$500 *submission* fee per article.

As Jacobson points out, the proposed new model may be viewed as unfair to research institutions, but the shifting of publishing costs from readers to authors is inherent in the open access model. Regardless of the weaknesses in Jacobson’s proposal, he has opened the door to a substantive discussion about ACM’s publishing business model. If we are serious about open access, then we must discuss its underlying business model. Let’s get serious about open access!

Follow me on Facebook, Google+, and Twitter. 

Moshe Y. Vardi (vardi@cs.rice.edu) is the Karen Ostrum George Distinguished Service Professor in Computational Engineering and Director of the Ken Kennedy Institute for Information Technology at Rice University, Houston, TX, USA. He is the former Editor-in-Chief of *Communications*.

Copyright held by author.

a <https://lmcs.episciences.org/>

b <https://goo.gl/fHeZXs>

c <https://goo.gl/zUqP4r>

Predicting Failure of the University

HENRY C. LUCAS, JR.'S Viewpoint "Technology and the Failure of the University" (Jan. 2018) was a tour de force of unfounded assertions, beginning with a prediction that 50% of U.S. universities will fail in the next 15 years, justified by two articles posted on the blogs <http://www.futuristspeaker.com/> and <https://www.zerohedge.com/>. (The latter is described by RationalWiki as "apocalypse porn" that has "accurately predicted 200 of the last 2 recessions.") Lucas asserted "... technology-enhanced teaching and learning can dramatically improve the quality and success of higher education . . ." His Figure 1 and Figure 2, in outlining traditional versus technology-enhanced courses, suggested traditional teaching methods deliver a low-quality result, while professional (Hollywood) production methods deliver a high-quality result, with, again, no evidence provided.

The idea of universities as "content producers" giving students "content" consisting of "course materials and exercises" gave me an analogous idea. Families give food and clothing to their children, but families are inefficient and can involve bloated administrations (parents). Just as parents do more than feed (they try to create an environment where their children can develop and thrive), universities likewise try to create a learning environment for students. Indispensable elements include laboratory work, fieldwork, real essays marked by real scholars (not against a list of bullet points), and project work. And that is only the strictly academic side. Also indispensable are cultural and social events and pastoral care. A lot of these things are not easily delivered through a mobile device.

We should be willing to embrace new technology but must remember technology is not always beneficial. For example, during the 1990s, PowerPoint was touted as an efficient means of delivering content, but, alas, PowerPoint turned out to be soporific. The content was delivered, but the consumers were half asleep.

Lucas's section labeled "Threats to Technology" included a helpful list of practices that "create a huge barrier to adopting new technologies for education," including assistant professors "publishing scholarly articles and books" (the horror!), tenured faculty who can largely do what they want (news to me), and "faculty governance" (another horror—democracy). What we need apparently is "a fearless dean" (perhaps a fan of zerohedge.com) to impose unwanted technology on the faculty and expect them to embrace it with enthusiasm.

The funny thing is, you might think new technology would cut costs; the cost of higher education has reached alarming levels. But, Lucas said, ". . . it requires substantial resources invested in the technology." It seems that Hollywood production methods must actually be paid for.

Lawrence C. Paulson,
Cambridge, England

Author Responds:

I do not believe 50% of universities will fail but do expect a number will. The figures referred only to technology-enhanced courses and should not be interpreted as suggesting traditional teaching methods are inferior. Viewpoints present an author's opinions; in my case, the evidence behind mine was mainly from my personal experience teaching a variety of courses. Universities are tradition-bound, and a number of cultural factors make it difficult to bring about change, no matter how enlightened. Technology can enhance educational offerings and create a more active learning environment. But schools unable or unwilling to embrace technology-enhanced teaching and learning face an uncertain future.

Henry C. Lucas, College Park, MD, USA

Gender 'Equity' in Computer Science

As a scientist in Silicon Valley and longtime member of ACM, I am inspired by the growing representation

of women in science and engineering. (Incidentally, my only daughter has a degree in electrical engineering and is gainfully employed in artificial intelligence and robotics.) I support increased efforts to teach coding to girls (and boys) and eliminate gender bias (such as in grant and paper reviewing). I thus read Jodi L. Tims's "From the Chair of ACM-W" column "Achieving Gender Equity: ACM-W Can't Do It Alone" (Feb. 2018) with great interest, especially when she said, ". . . a nagging question that many of us who work so hard in the space of gender equity in computing have. Why, with so much sustained effort by so many individuals and organizations, is progress toward gender equity so slow?" My concern is that neither the column nor its cited works defined the "equity" mentioned in its headline. Is the only possible definition 50%/50% representation at every level of expertise? Or could it be, say, 56%/44% women/men—the percentages of all students in U.S. public colleges? Conversely, are the numerous professional disciplines where women outnumber or out-earn men manifestly "iniquitous" according to the column's assumed definition?

We STEM professionals and educators, and the public more generally, would gain clarity, and hence be better able to take enlightened action, if the goal were first made explicit and justified, then accepted by stakeholders.

David G. Stork, Portola Valley, CA, USA

Author Responds:

In working toward systemic change, we find it difficult to choose the appropriate gender-equity measure. Should we strive to be reflective of the college population, the workforce, or the overall population? Should the target vary by country or be culturally blind? Unfortunately, women account for only 26% of the technology workforce and less than 20% of undergraduate computing majors. Neither is close to being a useful definition of gender equity.

Jodi L. Tims, Berea, OH, USA

Get Serious About Social Responsibility

As a longtime activist in and former board member of Computer Professionals for Social Responsibility,¹ I was heartened by Moshe Y. Vardi's call in his "Vardi's Insights" column "Computer Professionals for Social Responsibility" (Jan. 2018) for greater focus on professional responsibilities in today's increasingly technology-driven era. I agree with Vardi that it should include new activity on the part of ACM, including critical introspection. I worry that while digital systems are being incorporated into every facet of modern life, many problems associated with these systems, including device addiction, surveillance, data harvesting, fake news, and worse, could be threatening, even as our collective ability to address them is diminishing.

While ACM could face significant structural barriers trying to address this need, including the potential for conflicts of interest, I hope it faces up to the challenge. At the very least, I suggest a basic goal should be to provide robust opportunities for open public discussion of the issues and related critiques concerning computers and society.

Ideally, computer scientists working with people from all elements of society would be able to help shape our future technology-using society, supporting the common good, not just the needs of corporations and governmental agencies. Uppermost would be to reduce inequality, improve educational opportunities for all, strengthen collective problem solving, and protect the natural environment, at least what is left of it. Sadly, it is not at all clear that the trajectory within the computer science community (or society in general) is today heading in this direction.

Reference

1. Computer Professionals for Social Responsibility; <http://cpsr.org/>

Douglas Schuler, Seattle, WA, USA

Why Voting in Secret Stays Secret

In his news item "Sharing Secrets (Without Giving Them Away)" (Jan. 2018), Arnout Jaspers aimed to de-

scribe a protocol created by Ronald Cramer, a cryptographer at the Dutch Centrum voor Wiskunde en Informatica research center, whereby three people are able to vote so the overall vote is known but individual votes are not disclosed. Cramer et al.¹ described the protocol, in essence, like this: Each voter selects two random numbers and computes the third one so the sum of all three is equal to the voter's own vote (1 if yes, 0 if no). To every voter (including him/herself) the voter then sends two of the three numbers, a different pair for each receiver. Each voter then adds all the numbers received from the other voters and from him/herself, and makes this sum public. The sum of all three public sums is thus twice the number of yes-votes.

Reference

1. Cramer, R., Damgard, I., and Nielsen, J. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, Cambridge, U.K., 2015.

Vladik Kreinovich, El Paso, TX, USA

Dismayed by Raised Fist on Cover

I am a lifetime subscriber to *Communications* and ACM and was dismayed by the cover of the Feb. 2018 issue. The raised, clenched fist is inextricably associated with leftist political movements. Please do not mix politics with hard science.

James Reynolds, Richardson, TX, USA

Not Based on Linux

In his news story "Going Serverless" (Feb. 2018), Neil Savage identified Kubernetes as "an open source container system based on Linux." Kubernetes is not based on Linux but rather is an open source container-management system based on Google's experience with Borg.—*The Editors*

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.

Speech Emotion Recognition: 20 Years in a Nutshell

The March into the Black Hole of Complexity

Internet Freedom in West Africa

Data Acquisition in VANET with Multihoning

More than Code: Learning Rules of Rejection in Writing Programs

Science, Policy, and Service: Some Thoughts on the Way Forward

Never-Ending Learning

Canary Analysis Service

Research for Practice: Cluster Scheduling for Datacenters

ACM's 2018 General Election: Meet the Candidates

Plus the latest news about biological computing, functional languages, and protecting medical data.

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



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

DOI:10.1145/3185514

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

Fostering Inclusion, Keeping the Net Neutral

ACM-W chair Jodi Tims offers ways everyone can promote inclusiveness, while Daniel A. Reed assesses the debate over Net neutrality.



Jodi Tims How Can We Foster Inclusiveness?

<http://bit.ly/2scVMMX>
January 3, 2018

This post is a follow-up to the article “Achieving Gender Equity: ACM-W Can’t Do It Alone,” which appeared in the February issue of *Communications*. If you have not yet read the article, doing so will provide relevant context.

The goal is to elicit thoughts on the question “What can an individual do on a day-to-day basis to ensure that her/his environment fosters inclusiveness?” When asking this question of ACM-W Council members, I received a number of suggestions, some of the “day-to-day” variety, and others that would require a bit more time and effort to enact.

Want to quickly achieve a better understanding of the issues faced by women in computing and contribute to more supportive environments for all computing professionals? You can:

- ▶ Once a month, reach out to a female colleague you do not know and ask about the work she does. Then, introduce her to someone she should

know (or who should know her) in your organization.

- ▶ Find an ACM article about equity and diversity, read it, and share it with peers, students, and others.

- ▶ Talk with peers and at staff meetings about issues of diversity such as unconscious bias and stereotype threat.

- ▶ Reach out to colleagues you trust and ask them to candidly assess if there are any gender or ethnic/minority biases in the current project.

- ▶ Ensure all members of a meeting, regardless of gender, have a chance to contribute to a discussion by explicitly inviting contributions from those who have been silent.

- ▶ Make sure original ideas are attributed to the person who generated them. It is frequently true that ideas offered up by women get remembered as coming from men.

- ▶ Seek out a person that most likely has a background or culture different from your own. Ask them how they made their career choice and what persuaded them to stick with a computing career. Use this input to encourage young women you meet to consider computer science as a future pathway.

- ▶ Invite a female colleague to give a presentation on her work at a weekly meeting or to a group of students.

- ▶ Once a month, become familiar with at least one woman (professor and/or student) on your campus and recognize the work they do and the accomplishments they have made to their chosen STEM profession. Introduce them to your students, peers, co-workers, friends, and others.

- ▶ Talk to people you meet from businesses/universities other than your own about issues of gender equity in their environments. Take good ideas back and share them with your colleagues.

Actions that may require more time or effort or may require the participation of others in your organization are:

- ▶ Team up with colleagues and adopt a local elementary, middle, or high school class. Visit three to five times a year and plan sessions and activities to sensitize/empower young men and women for inclusiveness.

- ▶ Use training resources to encourage young women to push back against negative peer pressure from both women and men that tries to dissuade them from sticking with computing.

- ▶ Mentor a female high school or college student interested in computing.

- ▶ Make sure that hiring, tenure, and promotion committees, as well as teaching faculty and managers, understand how unconscious bias can affect their decisions, and help those groups develop mechanisms to disrupt those biases.

- ▶ Nominate a female colleague for a promotion/award/recognition.

► Locate and attend a Women in Computing event. ACM Celebrations, ACM-W Student Chapters, and the Grace Hopper Celebration are options to consider.

Please contribute your ideas to this posting. ACM-W will feature the ideas generated on our Web page and in other publications. This will help us empower all computing professionals to do their part in transforming ACM into the premier example of a professional organization committed to gender equity.

Comments

One of the things I think is important is if you manage a group of students that do some level of community outreach (tech camps, school visits, and the like), it is important to ensure you have a diverse group of students. Not only will it help the target audience relate to those students and be more comfortable asking them questions, but it will also give your group different perspectives and ideas.

—Brian Krupp



Daniel A. Reed
The Shifting World
of Net Neutrality
<http://bit.ly/2IS4yJQ>
 December 11, 2017

N.B. While at Microsoft, I served on the U.S. Federal Communication Commission (FCC) Technical Advisory Committee, during a portion of Julius Genachowski's service as FCC chair (2009–2013). At that time, Tom Wheeler led the advisory committee, and he later succeeded Genachowski as chair of the FCC (2013–2017).

Utter the phrase “network neutrality” (<http://bit.ly/2E7WQmV>) and one is likely to engender two possible reactions. The first is a bewildered stare of incomprehension, something geeks experience frequently when using jargon-speak in inappropriate circumstances. (*Exhibit A: Holiday gatherings with extended family.*) The debate has also become major news, with extended coverage in such outlets as *The New York Times* and *The Wall Street Journal*, and it has begun to penetrate the popular consciousness.

The other response, from policy wonks, technical experts, and Internet/telecom service and content providers, is likely to be impassioned advocacy, with much gesticulation. They will either opine that we must ensure unfettered and equal Internet access by and

for all, or that we must ensure continued Internet innovation and free enterprise investment.

Both are clearly true. What, then, is the debate really about?

Although the early Internet grew from government research (see ARPANET and NSFNET, for example), today's Internet was largely built by the private sector, which rightly expects to profit from its investment. Simultaneously, the Internet is a crucial element of our society, supporting business and commerce, government services, and public communication; these are societal needs of great importance. Simply put, both the public and the private good matter, and they are sometimes in conflict.

Technically, network neutrality is about Internet traffic management and its possible prioritization. Can service providers give preference to some content based on defined criteria? Or is every packet the same and all content must be treated equally? The technical answer is obvious. Anyone who has operated networks or conducted network research knows that signaling and quality of service (QoS, <http://bit.ly/2E76wOm>) guarantees are essential elements of network management. The real issue is not technical network management, but about applying traffic shaping (<http://bit.ly/2nPLQZ4>) and other techniques to favor (or disfavor) certain entities based on business, market, or social advantage.

Thus, the network neutrality debate is largely a power and economic struggle between Internet service providers and those who deliver content and services. In an increasing number of cases, those two entities—service and content providers—are the same. Cellular operators and cable companies are two prime examples, providing broadband access while also offering content that competes with other content providers (such as Netflix).

The struggle is further convolved with a combination of social and political perspectives—pro-regulation or anti-regulation. Then there is the woefully obsolete nature of the governing law—the Communications Act of 1934 (<http://bit.ly/2FRbBL4>). There have been updates, most recently the Telecommunications Act of 1996 (<http://bit.ly/2GUTN2Y>), but 20-plus years is a geologic eon at Internet speed.

The legal and policy debate centers on whether the Internet should be con-

sidered a common carrier, like radio, television, and telephony, under Title II of the 1934 act, or as an information service under Title I of the 1934 act. The technical irony is that radio, television, and telephony are now all streamed over the Internet. That convolution is what makes application of the 1934 law so challenging. The Internet is a carrier but it is also an information service.

After much debate, via the Open Internet Order of 2015 (<http://bit.ly/2sccEmN>), the FCC, under Wheeler's leadership, chose to apply Title II, though forbearing several of the elements of Title II. I believe that was the right decision, allowing “light touch” regulation for equal access, but others disagree. On Dec. 14, 2017 (<http://bit.ly/2E5HjUw>), the FCC, under new chair Ajit Pai, reversed the 2015 ruling and shift questions about discriminatory rulings to the Federal Trade Commission (FTC).

Depending on one's perspective, the reversal of the 2015 order is either wonderful, allowing free enterprise to flourish without unnecessary and burdening government regulation, or disastrous, endangering fair access and innovation and allowing a small number of large companies to shape the future of a critical resource with little oversight. In practice, the full measure of either is unlikely to accrue, but there will be real effects. That is why there is so much heat surrounding the debate.

Regardless of one's business, legal, or social opinions, it is clear the network neutrality debate is yet another example of technical and business change rapidly outstripping outmoded laws, while powerful social and economic forces are at play. The nexus of digital privacy, transnational data flows, and the scope of extraterritorial legal reach is yet another. We badly need updated legal frameworks that reflect current realities and that are sufficiently flexible to accommodate rapidly evolving technologies. I wish I were more sanguine about that near-term probability. It is crucial that computer scientists become more involved as non-partisan experts.

Jodi Tims is chair of ACM-W, ACM's Council on Women in Computing. **Daniel A. Reed** is professor and university bioinformatics chair in the College of Liberal Arts & Sciences at the University of Iowa, Iowa City, IA, USA

© 2018 ACM 0001-0782/18/4 \$15.00

Always Out of Balance

Computational theorists prove there is no easy algorithm to find Nash equilibria, so game theory will have to look in new directions.

WHEN JOHN NASH WON the Nobel Prize in economics in 1994 for his contribution to game theory, it was for an elegant theorem. Nash had shown that in any situation where two or more people were competing, there would always be an equilibrium state in which no player could do better than he was already doing. That theorem has since been used to model all sorts of competitive systems, from markets to nuclear strategy to living creatures competing for finite resources.

“In some sense, it started not just game theory, but also modern economics,” says Christos Papadimitriou, a professor of computer science at Columbia University. Nash’s idea gave economists the ability to create hypotheses about market design, for instance. They could now ask what happened when a market reached equilibrium.

Nash’s theorem is also an essential component of game theory, which had first been developed by computing pioneer John von Neumann. “Games are a mathematical thought experiment and we study them just because we want to understand how strategic rational players would behave in situations of conflict,” Papadimitriou says. “And that’s important because all of society is full of such situations.”



Though Nash proved that at least one such Nash equilibrium existed for all games, what he did not do was predict how an equilibrium might be reached in a given situation. Was there, scientists wanted to know, an algorithm that would

show players how to efficiently reach an equilibrium? After more than 65 years of researchers’ studying that question, the answer turns out to be no, there is not. That means economists had better start rethinking some of their models.

An Intractable Problem

“This is something important and consequential,” says Papadimitriou. “It undermines the widespread use of Nash equilibrium as the natural condition, what will happen in a game.”

It is not that Nash equilibria do not exist; they do. In some types of games, players converge on an equilibrium very quickly; in others, however, the result is out of reach. It might be calculated, but it would take longer to arrive at than the whole lifetime of the universe. “If even the fastest computers in the world cannot compute equilibrium, how would you expect a bunch of people—the market, a group of people—to do it?” Papadimitriou asks.

Nash’s theorem says that in a non-cooperative game with a finite number of players, each of whom has a finite number of possible moves called “strategies,” there exists a way for players to randomly choose strategies until they reach a point where there is no alternate strategy a player can use to improve his results. “Everyone’s as happy as can be with what they’re doing, given that everyone else is doing what they’re doing,” says Tim Roughgarden, a professor of computer science at Stanford University.

Take the game rock, paper, scissors, for example. If player A is playing rock more often, player B can beat him by playing paper more often. When player A notices that, he will switch to playing scissors more often, which will in turn cause player B to switch to playing rock. If, however, each player randomly makes each of the three moves a third of the time, they will both start winning in equal proportions, and neither will want to change what he is doing because it will not improve his odds.

In zero-sum games, where one person loses what the other wins, there is a natural process by which play converges to equilibrium, Roughgarden says. Each player knows his own set of strategies, and he knows what the other player has done in the past. By randomly choosing strategies, with a bias toward strategies that have worked well in the past, the players quickly find an approximate equilibrium, where a player would have very little incentive to do something different. The number of moves the players have to make to get there is a logarithm of their number of strategies. “If you and I each have a million strategies,

Even by inferring the motivations of other players from their moves, there is no way to compute even an approximate Nash equilibrium in polynomial time.

we’re not going to have to play a million times to reach an approximate Nash equilibrium; we’re each going to have to play maybe 100 times,” he says.

The next logical question is what happens when the same algorithm is applied to a game that is not zero-sum? There, it turns out, the players can reach a weaker type of equilibrium. Perhaps, though, a different algorithm might work better. “It was not known whether or not there could be learning algorithms of this form which over just a sublinear number of steps could reach a Nash equilibrium,” says Roughgarden.

Papadimitriou, along with Costis Daskalakis of the Massachusetts Institute of Technology and Paul Goldberg of Oxford University, had shown in 2007 that for a class of games in which all players knew the payoff for each other, there is no polynomial-time algorithm for computing a Nash equilibrium under standard complexity assumptions. “If everybody knows the game, there is no way to compute a Nash equilibrium in the lifetime of the universe, if the world’s as complex as we think it is,” Papadimitriou says.

Aviad Rubinstein, a Ph.D. candidate under Papadimitriou, looked at what happened in situations where players do not know what benefits other players might get out of the game, making it more difficult to predict the other players’ strategies. Again, he found there was no way to compute the results in polynomial time, where the time it takes a computer to solve a problem grows as a polynomial function of the problem’s size.

Three Strikes

Rubinstein and Yakov Babichenko, a

professor of economics at Technion, the Israel Institute of Technology, showed that even by inferring the motivations of other players from their moves, there was no way to compute even an approximate Nash equilibrium in polynomial time. Players would have to see most of the possible moves to infer motivation. It might work, but it would take too long.

The three results are essentially three strikes against the utility of the Nash equilibrium, Papadimitriou says. “Together they tell us a very consistent story, that the Nash equilibrium has very serious credibility problems as the right concept for game theory.”

Rubinstein, who earned his Ph.D. last year and is now a post-doctoral fellow at Harvard University, says there has long been a discussion about how relevant Nash’s theorem is to economics, because no one knows how to reach an equilibrium. Further, while an algorithm—if one could be found—would be seeking equilibrium, that is not what real players are out to achieve. “Real selfish players, they’re not trying to find equilibrium; they’re trying to maximize their payoff,” Rubinstein says. “If there’s no specially designed algorithm that finds an equilibrium, it’s even less likely that selfish players will somehow magically converge to an equilibrium.”

There are still open questions. Nash equilibria are a subset of a broader category of correlated equilibria. In a correlated equilibrium, both players have access to some public information that informs their moves. Roughgarden uses a traffic light as an example; if one driver approaching an intersection sees the light in his direction is green, he can assume the driver coming down the cross-street will see a red light, so he can base his decision to keep going on the knowledge that the other driver is being signaled to stop.

Rubinstein says that, although he and Bobichenko proved that finding Nash equilibria is hard, no one knows if the same is true for correlated equilibria. The Nash equilibrium problem is an issue of communication complexity; to calculate the equilibrium, the players would have to communicate almost everything about their strategies, and as the game gets larger, that would take practically forever. Perhaps, though, it is easier for correlated equilibria.

Because the set of correlated equilibria is larger, it should be easier to find one. “If you want to find a needle in a haystack, it’s hard, but if you have lots of needles it might become easy,” says Karthik C.S., a Ph.D. student at Weizmann Institute of Science in Israel. He and fellow student Anat Ganor presented a paper last year that looked at the difficulty of finding an approximate correlated equilibrium. “What we show in our paper is that for small approximation values, it’s not any easier,” Karthik says.

Even though Bobichenko’s and Rubinstein’s result is in some sense negative—they prove there is no easy way to find Nash equilibria—the theorists do not see that as a necessarily bad thing. “It says what you can’t hope to do and it guides you to directions that are going to be fruitful,” Roughgarden says. “Without that theory, you might waste an enormous amount of time trying to find efficient algorithms when people don’t think they exist.” Now researchers can focus on finding special cases

where algorithms might work, as in zero-sum games, or look for ways to compromise on the results they can expect.

Economists will have to find some other basis on which to model markets, Rubinstein says. “It means you should be really careful about how you model selfish players,” he says. “For some problems, Nash equilibrium is just not the right model. We should find better models.”

Papadimitriou says he and other scientists are already looking for better ideas as to what economists should use in their models. He is delighted that this line of inquiry produced such solid results, and says computational theorists should be proud that it came from their field. “I started working on this in 1983, 35 years ago,” he says. “I thought this was one of these problems that we would never see solved.”

Further Reading

Babichenko, Y. and Rubinstein, A.
Communication complexity of approximate

Nash equilibria, arXiv, 2016.
<https://arxiv.org/abs/1608.06580>

Papadimitriou, C.H. and Roughgarden, T.
Computing Correlated Equilibria in Multi-Player Games, *Journal of the ACM*, July 2008
<https://dl.acm.org/citation.cfm?id=1379762>

Ganor, A. and C.S., Karthik
Communication Complexity of Correlated Equilibrium in Two-Player Games, *Electronic Colloquium on Computational Complexity*, 2017.
<https://arxiv.org/abs/1704.01104>

Myerson, R.B.
Nash Equilibrium and the History of Economic Theory, *Journal of Economic Theory*, 37, 1999.
<https://www.aeaweb.org/articles?id=10.1257/jel.37.3.1067>

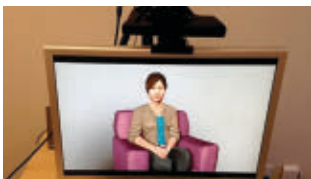
Rubinstein, A.
Communication complexity of approximate Nash equilibria
Institute for Advanced Study
<https://www.youtube.com/watch?v=hRV9chspWIO>

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

© 2018 ACM 0001-0782/18/4 \$15.00

ACM News

Users Open Up to Cartoon Clinician



Researchers have discovered that people are sometimes more willing to share their fears when they are able to open up to a virtual interviewer, rather than to a real person.

Researchers at the University of Southern California (USC) Institute for Creative Technologies (ICT) found soldiers suffering from post-traumatic stress disorder (PTSD) were more likely to reveal their problem to a on-screen cartoon character (as opposed to a live human being) as long as they could be sure that information remained anonymous.

“Compared to the gold-standard post-deployment health assessment (PDHA) used by the U.S. military, our virtual interviewer was able to increase reporting of post-traumatic stress symptoms among soldiers by over threefold,” says Gale M. Lucas, a senior research as-

sociate at ICT and lead researcher on the study.

While ICT’s researchers are loathe to refer to their cartoon clinician ‘Ellie’ as a virtual therapist, others working in the same area see the ICT avatar as an inevitable precursor to virtual therapy.

“If a car can drive itself, a therapy session is probably not far behind,” says Robert Schachter, a psychologist and press spokesperson for the Association of Cognitive and Behavioral Therapies (ACBT).

Ellie was able to build rapport with the 29 soldiers ‘she’ interviewed in the study by closely monitoring their smiles, frowns, gazes, and scores of other state-of-mind indicators as they engaged in conversation with her, according to ICT’s Lucas. Simultaneously, Ellie also responded empathetically to those cues by nodding where appropriate, asking for more detail about a particular anecdote, commiserating with a soldier over a sad story, and offering other seemingly ‘I-get-you’ responses.

Under the hood, Ellie was able to pull off the human-like behavior with ICT’s SimSensei software—ar-

tificial intelligence (AI) code that enables the avatar to recognize and respond to emotional cues for depression, anxiety, and PTSD.

In addition, the avatar tracked its interaction with soldiers in real time using face-tracking and head-tracking monitors, a COVAREP (Cooperative Voice Analysis Repository for Speech Technologies) speech analyzer, and an off-the-shelf desktop computer, monitor, webcam, and monitor, according to Stefan Scherer, a research assistant professor at ICT.

“I believe Ellie has a lot of potential,” says Matthew Pickard, an assistant professor at the University of New Mexico (UNM) doing similar research into what are referred to as ‘rapport agents.’ “There is an element of anonymity and lack of social and moral judgment that a virtual human brings that reduces the risks individuals perceive in opening up to them.”

Despite Ellie’s accomplishments with soldiers, ICT is adamant its cartoon clinician is not a virtual therapist, nor is ICT interested in creating a virtual therapist.

“We are not in the business of creating a ‘doc-in-a-box,’” says Albert “Skip” Rizzo, ICT’s director for medical virtual reality. “Rather, we aim to use rapport-building virtual human agents to engage patients with information that might help them to develop a better understanding of their situation and which may support them in making the decision to seek care with a live provider.”

UNM’s Pickard sees an Age of Virtual Therapy as a challenge, but not at all impossible. “The semantics of communication alone are diverse and pervasive; the tilt of a head, the vocalic emphasis on a word, the synchronicity of a phrase and a non-verbal gesture,” Pickard says.

“So, it takes a lot for a virtual human to ‘pass the Turing test’ and I think we are a long way from that milestone,” Pickard says. “But virtual therapists can still be useful with far fewer capabilities than a human possesses; that is why I think AI-driven virtual therapists are inevitable.”

—Joe Dysart is an Internet speaker and business consultant based in Manhattan, NY, USA.

Chips for Artificial Intelligence

Companies are racing to develop hardware that more directly empowers deep learning.

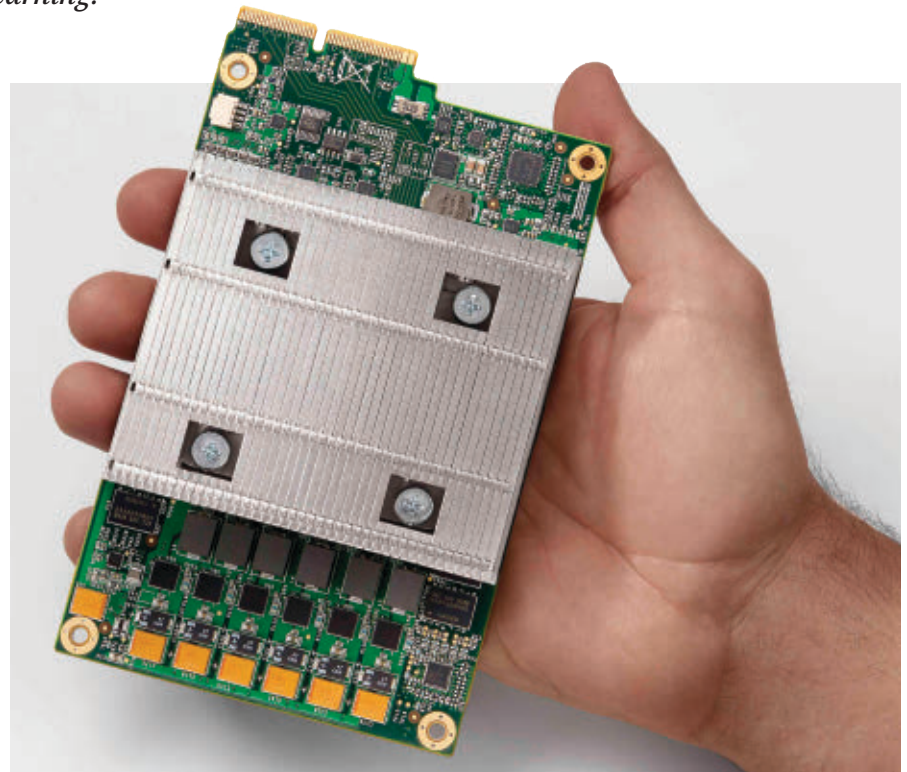
A LOOK UNDER the hood of any major search, commerce, or social-networking site today will reveal a profusion of “deep-learning” algorithms. Over the past decade, these powerful artificial intelligence (AI) tools have been increasingly and successfully applied to image analysis, speech recognition, translation, and many other tasks. Indeed, the computational and power requirements of these algorithms now constitute a major and still-growing fraction of datacenter demand.

Designers often offload much of the highly parallel calculations to commercial hardware, especially graphics-processing units (GPUs) originally developed for rapid image rendering. These chips are especially well-suited to the computationally intensive “training” phase, which tunes system parameters using many validated examples. The “inference” phase, in which deep learning is deployed to process novel inputs, requires greater memory access and fast response, but has also historically been implemented with GPUs.

In response to the rapidly growing demand, however, companies are racing to develop hardware that more directly empowers deep learning, most urgently for inference but also for training. Most efforts focus on “accelerators” that, like GPUs, rapidly perform their specialized tasks under the loose direction of a general-purpose processor, although complete dedicated systems are also being explored. Most of the companies contacted for this article did not respond or declined to discuss their plans in this rapidly evolving and competitive field.

Deep Neural Networks

Neural networks, in use since the 1980s, were inspired by a simplified model of the human brain. Deep learning



Google's tensor processing unit is designed for high throughput of low-precision arithmetic.

techniques take neural networks to much higher levels of complexity, their growing success enabled by enormous increases in computing power plus the availability of large databases of validated examples needed to train the systems in a particular domain.

The “neurons” in neural networks are simple computer processes that can be explicitly implemented in hardware, but are usually simulated digitally. Each neuron combines tens or hundreds of inputs, either from the outside world or the activity of other neurons, assigning higher weights to some than to others. The output activity of the neuron is computed based on a nonlinear function of how this weighted combination compares to a chosen threshold.

“Deep” neural networks arrange the neurons into layers (as many as tens of layers) that “infer” successively

more abstract representations of the input data, ultimately leading to its result; for example, a translated text, or recognition of whether an image contains a pedestrian.

The number of layers, the specific interconnections within and between layers, the precise values of the weights, and the threshold behavior combine to give the response of the entire network to an input. As many as tens of millions of weights are required to specify the extensive interconnections between neurons. These parameters are determined during an exhaustive “training” process in which a model network is given huge numbers of examples with a known “correct” output.

When the networks are ultimately used for inference, the weights are generally kept fixed as the system is exposed to new inputs. Each of the many neurons in a layer performs an



Advertise with ACM!

Reach the innovators and thought leaders working at the cutting edge of computing and information technology through ACM's magazines, websites and newsletters.



Request a media kit with specifications and pricing:

Ilia Rodriguez
+1 212-626-0686
acmm mediasales@acm.org



independent calculation (multiplying each of its inputs by an associated weight, adding the products, and doing a nonlinear computation to determine the output). Much of this computation can be framed as a matrix multiplication, which allows many steps to be done in parallel, said Christopher Fletcher, a computer scientist at the University of Illinois at Urbana-Champaign, and “looks like problems that we’ve been solving on GPUs and in high-performance computing for a very long time.”

Customizing Hardware

During inference, unlike in offline training, rapid response is critical, whether in self-driving cars or in web applications. “Latency is the most important thing for cloud providers,” Fletcher noted. In contrast, he said, traditional “GPUs are designed from the ground up for people who don’t care about latency, but have so much work that as long as they get full throughput everything will turn out OK.”

Recognizing the importance of response time and anticipating increasing power demands by neural-network applications, cloud behemoth Google developed its own application-specific integrated circuit (ASIC) called a “tensor-processing unit,” or TPU, for inference. Google reported in 2017 that, in its data-centers, the TPU ran common neural networks 15 to 30 times faster than a contemporary CPU or GPU, and used 30 to 80 times less power for the same computational performance (operations per second). To guarantee low latency, the designers streamlined the hardware and omitted common features that keep modern processors busy, but also demand more power. The critical matrix-multiplication unit uses a “systolic” design in which data flows between operations without being returned to memory.

So far, Google seems to be unusual among Web giants in designing its own chip, rather than adapting commercially available alternatives. Microsoft, for example, has been using field-programmable gate arrays (FPGAs), which can be rewired after deployment to perform specific circuit functions. Facebook is collaborating with Intel to evaluate its ASIC, called the Neural Network Processor. That chip, aimed at artificial-intelligence applications,

started life in Nervana, a startup that Intel acquired in 2016. Unsurprisingly, Nvidia, already the dominant vendor of GPUs, has released updated designs that it says will better support neural network applications, in both inference and training.

These chips follow a strategy that is familiar from other specialized applications, like gaming. Farming out the heavy calculations to a specialized accelerator chip sharing a bus with a general processor and memory allows rapid implementation of new ideas, and lets chip designers focus on dedicated circuits assuming all needed data will be at hand. However, the memory burdens posed by this “simplest” approach is likely to lead to systems with tighter integration, Fletcher said, such as bringing accelerator functions on-chip with the processor. “I think we will inevitably see the world move in that direction.”

Neuromorphic Hardware

One technique exploited by the new chips is using low-precision, often fixed-point data, eight bits or even fewer, especially for inference. “Precision is the wild, wild west of deep learning research right now,” said Illinois’s Fletcher. “One of the major open questions in all of this as far as hardware accelerators are concerned is how far can you actually push this down without losing classification accuracy?”

Results from Google, Intel, and others show that such low-precision computations can be very powerful when the data is prepared correctly, which also opens opportunities for novel electronics. Indeed, neural networks were inspired by biological brains, and researchers in the 1980s implemented

Google developed its own application-specific integrated circuit, the tensor processing unit (TPU), for inference.

them with specialized hardware that mimicked features of brain architecture. Even within the last decade, large government-funded programs in both the U.S. and Europe pursued “neuromorphic” chips that operate on biology-inspired principles to improve performance and increase energy efficiency. Some of these projects, for example, directly hard-wire many inputs to a single electronic neuron, while others communicate using short, asynchronous voltage spikes like biological neurons. Despite this history, however, the new AI chips all use traditional digital circuitry.

Qualcomm, for example, which sells many chips for cellphones, explored spiking networks under the U.S. Defense Advanced Research Projects Agency (DARPA) program SyNAPSE, along with startup Brain Corporation (in which Qualcomm has a financial stake). But Jeff Gehlhaar, Qualcomm’s vice president for technology, said by email that those networks “had some limitations, which prevented us from bringing them to commercial status.” For now, Qualcomm’s Artificial Intelligence Platform aims to help designers exploit digital circuits for these applications. Still, Gehlhaar noted the results are being studied by others as “this field is getting a second look.”

Indeed, although its NNP chip does not use the technology, Intel also announced a test chip called Loihi that uses spiking circuitry. IBM exploited its SyNAPSE work to develop powerful neuromorphic chip technology it called TrueNorth, and demonstrated its power in image recognition and other tasks.

Gill Pratt, a leader for SyNAPSE at DARPA and now at Toyota, said even though truly neuromorphic circuitry has not been adopted commercially yet, some of the lessons from that project are being leveraged in current designs. “Traditional digital does not mean lack of neuromorphic ideas,” he stressed. In particular, “sparse computation” achieves dramatically higher energy efficiency by leaving large sections of the chip underused.

“Any system that is very power efficient will tend to be very sparse,” Pratt said, the best example being the phenomenal computational power that our brains achieve with less than 20 watts of power.

Although power is critical to data-

During the last decade, government-funded programs in the U.S. and Europe have pursued the development of neuromorphic chips.

centers and especially for handheld devices, Pratt noted that even cars can face serious power challenges. Prototype advanced safety and self-driving features require thousands of watts, but would need much more to approach human capabilities, and Pratt thinks hardware will eventually need to exploit more neuromorphic principles. “I am extremely optimistic that is going to happen,” he said. “It hasn’t happened yet, because there have been a lot of performance improvements, both in terms of efficiency and raw compute horsepower, to be mined with traditional methods, but we are going to run out.” **C**

Further Reading

Joupi, N.P., et al

In-Datcenter Performance Analysis of a Tensor Processing Unit
44th International Symposium on Computer Architecture (ISCA), Toronto, Canada, June 26, 2017
<https://arxiv.org/ftp/arxiv/papers/1704/1704.04760.pdf>

Monroe, D.

Neuromorphic Computing Gets Ready for the (Really) Big Time, *Communications*, April 2014, pp. 13-15
<https://cacm.acm.org/magazines/2014/6/175183-neuromorphic-computing-gets-ready-for-the-really-big-time/fulltext>

U.S. Defense Advanced Research Projects Agency DARPA SyNAPSE Program
<http://www.artificialbrains.com/darpa-synapse-program>

Don Monroe is a science and technology writer based in Boston, MA, USA.

© 2018 ACM 0001-0782/18/4 \$15.00

ACM Member News

USING SPINTRONICS AFTER MOORE’S LAW



“The work I do is at the interface of computer science and electrical engineering,”

says Sachin Sapatnekar, a professor in the Department of Electrical and Computer Engineering at the University of Minnesota, where he holds the Robert and Marjorie Henle Chair, and the Distinguished McKnight University Professorship. His research interests lie in developing efficient techniques for the computer-aided design of integrated circuits.

Sapatnekar received his undergraduate degree in electrical engineering from the Indian Institute of Technology, Bombay; his master’s degree in computer engineering from Syracuse University, and his Ph.D. in electrical engineering from the University of Illinois at Urbana-Champaign.

In recent years, as Moore’s Law has matured, there has been concern about circuits growing old and degrading over time. Sapatnekar spent time looking at the reliability of integrated systems, and developed algorithms that allow the design of chips that operate reliably, even as they degrade with age.

His current interest is exploring what happens generally after Moore’s Law ends, and specifically what will happen to combined metal oxide semiconductors (CMOS). “There are some interesting directions there, with new architectures coming up,” he asserts.

Sapatnekar says he has been using spintronics technology (an emerging field that utilizes electron spin to improve efficiencies and create new functionalities in electronic devices) to look at building logic and computer memory structures. “It is really exciting, because as Moore’s Law ends, it is creating a rejuvenation in the way people think about design, and that brings a lot of new technical problems.”

—John Delaney

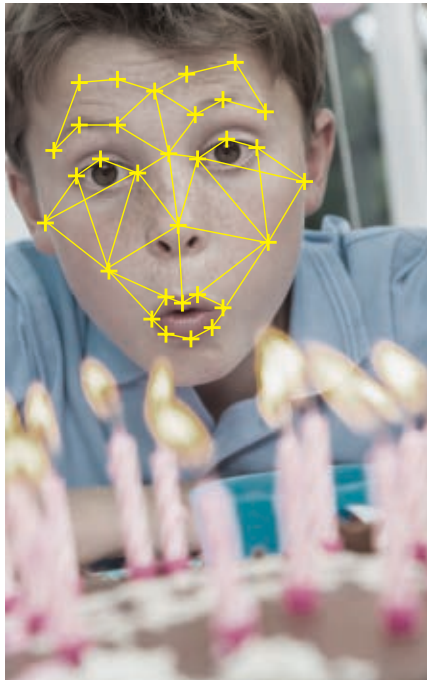
Artificial (Emotional) Intelligence

Enabled by advances in computing power and neural networks, machines are getting better at recognizing and dealing with human emotions.

ANYONE WHO HAS BEEN frustrated asking questions of Siri or Alexa—and then annoyed at the digital assistant’s tone-deaf responses—knows how dumb these supposedly intelligent assistants are, at least when it comes to emotional intelligence. “Even your dog knows when you’re getting frustrated with it,” says Rosalind Picard, director of Affective Computing Research at the Massachusetts Institute of Technology (MIT) Media Lab. “Siri doesn’t yet have the intelligence of a dog,” she says.

Yet developing that kind of intelligence—in particular, the ability to recognize human emotions and then respond appropriately—is essential to the true success of digital assistants and the many other artificial intelligences (AIs) we interact with every day. Whether we’re giving voice commands to a GPS navigator, trying to get help from an automated phone support line, or working with a robot or chatbot, we need them to really understand us if we’re to take these AIs seriously. “People won’t see an AI as smart unless it can interact with them with some emotional *savoir faire*,” says Picard, a pioneer in the field of affective computing.

One of the biggest obstacles has been the need for context: the fact that emotions can’t be understood in isolation. “It’s like in speech,” says Pedro Domingos, a professor of computer science and engineering at the University of Washington and author of *The Master Algorithm*, a popular book about machine learning. “It’s very hard to recognize speech from just the sounds, because they’re too ambiguous,” he points out. Without context, “ice cream” and “I scream” sound



identical, “but from the context you can figure it out.”

The same is true of emotional expression. “If I zoom in on a Facebook photo and you see a little boy’s eyes and mouth,” says Picard, “you might say he looks surprised. And then if we zoom out a little bit you might say, ‘Oh, he’s blowing out a candle on his cake—he’s probably happy and excited to eat his cake.’” Getting the necessary context requires the massive amounts of data that computers have only recently been able to process. Aiding that processing, of course, are today’s powerful deep-learning algorithms.

These advances have led to major breakthroughs in emotion detection just in the last couple of years, such as learning from the raw signal, says Björn Schuller, editor-in-chief of *IEEE Transactions on Affective Computing* and head of Imperial College London’s Group on Language, Audio & Music. Such “end-

to-end” learning, which Schuller himself has helped develop, means a neural network can use just the raw material (such as audio or a social media feed) and the labels representing different emotions to “learn all by itself to recognize the emotion inside,” with minimal labeling by humans.

The latest algorithms are also enabling what scientists call “multimodal processing,” or the integration of signals from multiple channels (“modalities”), such as facial expressions, body language, tone of voice, and physiological signals like heart rate and galvanic skin response. “That’s very important because the congruence of those channels is very telling,” says Maja Matarić, a professor of computer science, neuroscience, and pediatrics at the University of Southern California. Given people’s tendency to mask their emotions, information from only a single channel, such as the face, can mislead; a more accurate picture emerges by piecing together multiple modalities. “The ability to put together all the pieces is getting a lot more powerful than it ever has been,” adds Picard, whose research group has used the multimodal approach to not only discern a person’s current mood, but even to predict their mood the next day, with the goal of using the information to improve future moods.

Along similar lines, several researchers have engineered ways to mine multiple streams of data to detect severity of depression, thus potentially predicting the suicide risk of callers on a mental-health helpline.

Although recognizing emotions might seem like a uniquely human strength, experts point out that emotions can be distilled to sets of signals that can be measured like any other phenomenon. In fact, the power of

a multimodal approach for emotion recognition suggests that computers actually have an edge over humans. In a recent study, Yale University social psychologist Michael Kraus found that when people try to guess another's emotion, they're more accurate when they only hear the person's voice than when they're able to use all their senses, which tend to distract. In other words, for humans, less is more. Kraus attributes this effect to our limited bandwidth, a processing constraint computers are increasingly shedding. "I think computers will surpass humans in emotion detection because of the bandwidth advantage," says Kraus, an assistant professor of organizational behavior.

This computational advantage also shows itself in recent advances in recognizing "microexpressions," involuntary facial expressions so fleeting they are difficult for anyone but trained professionals to spot. "Modeling facial expressions is nothing new," says James Z. Wang, a professor of Information Sciences and Technology at Pennsylvania State University. But microexpressions—which might reveal a telltale glimpse of sadness behind a lingering smile, for example—are trickier. "These are so subtle and so fast"—flashing for less than a fifth of a second—that they require the use of high-speed video recordings to capture and new techniques to model computationally. "We take the differences from one frame to another, and in the end we are able to classify and identify these very subtle, very spontaneous expressions," says Wang.

What happens after a computer correctly recognizes an emotion? Responding appropriately is a separate challenge, and the progress in this area has been less revolutionary in recent years, according to Schuller. Most dialogue systems, for example, still follow hand-crafted rules, he says. A phone help line might be just smart enough to transfer you to a human operator if it senses you are angry, but still lacks the sophistication to calm you down by itself.

On the other hand, some researchers are already designing robots that can respond intelligently enough to influence human behavior in positive ways. Consider the "socially assistive robots" that Mataric designs to help

autistic children learn to recognize and express emotions. "All robots are autistic to a degree, as are children who are autistic," Mataric points out, "and that's something to leverage."

Autistic children find robots easier to interact with than humans, yet more engaging than a disembodied computer, making them ideal learning companions. If a kid-friendly robot seeing a child act in a socially appropriate way during a training session makes the child happy by blowing bubbles, for example, the child will be more motivated to improve. But while the robots must recognize and reward appropriate behavior, "the idea is not to just reward," Mataric explains, "but rather to serve as a peer in an interaction that gives children with autism the opportunity to learn and practice social skills."

Mataric and her colleagues are using similar approaches in designing robots that work with stroke patients and with obese teens, with robots understanding how much they can push each user to exercise more.

These robots are not meant to replace human caregivers, Mataric says, but to complement them. That's also the goal of Jesse Hoey, an associate professor of health informatics and AI at the University of Waterloo, who is developing an emotionally aware system to guide Alzheimer's disease patients through hand-washing and other common household tasks.

"At first glance it seems like a straightforward problem," Hoey says: just use sensors to track where the patients are in the task and use a recorded voice to prompt them with the next step when they forget what they've done. The mechanics of the system work just fine—but too often, people with Alzheimer's ignore the voice prompt. "They don't listen to the prompt, they don't like it, they react negatively to it, and the reason they react in all these different ways, we started to understand, was largely to do with their emotional state at a fairly deep level; their sense of themselves and who they are and how they like to be treated."

Hoey's starkest example is of a World War II veteran who grew very distressed because he thought the voice was a call to arms; for this user, a female voice might have been more effective. Another patient, who had once

been a lawyer, shifts his self-image from day to day, sometimes doling out legal advice and other days acting more in line with his current identity. "The human caregivers are good at picking this up," Hoey says, and that sensitivity enables them to treat the patient with the appropriate level of deference. So the challenge is to create a computer system that picks up on signals of power (such as body posture and speech volume), as well as signals of other aspects of emotion.

Applying artificial emotional intelligence to help those struggling with Alzheimer's, autism, and the like certainly seems noble—but not all applications of such technology are as admirable. "What every company wants to do, and won't necessarily admit it, is to know what your emotional state is second by second as you're using their products," says Domingos. "The state of the art of manipulating people's emotions is less advanced than detecting them, but the point at the end of the day is to manipulate them. The manipulations could be good or bad, and we as consumers need to be aware of these things in self-defense." **C**

Editor's Note: For more information on speech emotion recognition, look for Björn Schuller's article in the May 2018 issue.

Further Reading

Jaques, N., Taylor, S., Nosakhare, E., Sano, A., and Picard, R.

Multi-task Learning for Predicting Health, Stress, and Happiness, NIPS Workshop on Machine Learning for Healthcare, December 2016, Barcelona, Spain <http://affect.media.mit.edu/pdfs/16.Jaques-Taylor-et-al-PredictingHealthStressHappiness.pdf>

Tzirakis, P., Trigeorgis, G., Nicolaou, M.A., Schuller, B., and Zafeiriou, S. End-to-End Multimodal Emotion Recognition using Deep Neural Networks, *Journal of LaTeX Class Files*, Vol. 14 No.8, August 2015 <https://arxiv.org/abs/1704.08619>

Xu, F., Zhang, J., and Wang, J. Z. Microexpression Identification and Categorization using a Facial Dynamics Map, *IEEE Transactions on Affective Computing*, vol. 8, no. 2, pp. 254-267, 2017. <http://infolab.stanford.edu/~wangz/project/imsearch/Aesthetics/TAC16/>

Based in San Francisco, CA, USA, Marina Krakovsky is the author of *The Middleman Economy: How Brokers, Agents, Dealers, and Everyday Matchmakers Create Value and Profit* (Palgrave Macmillan).

© 2018 ACM 0001-0782/18/4 \$15.00

Technology Strategy and Management

Business Ecosystems: How Do They Matter for Innovation?

Considering the significant interrelationship of innovation, corporate strategy, and public policy for business ecosystems.

MORE AND MORE people are living and working in business ecosystems. We read and talk about the entrepreneurial ecosystem, the e-commerce ecosystem, and the mobility ecosystem. But we do not know enough about the key characteristics of an ecosystem that make it innovative. For some, any cluster that involves multiple types of actors—entrepreneurs, investors, intermediaries such as incubators and accelerators—constitutes an ecosystem. For others, it is a biological metaphor that, when applied to manmade systems, is only partially useful.

Precise understanding of a business ecosystem would help startup entrepreneurs and incumbent businesses compete and collaborate more effectively. This column clarifies the concept in order to identify good use and misuse of the term “ecosystem.” It also elaborates its utility when busi-

nesses formulate their strategy, and when policymakers wish to promote and regulate business ecosystems.

Ecosystems: What Is Different from Clusters?

The second half of the 20th century saw the rise of global value chains in manufacturing covering dispersed production locations. Clusters—networks of firms co-located in a specific region—developed in some of these locations, each specializing in a product. For example, the industrial districts in Emilia Romagna, Italy, discovered vibrant export markets for their textiles, footwear, machinery, and machine tools. Baden-Württemberg, Germany, has an automotive cluster, with luxury brands such as Mercedes-Benz and Porsche located in close proximity to component suppliers and advanced automotive research institutes. Clusters are more flexible and agile than vertically integrated firms because they can take

advantage of both economies of scale and scope.

Why do we need the notion of “ecosystems,” when “clusters” or their close cousin “networks” capture much of the phenomenon? Why are people finding it more useful to talk in terms of “ecosystems” rather than “clusters” today? Is this just a fad, or is there something substantive worthy of attention? James Moore in his 1993 *Harvard Business Review* article first popularized the idea of a “business ecosystem.”² But that was 25 years ago, and we need to re-evaluate its relevance in the light of technological developments since then.

To define a business ecosystem, we focus on a value-creating activity, such as entrepreneurship or innovation, rather than an industrial sector. A business ecosystem therefore tends to cover a variety of industries. There are three meta-characteristics of business ecosystems which, taken together, distinguish

an ecosystem from a cluster. Here, I establish what they are and thus the essence of what is going on in an ecosystem (see the accompanying table).

The first characteristic is *sustainability*. A biological ecosystem is defined as a system that includes all living organisms (biotic factors) in an area and its physical environment (abiotic factors) functioning together as a unit. Just as we identify living things like animals and plants as well as non-living things such as rocks and soil in a biological ecosystem, a business ecosystem consists of humans (for example, entrepreneurs) and environmental structures (such as incubators). Just as there are food chains in which resources are used and recycled in a biological ecosystem, there is a hierarchy of human actors who use and reuse resources in a sustainable manner in a business ecosystem.

Sustainability implies that the ecosystem can thrive without outside influence or assistance. That is, the ecosystem can meet the needs of the present without compromising the ability to satisfy the needs for the future. Sustainability is of course an important theme for public policymakers in cities and regions, and for major businesses for their own survival today.

The second characteristic is *self-governance*. This implies the ecosystem is not dependent on an outside force, nor is it controlled by a single dominant actor within the ecosystem. There is therefore no unilateral top-down hierarchical control. It also implies that although some activities are governed by a shared set of formal rules and informal norms, the ecosystem allows for the emergence of competing rules or standards that challenge established ones. The self-governing structure is attractive to many who work inside the business ecosystem. On the basis of this definition, it is erroneous to talk about the Wal-Mart ecosystem or the Toyota ecosystem.

The third essential characteristic of business ecosystems is *evolution*, that is, their ability to evolve over time through competition and experimentation. A biological analogy is the survival of the fittest, which involves combining competition and collaboration between species. Experimenta-



tion may be via R&D leading to invention, but also over business models leading to business model innovation.³ In the process of evolution, some species adapt and survive, while others cannot adapt and therefore become extinct. Over a long period of time, some ecosystems thrive, while other ecosystems stagnate or die. These dynamics apply just as much to business ecosystems as to biological ecosystems.

In short, a business ecosystem is a collection of business and other actors with resources operating as an interdependent system. Business ecosystems differ from clusters in sustainability, self-governance, and capacity to evolve over time.

Combining the Digital World with the Physical World

Digital technologies and infrastructure create significant opportunities for

business ecosystems. In fact, the ecosystem perspective is particularly useful when applied to digitally mediated ecosystems, but subtle differences persist.

Consider the ecosystems built by platform leaders such as Apple or Google. They provide a technology platform for the purpose of creating a community of supporting producers, notably application software developers. While such an ecosystem may be sustainable over time, complementary producers' hands are quite tied as they share resources and standards on terms set by the platform leader. In this sense, this sort of ecosystem is thin on the meta-characteristics of self-governance and evolution, as its ability to adapt to changing environments is limited by the dominance of a single platform leader.

Other types of ecosystems are more complete in their meta-characteristics. For example, startup ecosystems, be

they in Silicon Valley or Bangalore, are communities of stakeholders with resources organized around the process of entrepreneurial opportunity discovery, pursuit, and scale-up. As such, this type of ecosystem is defined as a system of value-creating activity that can be sustainable, self-governing, and evolutionary. To the extent that people and resources are imperfectly mobile, startup ecosystems continue to be embedded in a specific geography, even with the spread of crowdsourcing and virtual communication.

Another notable example is the mobility ecosystem with connected cars, ride sharing, and driverless transportation. This ecosystem is in the making in so-called “smart cities” due to huge potential in exploiting digital technologies. At present, all sorts of actors are vying to become the aggregator of “mobility as service,” with a view to reaching a shared goal to make a step change in improving mobility, reducing congestion and pollution, and raising the quality of life of citizens. The mobility ecosystem is geographically anchored, but cuts across industries with competition and collaboration between incumbents (such as automakers and component suppliers) and new entrants (such as technology startups but also incumbents in electronics and telecommunication). Collaboration is also necessary between the private sector (with the likes of Uber and Lyft championing the “sharing economy”) and the public sector with its city planners and politicians.

The mobility ecosystem has the promise of being sustainable (because the actors’ shared goal is to reduce congestion and pollution), self-governing (though for now with different ideas about the nature of regulatory oversight), and evolutionary (with different smart cities likely to find different solutions to the same problem over time). In this way, the mobility ecosystem helps us focus our attention on the essence of business ecosystems in general.

Ecosystems in the 21st Century

Business ecosystems may be an elusive metaphor for many readers. In this column, I have suggested the term is applicable only when we see elements of sustainability, self-governance, and

Different types of business ecosystems.

	Platform-based ecosystem	Startup ecosystem	Mobility ecosystem
Sustainable (in resource use today and the future)	×	×	×
Self-governing (with some competing rules)		×	×
Evolutionary (via competition and experimentation)		×	×

evolution. The ecosystem perspective will remain particularly useful for the 21st century. I conclude by considering the implications for innovation, corporate strategy, and public policy.

Being part of a business ecosystem implies that actors are associated with opportunities for value creation and risks of value destruction. In other words, business (and non-business) actors are interested in value “co-creation,” but to state the obvious, some win and others lose. Although there has been much recent discussion regarding competition between digital platform ecosystems, we must focus our attention more on within-ecosystem competition between different types of actors. The mobility ecosystem is a good case in point. To win, being a first mover with a novel idea or new technology always helps, but that is not the only way, as I explain below.

Balancing collaboration and competition with a broad range of ecosystem actors is central to engage in cross-cutting innovation. As a business entity, your suppliers are “partners” as well as potential competitors. For instance, in a mobility ecosystem, automakers must treat electronic software providers or public regulators as partners to seek a joint innovative solution for a specific smart city. In this world of partnering, business activities are likely to be evaluated for simultaneously creating value for the business and satisfying social goals. Moreover, business success or failure will depend in part on cultivating skills to facilitate complex coordination of expertise, and to access assets that you do not own. These capabilities create distinctive advantages that some players may leverage to win out in a business ecosystem.

Last and not least, business ecosystems require good governance for sustainability and evolution. This is challenging at the best of times with no easy solution, and should not be a concern just for

public policymakers. Business ecosystems pose an extra challenge for regulators because innovation by ecosystem actors tends to use novel technologies and business models for which adequate regulation does not yet exist—think of ride-sharing services or cryptocurrency. Moreover, ecosystems blur boundaries of industries for which traditional regulation has been crafted—think of mobile money that falls between financial regulation and telecom regulation.

Ensuring good governance of business ecosystems may rely on multiple mechanisms. First, new levels of transparency of buyer-supplier rating systems would help maintain good service quality and keep corporate conduct in line. Second, governments may work more collaboratively with private-sector actors who know more about the new technologies and markets to be regulated, though one must beware of “regulatory capture” by powerful interests. Third, governments may endorse outcome-based regulation by becoming “super-regulators.”¹ Super-regulators approve non-government (profit and not-for-profit) organizations as regulators. Private regulators are then authorized to set their own standards and rules as long as they meet certain outcomes, such as data security or privacy. In today’s geopolitical world, digital technology enables, but also potentially threatens, the viability of business ecosystems. Thus, business ecosystems would be in trouble if not governed well. □

References

- Hadfield, G. *Rules for a Flat World*. Oxford University Press, New York, 2016.
- Moore, J.F. Predators and prey: A new ecology of competition. *Harvard Business Review*, (1993), 75–86.
- Sako, M. Business models in strategy and innovation. *Commun. ACM* 55, 7 (July 2012), 22–24.

Mari Sako (mari.sako@sbs.ox.ac.uk) is Professor of Management Studies at Saïd Business School, University of Oxford, U.K.

Copyright held by author.



Kode Vicious

Popping Kernels

Choosing between programming in the kernel or in user space.

Dear KV,

I have been working at the same company for more than a decade, and we build what you can think of as an appliance—basically a powerful server meant to do a single job, instead of operating as a general-purpose system. When we first started building this system, nearly all the functionality we implemented was added to the operating system kernel as extensions and kernel modules. We were a small team and capable C programmers, and we felt that structuring the system this way gave us more control over the system generally, as well as significant performance gains since we did not have to copy memory between the kernel and user space to get work done.

As the system expanded and more developers joined the project, management started to ask questions about why we were building software in such a difficult-to-program environment and with an antiquated language. The HR department complained it could not find sufficient, qualified engineers to meet the demands of management for more hands to make more features. Eventually, the decision was made to move a lot of functions out of the kernel and into user space. This resulted in a split system, where nearly everything had to go through the kernel to get to any other part of the system, which resulted in lower performance as well as a large number of systemic errors. I have to admit that those errors, if they occurred in



the kernel, would have caused the system to panic and reboot, but even in user space, they caused functions to restart, losing state and causing service interruptions.

For our next product, management wants to move nearly all the functions into user space, believing that by having a safer programming environment, the team can create more features more quickly and with fewer errors. You have written about kernel programming from time to time: Do you also think the kernel is not for “mere

mortals” and that most programmers should stick to working in the safer environment of user space?

Safety First

Dear Safety,

The wheel of karma goes around and around and spares no one, including programmers, kernel, user space, or otherwise.

Programming in user space is safer for a very small number of reasons, not the least of which is the virtual

memory system, which tricks programs into believing they have full control over system memory and catches a small number of common C-language programming errors, such as touching a piece of memory that the program has no right to touch. Other reasons include the tried-and-true programming APIs that operating systems have now provided to programs for the past 30 years. All of which means programmers can possibly catch more errors before their code ships, which is great news—old news, but great news. What building code in user space does not do is solve the age-old problems of isolation, composition, and efficiency.

If what you are trying to build is a single program that takes some input and transforms it into another form, think of common tools such as `sed`, `diff`, and `awk`, and then, yes, those programs are perfectly suited to user space. What you describe is a system that likely has more interactions with the outside world than it has with a typical end user.

Once we move into the world of high-throughput and/or low-latency systems for the processing of data, such as a router, high-end storage device, or even some of the current crop of devices in the Internet of Things (see my October 2017 column IoT: The Internet of Terror; 10.1145/3132728), then your system has a completely different set of constraints, and most programmers are not taught how to write code for this environment; instead, they learn it through very painful experience. Of course, trying to explain that to HR, or management, is a lot like beating your head on your desk—it only feels good when you stop.

You say you have been at this for a while, so surely you have already seen that things that are difficult to do correctly in the kernel are nearly as difficult to get right in user space and rarely perform as well. If your problem must be decomposed into a set of cooperating processes, then programming in user space is the exact same problem as programming in the kernel, only with more overhead to pay for whatever hybrid form of interprocess communication you use. My per-

The tension in any of these systems is between performance and isolation.

sonal favorite form of this stupidity is when programmers build systems in user space, using shared memory, and then reproduce every possible contortion of the locking problem seen in kernel programming. Coordination is coordination, whether you do it in the kernel, in user space, or with pigeons passing messages—though the first two places have fewer droppings to clean up.

The tension in any of these systems is between performance and isolation. Virtual memory—which gives us the user/kernel space split and the process model of programming whereby programs are protected from each other—is just the most pervasive form of isolation. If programmers were really trusting, then they would have all their code blended into a single executable where every piece of code could touch every piece of memory, but we know how that goes. It goes terribly. What is to be done?

Over the past few years, there have been a few technological innovations that might help with this problem, including new systems programming languages such as Rust and Go, which have more built-in safety, but they have yet to prove their worth in a systems environment such as an operating system. No one is replacing a Unix-like operating system with something written in Go or Rust just yet. Novel computer architectures such as the work on Capabilities carried out in the CHERI project, developed at SRI International and the University of Cambridge, might also make it possible to decompose software for safety and retain a high level of performance in the overall system, but again, that has yet to be proven in a real deployment of the technology.

For the moment, we are stuck with the false security of user space, where we consider it a blessing that the whole system does not reboot when a program crashes, and we know how difficult it is to program in the wide open, single address space of an operating system kernel.

In a world in which high-performance code continues to be written in a fancy assembler, a.k.a. C, with no memory safety and plenty of other risks, the only recourse is to stick to software engineering basics. Reduce the amount of code in harm's way (also known as the attack surface), keep coupling between subsystems efficient and explicit, and work to provide better tools for the job, such as static code checkers and large suites of runtime tests.

Or, you know, just take all that carefully crafted kernel code, chuck it into user space, and hope for the best. Because, as we all know, hope is definitely a programming best practice.

KV

Related articles on queue.acm.org

A Nice Piece of Code

George V. Neville-Neil

Colorful metaphors and properly reusing functions

<https://queue.acm.org/detail.cfm?id=2246038>

The Cost of Virtualization

Ulrich Drepper

Software developers need to be aware of the compromises they face when using virtualization technology.

<https://queue.acm.org/detail.cfm?id=1348591>

Unikernels: Rise of the Virtual Library Operating System

Anil Madhavapeddy and David J. Scott

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

<https://queue.acm.org/detail.cfm?id=2566628>

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

Viewpoint

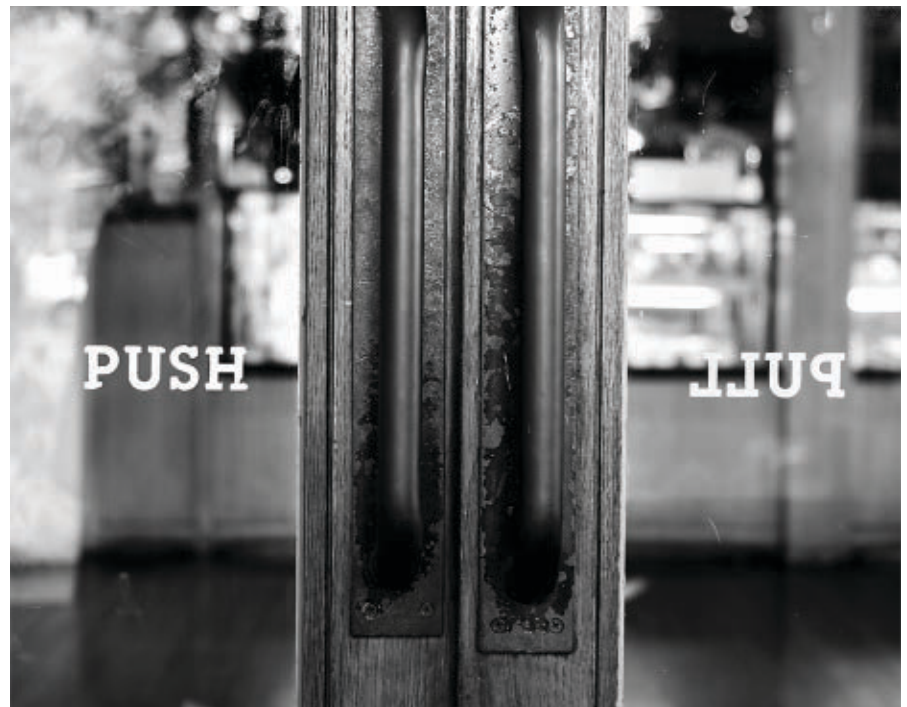
Push Versus Pull

Flipping the publishing business model.

THE ACADEMIC PUBLISHING industry provides an invaluable service to the academic research community. This publication infrastructure allows for the sharing of knowledge, facilitates new discoveries, and supports a vibrant exchange of ideas. Universities require and support the publication infrastructure and the associated peer-review vetting process through access subscription fees. A by-product of such scholarly activities and dissemination is that research institutions use peer-reviewed publications to make hiring, retention, promotion, and tenure decisions. Open access pushes publication costs onto authors (and their funding agencies), which devalues access subscription fees paid by institutions. The weak link for research vetting and dissemination is the business model to sustain the academic publication ecosystem.

Nearly everyone can agree that the free exchange of research ideas via publications is desirable and beneficial. In a financially strained environment for public and private academic institutions in the U.S., as well as numerous international institutions, particularly in the developing world, there is a growing interest to make such access to publications available at little or no cost.

The access subscription fee model represents a “Push Model” for financing publications, whereby publishers push access onto institutions, for an access subscription fee. An alternative “Pull Model” positions publishers to pull submissions to their journals. In such a model, no access subscription fees are paid, and hence, anyone can access research publications. What



institutions would pay are *submission* subscription fees, an institution-wide fee that permits their entire faculty to submit papers to a particular journal or a publisher’s portfolio of journals.

By design, the Pull Model is a systemized version of open access, where institutional submission subscription fees can be drawn from multiple sources (libraries, indirect cost funds, gift funds, or direct costs from research grants). What this does, however, is empower institutions to provide input on publication outlets that are valued, and provide their faculty the mechanism to publish in such journals. If researchers choose to submit research to journals for which their institution does not pay a submission subscription fee, researchers can

still submit by paying a single paper submission fee, akin to an open access fee. This positions the Pull Model as a natural extension of the current Open Access model.

The publishing ecosystem is complex. A simple shift from Push to Pull will not end the publishing business model debate. The quality of research is often associated with the perceived quality of the journals (often based on the rigor and integrity of the peer review process) where they are disseminated, as well as quantitative measures like impact factors and citation counts. Several professional societies publish their own journals, which provide a service to their members, and quite often, are a reliable stream of revenue, funded

primarily by academic/research institutions based on access subscription fees. In some instances, professional society membership may be partially driven by access to their journals. For-profit publishers like Springer, Elsevier, and Taylor and Francis, among others, serve a similar purpose as professional societies, in regard to publications.

By inverting the business model for publications based on submission rather than access, anyone can access published material. This is the future for academic research publications. One consequence of the Pull Model is that it places more responsibility on institutions and faculty to choose which publications to pay institutional submission subscription fees. Such a free market publication model will also put pressure on for-profit predatory publications (with thinly veiled review processes) to reevaluate their role in the publication ecosystem, since if no institutions pay their institutional submission fee, they will rely on per-paper submission fees (as they do now), which may not be viable in this new publication environment. Moreover, the Pull Model will implicitly create a journal hierarchy (and ranking), based on the number and reputation of institutions willing to pay their submission subscriptions fees. Such a market-based approach to fund the publishing ecosystem will enhance the value of quality, peer-reviewed publications and devalue predatory journals.

The proposed Pull Model may be viewed as unfair to research institutions, who contribute the preponderance of journal content, while also paying a significant portion of submission subscription fees. To assess this conjecture, in 2016, the ACM Digital Library (DL) generated over US\$20M in revenue, publishing approximately 24,000 papers by 44,000 authors. Assuming a 30% acceptance rate, this translates into approximately 80,000 papers submitted and US\$250 per submitted paper. Given

that there were over 7,100 institutions that had one or more authors on these papers, a case study can be constructed that parses these institutions into three tiers (see the table here), based on their submissions volume.

To determine what each of the three tiers listed in the accompanying table would pay as a submission subscription fee, a baseline of US\$500 per submission was used, which is in line with the high end of ACM's open access fees and the assumed acceptance rate. Tier I and II institutions would pay submission subscription fees of US\$16,000 and US\$10,000, respectively. The total amount collected by ACM for their DL would be the same as what it is currently collecting based on access subscription fees. The result of this allocation would mean that the highest volume submitting institutions would pay a slightly higher fee than they current do (for example, the University of Illinois at Urbana-Champaign paid just under \$15,000 annually in 2016 for their access subscription.) Schools that have few submissions would pay less than what they do today. Of course, these numbers represent rough estimates, since once implemented, equilibrium pricing will be reached based on supply and demand pressures. For example, it is difficult to predict how Tier III institutions will react to this pricing, since many of their authors may have co-authors in higher tiers (and hence, avoid any fee). However, by including all institutions with up to 20 submissions in this tier (based on the 30% acceptance rate), some of the revenue generation uncertainty is mitigated.

Based on this data, institutions with few submitters may be viewed as free-loaders in the system, gaining access to everything while contributing no content. I would counter that such thinking focuses entirely on a single dimension of value (cost), and does not recognize the high value that the publication eco-

system provides to research institutions and their internal faculty review process. One can even argue that research institutions have been underpaying for this service, and that the Pull Model rights this transgression, while stabilizing the current peer-review publishing ecosystem. These points suggest that beyond economic factors, the Pull Model has more potential benefits than detriments to research institutions. Moreover, the current access subscription fee paradigm is unsustainable and being challenged by sites like Sci-Hub,¹ which allows anyone to access scientific research articles at no cost.

The case study given here is but one possible scenario; specifics of the proposed subscription-submission-fee based Pull Model must be worked out among the stakeholders. The transition from a Push Model to a Pull Model will likely be traversed via incremental steps from one paradigm to the other. In fact, the transition period may be highly problematic, since no institution wants to pay both submission and access subscription fees. One option will be a phased transition, whereby some journals are Push and some are Pull. The transition can also be executed on an institution by institution basis. Both these transitions can be done by subject area, or across all areas, balancing the volume of submissions so that the transitions affect a comparable number of submitters in different communities. During such transition periods, submission subscription fees may be instituted for certain journal, while access subscription fees will be reduced by a commensurate amount. For the transition to be effective, numerous questions must be addressed. What factors will go into setting submission subscription fees? Would it be based on historical submission data or size of institution (number of faculty, number of students, size of funded research programs)? Would such fees be sufficient for the publishers to continue to deliver their products? It is reasonable to expect that the actual aggregate amounts paid by institutions will be similar to what is being expended today? At first glance, it may be that research institutions will end up paying more during this transition period, until some type of steady state is reached. Clearly, the pathway from Push to Pull will be fraught with

Pull Model case study.

Tier	Number of Papers Published Range	Number of Institutions	Percentage of Total Papers	Average Number of Submissions	Proposed Submission Subscription Fee (US\$)	Average Cost per Submission (US\$)
I	44+	70	19%	220	16,000	75
II	6-44	850	54%	50	10,000	200
III	0-6	6,250	27%	3.5	—	500

an endless stream of questions, with each stakeholder protecting their position. The Pull Model may also lead to for-profit publishers abandoning the academic publishing market (if it becomes unprofitable for them to exist), relinquishing this responsibility to not-for-profit entities who would fill the void. Laasko et al.² provides an overview of approaches to convert to open access, some of which can be used to guide the Pull Model transition and mitigate pitfalls along the way.

Time is becoming a factor in this debate, as the seeds of such a transformation are already in motion. Vogel⁵ notes that universities in Germany decided to not pay subscription fees to Elsevier, effectively challenging the subscription fee model. Park and Seo³ outline a Korean publishing service that facilitates open access. Satlow⁴ provides a commentary on separating the review process from the dissemination process. By separating the key facets of the publishing ecosystem, the proposed idea provides an à la carte menu for financial support, which shares some of the free market aspects of the Pull Model.

Reputable society publications and publishers should welcome such a business model shift. Some academic institutions may find ways to avoid paying submission fees while having their researchers still publish their research; however, the sheer diversity of faculty and their publication needs will make this a challenge. In the Push Model, researchers who gain access without paying access fees can remain anonymous. In the Pull Model, with submission and eventual publication, researchers who avoid paying submission fees will be exposed by who they list as their institution affiliation and/or co-authors. Therefore, the Pull Model creates a transparency (for submissions) that the Push Model is challenged to achieve (for access).

The best consequence of the proposed Pull Model is access for all. It also introduces a free market mechanism for scholarly publications, whereby publishers must compete for institution submission subscription fees, by establishing themselves to be worthy outlets for dissemination, maintaining their reputation for quality, and preserving the integrity of the peer-review process. Lastly, it

encourages institutions and their faculty to work more closely in assessing publication quality. With these ends in mind, the future of publications will continue to change, and the Pull Model, though disruptive to the existing publishing ecosystem, is one step to initiate a discussion on such a transformation. ■

References

1. Bohannon, J. Who's downloading pirated papers? Everyone. *Science* 352, 6285, 508–512.
2. Laasko, M., Solomon, D., and Bjork, B.-C. How subscription-based scholarly journals can convert to open access: A review of approaches. *Learned Publishing* 29, 4 (Apr. 2016), 259–269.
3. Park, M. and Seo T.-S. Creating a national open access journal system: The Korean journal publishing service. *Journal of Scholarly Publishing* 48, 1 (Jan. 2016), 53–67.
4. Satlow, M. Academic publishing: Toward a new model. Commentary, *Chronicle of Higher Education* 62, 38 (June 2016).
5. Vogel, G. German researchers start 2017 without Elsevier journals. *Science* 355, 6320 (2017), 17.

The author thanks two anonymous reviewers for their comments, resulting in a significantly improved Viewpoint, which was also written with support to the author from the National Science Foundation (CMMI-1629955). Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not reflect the views of the U.S. government, or the National Science Foundation.

Sheldon H. Jacobson (shj@illinois.edu) is Founder Professor in Computer Science at the University of Illinois at Urbana-Champaign, USA.

Copyright held by author.



Publishing at the forefront of computer science and engineering.



mitpress.mit.edu/computing

Viewpoint

Smartphones, Contents of the Mind, and the Fifth Amendment

Exploring the connection qualities between smartphones and their users.

PAPERS ARE THE OWNER'S goods and chattels; they are his dearest property, and are so far from enduring a seizure, that they will hardly bear an inspection ..."

—Lord Camden,
Entick v. Carrington (1765)

"I write separately, however, just to make explicit what is implicit in the analysis of that opinion: that the Fifth Amendment provides absolutely no protection for the contents of private papers of any kind."

—Justice O'Connor,
United States v. Doe (1984)

Smartphones are both ubiquitous—more than two billion are in use throughout the world today—and they collect a vast amount of personal data.⁸ A great deal of attention has been placed on the data itself, with the 2015 Apple/FBI confrontation providing a prominent example. The FBI obtained an iPhone that had been used by one of the terrorists killed in the December 2015 attack in San Bernardino, CA.⁹ The FBI requested an order from the U.S. District Court for the Central District of California directing Apple to create and provide an operating system that would bypass the phone's defenses, giving the FBI far easier access to whatever data remained on the phone. The denouement of this court



confrontation would have been interesting—the FBI's reliance on the 1789 All Writs Act could have had far-reaching consequences^a—but in the end the FBI apparently obtained access to the iPhone's data through other means.

^a An overview of the case can be found at <http://bit.ly/2DFqdzD>. An amicus brief filed by 32 law professors provides more detail regarding the applicability of the All Writs Act (see "Amicus Curiae Brief Of Law Professors In Support Of Apple," Case No. 5:16-CM-00010-SP).

This case is but one of hundreds that points to the importance of smartphones in our everyday lives and, perhaps more importantly, to the ability of smartphones to record the details of those lives.

This Viewpoint suggests the data-centric focus ignores an equally important issue: the nature of the relationship between smartphones and their users. I begin with a brief review of the erosion in the belief in

a mind-body separation and a growing recognition that the boundaries between the individual and the “outside” world are far more tenuous than once thought. As a result, interrelated strands of philosophy, artificial intelligence, and psychology now point to the conclusion that the smartphone, as used, is more an extension of the user’s mind than simply a useful artifact. This raises a host of issues that suggest a reconsideration of the smartphone’s legal status, or at least motivates additional protective measures for the user.

An Extension of the Self

I begin with a brief comparison of the differing approaches to cognition attributed to René Descartes and Martin Heidegger. In *Meditations on First Philosophy* (1641), Descartes asserted that the mind is a non-physical, non-spatial substance that is separate and distinct from the world of physical objects, a world that includes our bodies. Given this distinction, mind, cognition, and intelligence are explained in terms of “internal” representation and computation. Put simply, external objects cause sensations in the thinker’s mind, which in turn cause perceptions that are arranged into representations of the outside world. When the thinking thing thinks about the outside world, it performs computations on these internal representations.

In his 1922 work, *Being and Time*, Martin Heidegger threw out Descartes’ mind-body distinction. According to Heidegger, the fundamental basis for human existence is our “being in the world,” something Heidegger called *dasein*. With *dasein*, interaction with the world takes precedence over detached contemplation, and internal representations become an unnecessary hypothesis.

In his *Phenomenology of Perception* Maurice Merleau-Ponty extended Heidegger’s approach into a theory of embodied relations.⁶ When one develops the skill of handling an object, the object is incorporated into one’s bodily framework, or “schema.” Merleau-Ponty gave several examples, including that of the skilled typist and his typewriter (though the example is dated, it is easily adapted for the 21st century). Using the language of cog-

nitive extension—incorporating the typewriter keys into the self—he described in the following way how the typist interacts with the typewriter: “When the typist performs the necessary movements on the typewriter, these movements are governed by an intention, but the intention does not posit the keys as objective locations. It is literally true that the subject who learns to type incorporates the keyboard space into his bodily space.”⁶

For purposes of this Viewpoint, I will loosely group Heidegger and Merleau-Ponty under the heading of phenomenology. The distinction between the Cartesian and the phenomenological approaches to the mind can be seen in the differing emphases of AI research programs. Many early AI experiments adopted a Cartesian approach, establishing internal (code space) representations and then performing simple tasks through computations on those representations. These early experiments quickly ran into the frame problem: If the state of the world changes, how is a programmed entity to determine which elements of its internal representation have changed, and which have stayed the same? Early AI researchers sought to mitigate the frame problem by drastically constraining the scope of their artificial worlds. While this approach often lead to interesting results, it made it extremely difficult to model everyday interaction with any precision.

More recent AI researchers such as Rodney Brooks and Lucy Suchman have avoided the frame problem by eschewing internal representations and modeling cognition in the context of “situated action.”^{2,7} This non-representational approach to AI has connections to several generations of cognitive psychologists who have questioned the Cartesian model and pursued an extended, interactive approach to cognition. In 1904, William James published a paper entitled “Does ‘Consciousness’ Exist?” in which he challenged the existence of a mind that contains representations of an external world.⁴ James claimed we perceive the world “directly,” as opposed to through the mediation of internal representations. James Gibson extended this approach in his *Ecological Approach to Visual Perception*.³ Expressly echoing the work of

Calendar of Events

April 9–13

ICPE ‘18: ACM/SPEC International Conference on Performance Engineering
Berlin, Germany,
Contact: Matthew Forshaw,
Email: matthewforshaw@gmail.com

April 9–13

SAC 2018: Symposium on Applied Computing
Pau, France,
Sponsored: ACM/SIG,
Contact: Hisham M. Haddad,
Email: hhaddad@kennesaw.edu

April 21–26

CHI ‘18: CHI Conference on Human Factors in Computing Systems
Montreal, QC, Canada,
Sponsored: ACM/SIG,
Contact: Regan Lee Mandryk,
Email: regan@cs.usask.ca

April 23–26

EuroSys ‘18: 13th EuroSys Conference 2018
Porto, Portugal,
Sponsored: ACM/SIG,
Contact: Rui Oliveira,
Email: rco@di.uminho.pt

May

May 4–6

I3D ‘18: Symposium on Interactive 3D Graphics and Games,
Sponsored: ACM/SIG,
Contact: Morgan McGuire,
Email: morgam@cs.williams.edu

May 16–18

GLSVLSI ‘18: Great Lakes Symposium on VLSI 2018
Chicago, IL,
Sponsored: ACM/SIG,
Contact: Deming Chen,
Email: dchen@illinois.edu

May 23–25

SIGSIM-PADS ‘18: SIGSIM Principles of Advanced Discrete Simulation
Rome, Italy,
Sponsored: ACM/SIG,
Contact: Alessandro Pellegrini,
Email: pellegrini@dis.uniroma1.it

Merleau-Ponty, Gibson suggested that our perception is direct and not supplemented through mental representations. He further characterized our interaction with the world in terms of “affordances”—opportunities, or invitations to action. More recent work in extended cognition has used dynamical systems theory to capture Gibson’s ideas, modeling cognition in terms of the state-space evolution of dynamical systems that include both the person and the immediate objects of her surroundings.⁵

Phenomenology, AI, and extended cognition thus suggest that when we interact with “external” objects such as our smartphones, cognition is taking place in a system that includes both our persons and the phone. But we can actually take this a step further: it is not just that the smartphone is part of the thinking thing, as it were, but that we actually offload cognitive functions from our (internal) selves onto the phone. This is not as unusual as one might think. Many of us use notes when teaching a class rather than relying on internal memory, much to the benefit of all involved. In their 2015 article, “The Brain in Your Pocket: Evidence that Smartphones are Used to Supplant Thinking,”¹ Barr et al. explored the notion of “cognitive miserliness”—the idea that humans are prone to avoiding costly analytic thought in favor of simple heuristics and mental shortcuts. Barr et al. demonstrated through a series of experiments that some people actually offload cognitive tasks onto their smartphones. The ability of a smartphone to accept such offloading distinguishes it from most objects encountered in everyday experience.

One might expect that the smartphone’s role in cognition has been incorporated into legal thinking, with various courts recognizing the uniquely personal nature of the smartphone within the context of Fourth Amendment prohibitions against illegal search and seizure and Fifth Amendment prohibitions against compulsory self-incrimination. In particular, one might think that Fifth Amendment prohibitions against the compulsory disclosure of the “contents of the mind”^b might now be tied to the cognitive as-

pects of smartphone use. One would be wrong. As I will discuss here, the law has been moving in precisely the opposite direction for a very long time.

The Fifth Amendment and Private Papers: *Entick to Fisher*

To highlight the trajectory of the law, I will use personal papers as an analogous personal tool; I begin with *Entick v. Carrington*,^c a prominent case in English Common Law that dates back to 1765. In this case one John Entick sued four Messengers of the King for entering his home and seizing his personal papers and books. The presiding judge, Lord Camden, came down firmly on Entick’s side. Lord Camden highlighted the special nature of personal papers, stating that not only was their seizure improper, but they should not have even been *inspected*: “Papers are the owner’s goods and chattels; they are his dearest property, and are so far from enduring a seizure, that they will *hardly bear an inspection*; and though the eye cannot by the laws of England be guilty of a trespass, yet where private papers are removed and carried away the secret nature of those goods will be an aggravation of the trespass, and demand more considerable damages in that respect.”—*Entick v. Carrington* (1765); emphasis added

The United States Supreme Court closely followed the *Entick* case in its *Boyd v. United States* (1886) decision, quoting it extensively.^d In this case, the E.A. Boyd and Sons firm had run afoul

c *Entick v. Carrington*, 19 Howell’s State Trials 1029 (1765).

d *Boyd v. United States*, 116 U.S. 616 (1886).

The smartphone, as used, is more an extension of the user’s mind than simply a useful artifact.

of the 1874 Customs Act, an act that made it illegal to import goods without paying the appropriate duties. It provided for penalties that included substantial fines, possible jail terms, and the forfeiture of goods. The 1874 Act also included a clause that gave real teeth to any government document request:

“[I]f the defendant or claimants shall fail or refuse to produce such book, invoice, or paper in obedience to such notice, the allegations stated in the said motion shall be taken as confessed, unless his failure or refusal to produce the same shall be explained to the satisfaction of the court.”

In short, if one did not provide the papers requested by the government, one was assumed to be guilty of whatever crime the government was attempting to prove. Boyd and Sons were compelled to produce papers that were used to incriminate them. They argued that this was compelled self-incrimination, and thus “unconstitutional and void.” A unanimous Supreme Court agreed with the Boyds, issuing a far-reaching ruling that placed personal papers *beyond the reach of law enforcement*. The Boyd Court quoted the *Entick* decision at length to make three important points:

1. Personal papers are a form of private property that is particularly precious to its owner;
2. There is no precedent in English law for the seizure of personal papers; and
3. The drafters of the Fourth and Fifth amendments were aware of the *Entick* decision, so arguments based on original intent must account for the priority of personal papers stressed in *Entick*.

The Boyd Court concluded that when it comes to the seizure of private papers, “the Fourth and Fifth Amendments run almost into each other”: “[W] have been unable to perceive that the seizure of a man’s private books and papers to be used in evidence against him is substantially different from compelling him to be a witness against himself.”

The Boyd precedent lasted for almost 100 years, but after slow and steady erosion it finally came to an end in 1976 with *Fisher v. United States*.^e In *Fisher*, the Court found that when a defendant is required to surrender

b *Curcio v. United States*, 354 U.S. 118 (1957).

e *Fisher v. United States*, 425 U.S. 391 (1976).

documents held by a third party, “no constitutional rights are touched. The question is not of testimony, but of surrender.” In his dissent, Justice Brennan lamented a lost right and, perhaps unknowingly, summarized a great deal of research in a variety of fields: “An individual’s books and papers are generally little more than an extension of his person. They reveal no less than he could reveal upon being questioned directly.”—Justice Brennan, *Fisher v. United States* (1976)

Protecting the User

We have seen that for at least 100 years, British and American law held that personal papers were not subject to search and seizure. To this day Fifth Amendment law forbids the compelled production of the contents of a defendant’s mind. Recent research into extended cognition suggests that the contents of our minds may include data on our smartphones—the modern analogue of personal papers—and yet the law allows the seizure of smartphones under a wide variety of circumstances. There appears to be a contradiction: Is there a technical or legal resolution?

The technical path seems clear. Just as an 18th-century statesman might choose to keep his personal papers in a safe, smartphones can be adapted to back up selected data to privately held cryptographic vaults. The “private” element is important, as the third-party doctrine maintains that a user has no reasonable expectation of privacy in anything given to a third party. Established in the 1976 case *United States v. Miller*^f and applied to telephony in *Smith v. Maryland*^g in 1979, the doctrine predates cellular technology (1983), the World Wide Web (1990), and commercial access to the Internet (1995). Clearly the nature of the data individuals provide to third parties has changed radically since the *Miller* and *Smith* cases were decided.

There has been some progress in this area. In a concurring opinion in *U.S. v. Jones*^h Justice Sotomayor expressed concern regarding the amount and revelatory nature of data

Smartphones have become part and parcel of our everyday lives and an extension of our thinking selves.

provided to third parties in the “digital age”: “More fundamentally, it may be necessary to reconsider the premise that an individual has no reasonable expectation of privacy in information voluntarily disclosed to third parties. E.g., *Smith*, 442 U. S., at 742; *United States v. Miller*, 425 U. S. 435, 443 (1976). This approach is ill-suited to the digital age, in which people reveal a great deal of information about themselves to third parties in the course of carrying out mundane tasks. People disclose the phone numbers that they dial or text to their cellular providers; the URLs that they visit and the email addresses with which they correspond to their Internet service providers; and the books, groceries, and medications they purchase to online retailers.”—Justice Sotomayor, *United States v. Jones*

In *Carpenter v. U.S.* the question of whether cellular location data should be made available to law enforcement without a warrant is now squarely before the U.S. Supreme Court. The Court has an opportunity to recognize the unique nature of cellular handsets and to provide at least of modicum of protection to the user.

From this author’s lay perspective, it would seem there are further legal alternatives. At one extreme courts may grant the individual immunity from prosecution based solely on any information found on his or her smartphone. My friends who are prosecutors may howl in rage, but note that this extreme step still allows for the collection of data that may be used to track down co-conspirators. At the other extreme, we may continue to treat smartphones as we would any other piece of personal physical evidence, such as

fingerprints or blood samples. Such an approach ignores the cognitive aspect of the smartphone, and leaves the law in clear tension with technology and human psychology.

There is room between these two extremes, room that might include special warrants for smartphone data that require a showing of probable cause that a specific crime within a predefined class has been committed, and that specifies the information to which law enforcement is entitled. The “super warrants” required for wiretaps have similar requirements, though wiretaps provide less information than is commonly stored in a smartphone.

All law involves striking balances. Just as our right to free speech does not extend to falsely shouting “fire” in a crowded theatre, so law enforcement may not enjoy the benefits of an illegal search. So it should be here. Smartphones have become part and parcel of our everyday lives and an extension of our thinking selves. We should be able to enjoy this technology with at least some recognition that the contents of our minds may be found outside of our physical selves. ■

References

1. Barr, N. et al. The brain in your pocket: Evidence that smartphones are used to support thinking. *Computers in Human Behavior* 48 (July 2015), 473–480.
2. Brooks, R. Intelligence without representation. *Artificial Intelligence* 47, 1–3 (Jan. 1991), 139–159.
3. Gibson, J. *Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.
4. James, W. Does ‘consciousness’ exist? *Journal of Philosophy, Psychology, and Scientific Methods* 1, (1904), 477–491.
5. Käufer, S. and Chemero, A. *Phenomenology: An Introduction*. Polity Press, Cambridge, 2015.
6. Merleau-Ponty, M. *Phenomenology of Perception*. C. Smith (trans.). Routledge, New York and London. Originally published in French as *Phénoménologie de la Perception* (1962/1945).
7. Suchman, L. *Plans and Situated Actions: The Problem of Human-Machine Communication (Learning in Doing: Social, Cognitive and Computational Perspectives)*. Cambridge University Press, 1987.
8. Wicker, S.B. *Cellular Convergence and the Death of Privacy*. Oxford University Press, 2013.
9. Wu, F. No easy answers in the fight over iPhone decryption. *Commun. ACM* 59, 9 (Sept. 2016), 20–22.

Stephen Wicker (wicker@ece.cornell.edu) is a professor of Electrical and Computer Engineering at Cornell University and a Fellow of the IEEE.

The author gratefully acknowledges the comments of the reviewers—they have made this a much better Viewpoint. The author also gratefully acknowledges the comments, editing, and general support of Sarah Wicker.

This work was funded, in part, by the NSF TRUST Science and Technology Center.

Copyright held by author.

f *United States v. Miller*, 425 U.S. 435 (1976).

g *Smith v. Maryland*, 442 U.S. 735 (1979).

h *United States v. Jones*, 132 U.S. 945 (2012).

Article development led by [acmqueue](https://queue.acm.org)
queue.acm.org

BY NICOLE FORSGREN

DevOps Delivers

In many organizations across all industries, the core value of the business is now being delivered through software. For decades, software was carefully planned and then developed and delivered in lockstep processes (called phase gate or waterfall) that mirrored other disciplines such as architecture: planning, followed by design, then development, which was then handed off to testing and QA, and finally to operations for maintenance. This carefully orchestrated process with predefined deliverables and several strict hand-offs worked well enough for a time but did not allow for flexibility, changing requirements, or—most importantly—an increasingly competitive landscape that demanded speed in the way we deliver software that allows us to respond to customer demands and security threats.

DevOps is a software development and delivery methodology that provides exactly this: increased speed and stability while delivering value to organizations

and customers. (See the *State of DevOps Reports* for an overview of the performance gains possible by adopting DevOps principles; <https://devops-research.com/research.html>). The methodology has come of age in the past several years, and organizations are adopting key DevOps practices—which include technology practices, processes that draw from the lean and agile movements, and culture—to transform their software practices. DevOps practices allow the organizations that adopt them to leverage software so they can delight their customers, beat their competitors to market, pivot quickly when needed, respond to compliance and regulatory changes, and address security threats.

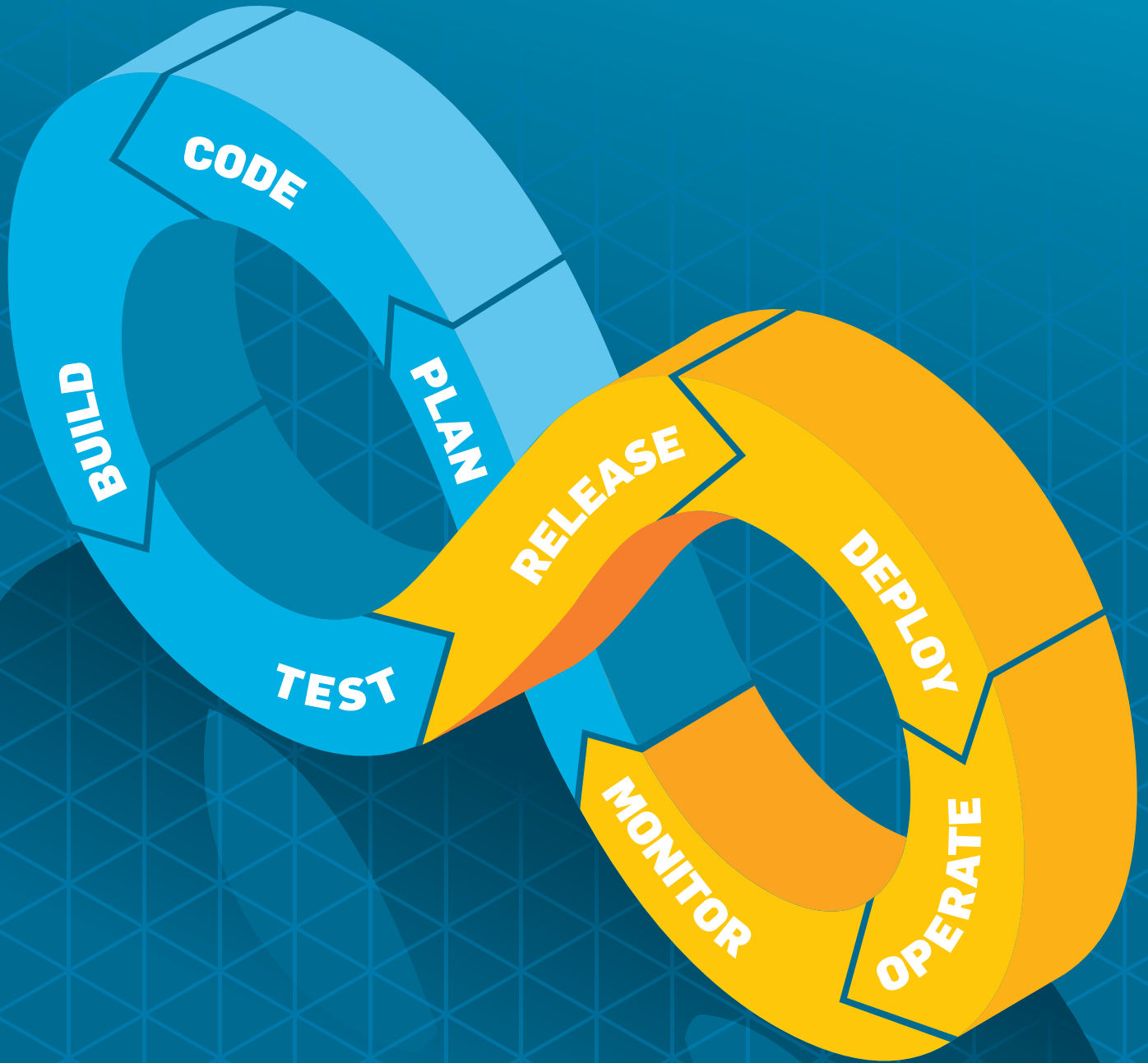
This section presents three articles that cover important aspects of DevOps. We begin with Jez Humble, author of *Continuous Delivery*, *Lean Enterprise*, and *The DevOps Handbook*, and coauthor of the forthcoming *Accelerate: The Science Behind DevOps*, discusses common objections to continuous delivery and why *all* organizations can and should be developing and delivering their software following patterns he helped pioneer a decade ago.

Bridget Kromhout, principal cloud developer advocate at Microsoft and expert in containers and Linux, shares the importance of culture in your technology transformation, and explains why containers will not solve all of your problems.

Mik Kersten, cofounder of Tasktop, teams up with me, cofounder of DORA (DevOps Research and Assessment) to present the types of data that teams must capture and collect to be sure their software development and delivery is effective, let alone successful.

We hope you find this work of great value.

Nicole Forsgren is co-founder of DORA (DevOps Research and Assessment) and coauthor of the forthcoming book *Accelerate: The Science Behind DevOps*.



Article development led by [acmqueue](https://queue.acm.org)
queue.acm.org

It's not magic, it just requires continuous, daily improvement at all levels.

BY JEZ HUMBLE

Continuous Delivery Sounds Great, but Will It Work Here?

CONTINUOUS DELIVERY IS a set of principles, patterns, and practices designed to make deployments—whether of a large-scale distributed system, a complex production environment, an embedded system, or a mobile app—predictable, routine affairs that can be performed on demand at any time. This article introduces continuous delivery, presents both common objections and actual obstacles to implementing it, and describes how to overcome them using real-life examples.

The object of continuous delivery is to be able to get changes of all types—including new features, configuration changes, bug fixes, and experiments—into production, or into the hands of users, safely and quickly in a sustainable way.

It is often assumed that deploying software more frequently means accepting lower levels of stability and reliability in systems. In fact, peer-reviewed research shows that this is not the case; high-performing teams

consistently deliver services faster and more reliably than their low-performing competition. This is true even in highly regulated domains such as financial services and government.

This capability provides a competitive advantage for organizations that are willing to invest the effort to pursue it. It allows teams to deliver new features as they are ready, test working prototypes with real customers, and build and evolve more stable, resilient systems. Implementing continuous delivery has also been shown to reduce the ongoing costs of evolving products and services, improve their quality, and reduce team burnout.

While continuous *deployment*, the practice of continuously releasing every good build of your software, is mainly limited to cloud- or datacenter-hosted services, continuous *delivery*—the set of practices described here that enables continuous deployment—can be applied in any domain.

A number of principles and practices form the continuous delivery canon (find out more at <https://continuousdelivery.com>).

Common Objections to Continuous Delivery

While people may know about continuous delivery, they often assume that “it won’t work here.” The most common objections cited are these:

- ▶ Continuous delivery is unsuitable when working in highly regulated environments.
- ▶ Continuous delivery is only for websites.
- ▶ Continuous delivery practices can’t be applied to legacy systems.
- ▶ Continuous delivery requires engineers with more experience and talent than are available here.

Here, I examine and debunk these claims, followed by a discussion of the *real* obstacles to implementing continuous delivery: inadequate architecture and a nongenerative culture.

Working in highly regulated environments. Objections to the use of continuous delivery in regulated environments



are usually of two types: first, the unfounded perception that continuous delivery is somehow “riskier;” second, the fact that many regulations are written in a way that is not easy to harmonize with the practices of continuous delivery.

The idea that continuous delivery somehow increases risk is in direct contradiction to both the entire motivation of continuous delivery—to reduce the risk of releases—and the data. Four years of data show that high performers achieve high levels of both throughput and stability.² This is possible because the practices at the heart of continuous delivery—comprehensive configuration management, continuous testing, and continuous integration—allow the rapid discovery of defects in code, configuration problems in the environment, and issues with the deployment process.

In continuous delivery, automated deployments to production-like environments are performed frequently throughout the deployment pipeline, and comprehensive automated tests are run against the builds thus deployed,

resulting in a higher level of confidence that the software being built is both deployable and fit for purpose.

In contrast, many organizations employ risk-mitigation strategies that, in practice, amount to theater: endless spreadsheets, checklists, and meetings designed more to ensure the process has been followed than to actually reduce the pain and risk of the deployment process. All this is not to say that more traditional risk-management processes can’t work when done well. Rather, this shows that continuous delivery provides an alternative risk-management strategy that has been shown to be at least as effective, while also enabling more frequent releases.

The idea that continuous delivery is at odds with common regulatory regimes also deserves closer inspection. Much of the guidance concerning the implementation of controls designed to meet regulatory objectives assumes infrequent releases and a traditional phased software delivery lifecycle complete with functional silos. It’s typically also possible,

however, to meet control objectives in a continuous paradigm. One example of this is Amazon, which in 2011 was releasing changes to production on average every 11.6 seconds, with up to 1,079 deployments in an hour (aggregated across Amazon’s production environment).⁵ As a publicly traded company that handles a substantial number of credit card transactions, Amazon is subject to both the Sarbanes-Oxley Act regulating accounting practices and the PCI DSS (Payment Card Industry Data Security Standard).

While Amazon has chosen not to describe in detail how it was able to achieve compliance despite the dizzying pace of changes, others have shared their experiences. For example, Etsy, an online handmade and vintage marketplace with more than \$1 billion in gross merchandise sales in 2013, described how it was able to meet the PCI DSS-mandated segregation of duties control while still practicing continuous deployment. Its “most important architectural decision was to decouple the cardholder data environment

(CDE) from the rest of the system, limiting the scope of the PCI DSS regulations to one segregated area and preventing them from ‘leaking’ through to all their production systems. The systems that form the CDE are separated (and managed differently) from the rest of Etsy’s environments at the physical, network, source code, and logical infrastructure levels. Furthermore, the CDE is built and operated by a cross-functional team that is solely responsible for the CDE. Again, this limits the scope of the PCI DSS regulations to just this team.”⁴

It is also important to note that segregation of duties “doesn’t prevent the cross-functional CDE team from working together in a single space. When members of the CDE team want to push a change, they create a ticket to be approved by the tech lead; otherwise, the code commit and deployment process is fully automated as with the main Etsy environment. There are no bottlenecks and delays, as the segregation of duties is kept local: a change is approved by a different person than the one doing it.”⁴

A well-designed platform-as-a-service (PaaS) can also provide significant benefits in a highly regulated environment. For example, in the U.S. federal government, the laws and policies related to launching and operating information systems run to more than 4,000 pages. It typically takes months for an agency to prepare the documentation and perform the testing required to issue the ATO (Authorization to Operate) necessary for a new system to go live.

Much of this work is implementing, documenting, and testing the controls required by the federal government’s risk-management framework (created and maintained by the National Institute of Standards and Technology). For a moderate-impact system, at least 325 controls must be implemented.

A team within the General Services Administration’s 18F office, whose mission is to improve how the government serves the public through technology, had the idea of building a PaaS to enable many of these controls to be implemented at the platform and infrastructure layer. Cloud.gov is a PaaS built using mainly open-source components, including Cloud Foundry, on top of Amazon Web Services (AWS). Cloud.gov

takes care of application deployment, service life cycle, traffic routing, logging, monitoring, and alerting, and it provides services such as databases and SSL (Secure Sockets Layer) endpoint termination. By deploying applications to cloud.gov, agencies can take care of 269 of the 325 controls required by a moderate-impact system, significantly reducing the compliance burden and the time it takes to receive an ATO.

The cloud.gov team practices continuous delivery, with all the relevant source code and configuration stored in git and changes deployed in a fully automated fashion through the course continuous integration tool.

Going beyond websites. Another objection to continuous delivery is that it can be applied only to websites. The principles and practices of continuous delivery, however, can be successfully applied to *any* domain in which a software system is expected to change substantially through its life cycle. Organizations have employed these principles building mobile apps and firmware.

Case Study: Continuous Delivery with Firmware at HP

HP’s LaserJet Firmware division builds the firmware that runs all its scanners, printers, and multifunction devices. The team consists of 400 people distributed across the U.S., Brazil, and India. In 2008, the division had a problem: it was moving too slowly. It had been on the critical path for all new product releases for years and was unable to deliver new features: “Marketing would come to us with a million ideas that would dazzle the customer, and we’d just tell them, ‘Out of your list, pick the two things you’d like to get in the next 6–12 months.’” The division had tried spending, hiring, and outsourcing its way out of the problem but nothing had worked. It needed a fresh approach.

The target set by the HP LaserJet leadership was to improve developer productivity by a factor of 10 so as to get firmware off the critical path for product development and reduce costs. There were three high-level goals:

- ▶ create a single platform to support all devices.
- ▶ increase quality and reduce the amount of stabilization required prior to release.

- ▶ reduce the amount of time spent on planning.

A key element in achieving these goals was implementing continuous delivery, with a particular focus on:

- ▶ the practice of continuous integration.
- ▶ significant investment in test automation.
- ▶ creation of a hardware simulator so that tests could be run on a virtual platform.
- ▶ reproduction of test failures on developer workstations.

After three years of work, the HP LaserJet Firmware division changed the economics of the software delivery process by adopting continuous delivery, comprehensive test automation, an iterative and adaptive approach to program management, and a more agile planning process. The economic benefits were substantial:

- ▶ Overall development costs were reduced by approximately 40%.
- ▶ Programs under development increased by approximately 140%.
- ▶ Development costs per program went down 78%.
- ▶ Resources driving innovation increased eightfold.

For more on this case study, see *Leading the Transformation: Applying Agile and DevOps Principles at Scale* by Gary Gruver and Tommy Mouser.

The most important point to remember from this case study is that the enormous cost savings and improvements in productivity were possible only with a large and ongoing investment by the team in test automation and continuous integration. Even today, many people think that lean is a management-led activity and that it’s about simply cutting costs. In reality, it requires investing to remove waste and reduce failure demand—it is a worker-led activity that can continuously drive down costs and improve quality and productivity.

Handling legacy systems. Many organizations hold mission-critical data in systems designed decades ago, often referred to as legacy systems. The principles and practices of continuous delivery, however, can be applied effectively in the context of mainframe systems. Scott Buckley and John Kordyback describe how Suncorp, Australia’s biggest insurance company, did exactly this.⁴

Case Study: Continuous Delivery with Mainframes at Suncorp


Australia's Suncorp Group had ambitious plans to decommission its legacy general insurance policy systems, improve its core banking platform, and start an operational excellence program. "By decommissioning duplicate or dated systems, Suncorp aims to reduce operating costs and reinvest those savings in new digital channels," said Matt Pancino, then-CEO of Suncorp Business Systems.

Lean practices and continuous improvement are necessary strategies to deliver the simplification program. Suncorp is investing successfully in automated testing frameworks to support developing, configuring, maintaining, and upgrading systems quickly. These techniques are familiar to people using new technology platforms, especially in the digital space, but Suncorp is successfully applying agile and lean approaches to the "big iron" world of mainframe systems.


In its insurance business, Suncorp is combining large and complex insurance policy mainframe systems into a system to support common business processes across the organization and drive more insurance sales through direct channels. Some of the key pieces were in place from the "building blocks" program, which provided a functional testing framework for the core mainframe policy system, agile delivery practices, and a common approach to system integration based on Web services.

During the first year of the simplification program, testing was extended to support integration of the mainframe policy system with the new digital channels and pricing systems. Automated acceptance criteria were developed while different systems were in development. This greatly reduced the testing time for integrating the newer pricing and risk-assessment system with multiple policy types. Automated testing also supported management and verification of customer policies through different channels, such as online or call center.

Nightly regression testing of core functionality kept pace with development and supported both functional testing and system-to-system integration. As defects were found in end-to-



The idea that continuous delivery somehow increases risk is in direct contradiction to both the entire motivation of continuous delivery—to reduce the risk of releases—and the data.



end business scenarios, responsive resolutions were managed in hours or days, not the weeks typical for larger enterprise systems.

In the process, Suncorp, which oversees several different brands, has reduced 15 complex personal and life insurance systems to 2 and decommissioned 12 legacy systems. Technical upgrades are done once and rolled out across all brands. The company has a single code base for customer-facing websites for all its different brands and products. This enables faster response to customer needs and makes separate teams, each responsible for one website, redundant.

From a business point of view, the simpler system has allowed 580 business processes to be redesigned and streamlined. Teams can now provide new or improved services according to demand, instead of improving each Suncorp brand in isolation. It has reduced the time to roll out new products and services, such as health coverage for its Apia brand customers or roadside assistance for its AAMI customers.

The investment in simplification and management of Suncorp's core systems means the company can increase its investment in all its touch points with customers. In both technology and business practices, Suncorp increased its pace of simplification, with most brands now using common infrastructure, services, and processes.

Suncorp's 2014 annual report (<http://bit.ly/2ExPnBG>) notes "simplification has enabled the Group to operate a more variable cost base, with the ability to scale resources and services according to market and business demand." Simplification activity was predicted to achieve savings of \$225 million in 2015 and \$265 million in 2016.

Developing people. Continuous delivery is complex and requires substantial process and technology investment. Some managers wonder if their people are up to the task. Typically, however, it is not the skill level of individual employees that is the obstacle to implementation but, rather, failures at the management and leadership level. This is illustrated in an anecdote told by Adrian Cockcroft, previously cloud architect at Netflix, who was often asked by Fortune 500 companies to present on Netflix's move to

the cloud. A common question they had for him was, “Where do you get Netflix’s amazing employees from?” to which he would reply, “I get them from you!”

Continuous delivery is fundamentally about continuous improvement. For continuous improvement to be effective, process improvement must become part of everybody’s daily work, which means that teams must be given the capacity, tools, and authority to do so. It’s not unusual to hear managers say, “We’d love to introduce test automation, but we don’t have time,” or “This is the way we’ve always done it, and there’s no good reason to change.” The one common factor in all high-performing organizations is they always strive to get better, and obstacles are treated as challenges to overcome, not reasons to stop trying.

Where workers are treated as fungible “resources” whose roles are to execute the tasks they are given as efficiently as possible, it’s no wonder they become frustrated and check out.

Continuous improvement cannot succeed in this type of environment. In the modern gig economy, workers are defined by the skill sets they possess, and many organizations make little effort to invest in helping their workers develop new skills as the organization evolves and the work changes. Instead, these companies fire people when their skills are no longer necessary and hire new people whose skills fit the new needs, and then wonder why there is a “talent shortage.”

These problems are related. An effective organization invests in developing people’s skills to help solve new problems, not the problems that existed at the time they were hired. One way to help achieve this is to problem-solve to remove obstacles to improved performance, learning new skills along the way: exactly what is required to effectively implement continuous delivery.

The barrier to achieving this is organizational culture, particularly the way leaders and managers behave.

Overcoming Obstacles to Continuous Delivery

The principles and practices of continuous delivery can be implemented in all kinds of environments, from mainframes to firmware to those that are highly regulated, but it’s certainly not easy. For example, Amazon took four years to re-architect its core platform to a service-oriented architecture that enabled continuous delivery.³ Typically, the biggest obstacles to this transformation are organizational culture and architecture.

Culture. What is culture? Edgar Schein, author of *The Corporate Culture Survival Guide*, defines it as “a pattern of shared tacit assumptions that was learned by a group as it solved its problems of external adaptation and internal integration, that has worked well enough to be considered valid and, therefore, to be taught to new members as the correct way to perceive, think, and feel in relation to those problems.”⁶

There are many models of culture, but one created by Ron Westrum⁷ illustrated in Figure 1, has been used to research the impact of culture on digital systems. Westrum’s research emphasizes the importance of creating a culture where new ideas are welcomed, people from across the organization collaborate in the pursuit of common goals, people are trained to bring bad news so it can be acted upon, and failures and accidents are treated as opportunities to learn how to improve rather than as witch-hunts.

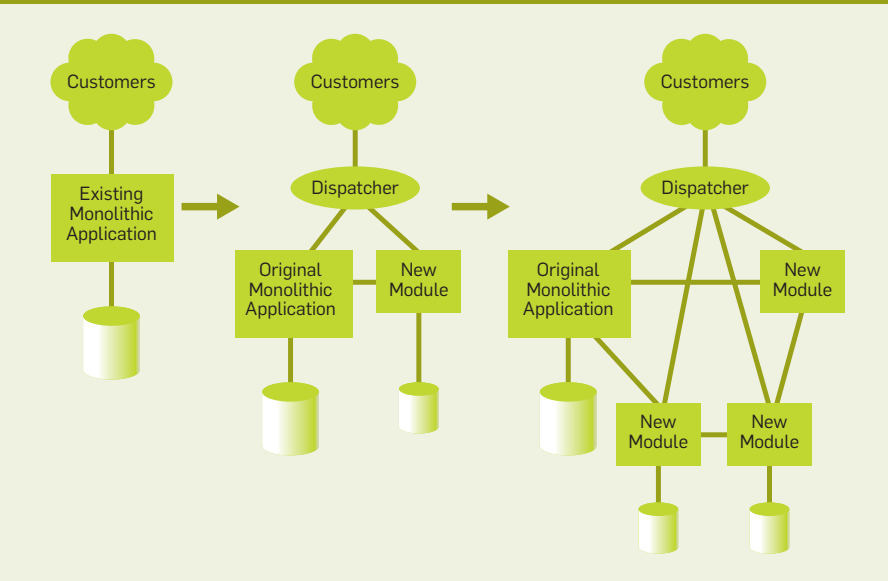
The DevOps movement has always emphasized the primary importance of culture, with a particular focus on effective collaboration between development teams and IT operations teams. Research shows that a win-win relationship between development and ops is a significant predictor of IT performance. Practitioners in the DevOps movement have also used a number of tools to help organizations process information more effectively, such as ChatOps (<https://www.youtube.com/watch?v=NST3u-GjjFw>), blameless post-mortems (<http://bit.ly/2C2Bud0>), and comprehensive configuration management (<http://bit.ly/2E1eh0R>).

Indeed, the highest-performing companies don’t wait for bad things to happen in order to learn how to improve; they create (controlled) accidents on a regular basis so as to learn

Figure 1. Westrum’s three cultures model.

Pathological (power-oriented)	Bureaucratic (rule-oriented)	Generative (performance-oriented)
Low cooperation	Modest cooperation	High cooperation
Messengers shot	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

Figure 2. The strangler application.



more quickly than the competition. Netflix took this to a new level with the Simian Army, which is constantly breaking the Netflix infrastructure in order to continuously test the resilience of its systems.

Architecture. In the context of enterprise architecture, there are typically multiple attributes to be concerned about—for example, availability, security, performance, usability, and so forth. Continuous delivery introduces two new architectural attributes: testability and deployability.

In a *testable architecture*, software is designed such that developers can (in principle, at least) discover most defects by running automated tests on their workstations. They shouldn't have to depend on complex, integrated environments to do most acceptance and regression testing.

In a *deployable architecture*, deployments of a particular product or service can be performed independently and in a fully automated fashion, without the need for significant levels of orchestration. Deployable systems can typically be upgraded or reconfigured with zero or minimal downtime.

Where testability and deployability are not prioritized, much testing requires the use of complex, integrated environments, and deployments are “big bang” events that require many services be released at the same time because of complex interdependencies. These big bang deployments require many teams to work together in a carefully orchestrated fashion with many hand-offs and dependencies among hundreds or thousands of tasks. Such deployments typically take many hours or even days, and require scheduling significant downtime.

Designing for testability and deployability starts with ensuring that products and services are composed of loosely coupled, well-encapsulated components or modules.

A well-designed modular architecture can be defined as one in which it is possible to test or deploy a single component or service on its own, with any dependencies replaced by a suitable test double, which could be in the form of a virtual machine, stub, or mock. Each component or service should be deployable in a fully automated fashion on developer workstations, in test

environments, or in production. In a well-designed architecture, it is possible to achieve a high level of confidence that the component is operating properly when deployed in this fashion.

To aid the independent deployment of components, creating versioned APIs that have backwards compatibility is worth the investment. This adds complexity to systems, but the flexibility gained in terms of ease of deployment will pay for it many times over.

Any true service-oriented architecture should have these properties—but, unfortunately, many do not. The microservices movement, however, has made explicit priorities of these architectural properties.

Of course, many organizations are living in a world where services are distinctly hard to test and deploy. Rather than re-architecting everything, we recommend an iterative approach to improving the design of an enterprise system, sometimes known as evolutionary architecture.¹ In the evolutionary architecture paradigm, we accept that successful products and services will require re-architecting during their life cycles because of the changing requirements placed on them.

One pattern that is particularly valuable in this context is the strangler application, shown in Figure 2. In this pattern, a monolithic architecture is iteratively replaced with a more componentized one by ensuring new work is done following the principles of a service-oriented architecture, while accepting that the new architecture may well delegate tasks to the system it is replacing. Over time, more functionality will be performed in the new architecture, and the old system being replaced is “strangled.” (See <https://www.martinfowler.com/bliki/StranglerApplication.html>.)

Conclusion

Continuous delivery is about reducing the risk and transaction cost of taking changes from version control to production. Achieving this goal means implementing a series of patterns and practices that enable developers to create fast feedback loops and work in small batches. This, in turn, increases the quality of products, allows developers to react more rapidly to incidents and changing requirements and, in turn, build more stable and higher-quality

products and services at lower costs.

If this sounds too good to be true, bear in mind: continuous delivery is not magic. It's about continuous, daily improvement at all levels of the organization—the constant discipline of pursuing higher performance. As presented in this article, however, these ideas can be implemented in any domain; this requires thoroughgoing, disciplined, and ongoing work at all levels of the organization. Particularly hard, though essential, are the cultural and architectural changes required.

Nevertheless, as organizations of all types and sizes from fintech startups to the U.S. government implement these ideas, they have transitioned from being exceptional to standard. If you have not yet started on this path, don't worry—it can be achieved, and the time to begin is now. □

Related articles on queue.acm.org

The Hidden Dividends of Microservices

Tom Killalea

<https://queue.acm.org/detail.cfm?id=2956643>

A Conversation with Tim Marland

<https://queue.acm.org/detail.cfm?id=1066063>

The Responsive Enterprise: Embracing the Hacker Way

Erik Meijer and Vikram Kapoor

<https://queue.acm.org/detail.cfm?id=2685692>

References

1. Ford, N., Parsons, R. and Kua, P. *Building Evolutionary Architectures: Support Constant Change*. O'Reilly Media, 2017; (<http://evolutionaryarchitecture.com>).
2. Forsgren, N. et al. *State of DevOps Report*. Puppet and DevOps Research and Assessment LLC, 2014–2017; (<https://devops-research.com/research.html>).
3. Gray, J. A conversation with Werner Vogels. *acmqueue* 4, 4 (2006); <http://queue.acm.org/detail.cfm?id=1142065>.
4. Humble, J., O'Reilly, B. and Molesky, J. *Lean Enterprise: How High Performance Organizations Innovate at Scale*. O'Reilly Media, 2014, 280–281.
5. Jenkins, J. Velocity culture (the unmet challenge in ops). O'Reilly Velocity Conference, 2011; <http://assets.en.oreilly.com/1/event/60/Velocity%20Culture%20Presentation.pdf>.
6. Schein, E. *The Corporate Culture Survival Guide*. Jossey-Bass, 1999.
7. Westrum, R. A typology of organizational structures. *BMJ Quality and Safety* 13, 2 (2004); http://qualitysafety.bmj.com/content/13/suppl_2/ii22.

Jez Humble is coauthor of *The DevOps Handbook*, *Lean Enterprise*, and *Continuous Delivery*. He is currently researching how to build high-performing teams at his startup, DevOps Research and Assessment LLC, and teaching at UC Berkeley, CA, USA.

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

Article development led by [acmqueue](https://queue.acm.org)
queue.acm.org

**Complex socio-technical systems are hard;
film at 11.**

BY BRIDGET KROMHOUT

Containers Will Not Fix Your Broken Culture (and Other Hard Truths)

WE FOCUS SO often on technical anti-patterns, neglecting similar problems inside our social structures. Spoiler alert: The solutions to many difficulties that seem technical can be found by examining our interactions with others. Let's talk about five things you will want to know when working with those pesky creatures known as humans.

1. Tech Is Not a Panacea

According to noted thought-leader Jane Austen, it is a truth universally acknowledged that a techie in possession of any production code whatsoever must be in want of a container platform.

Or is it? Let's deconstruct the unspoken assumptions. Don't get me wrong—containers are delightful! But let's

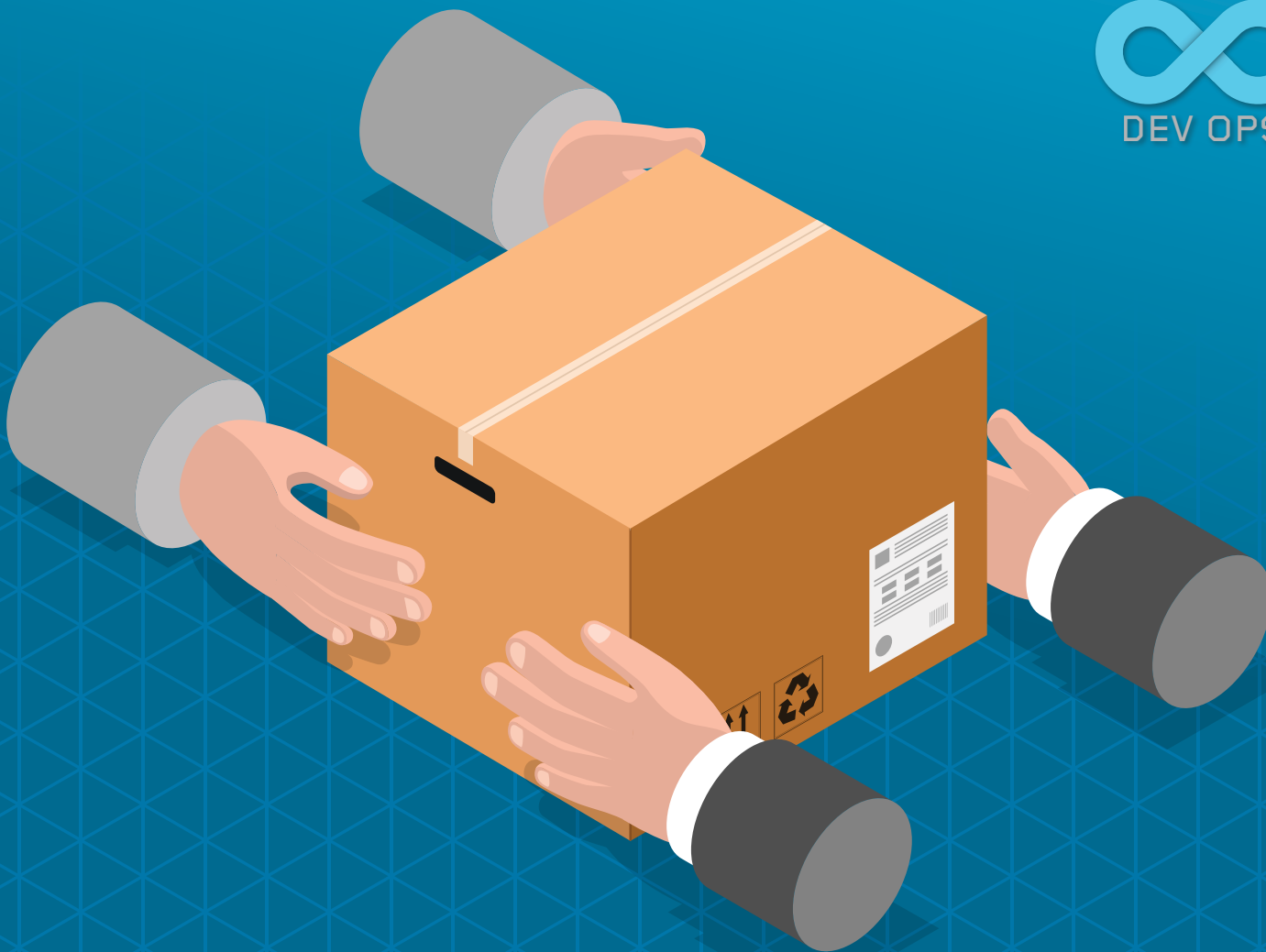
be real: We are unlikely to solve the vast majority of problems in a given organization via the judicious application of kernel features. If you have contention between your ops team and your dev team(s)—and maybe they are all facing off with some ill-considered DevOps silo inexplicably stuck between them—then cgroups and namespaces won't have a prayer of solving that problem.

Development teams love the idea of shipping their dependencies bundled with their apps, imagining limitless portability. Someone in security is weeping for the unpatched CVEs, but feature velocity is so desirable that security's pleas go unheard. Platform operators are happy (well, less surly) knowing they can upgrade the underlying infrastructure without affecting the dependencies for any applications, until they realize the heavyweight app containers shipping a full operating system are not being maintained at all.

Ah, but, you say, at our org we do this right (for sufficiently non-terrible values of "right")! We inject credentials at run time, and run exactly the same containers in every environment. Perhaps we even ship lightweight containers with only statically linked binaries. Okay, but traffic patterns and data tested across various environments are likely not close to the same. As the old joke goes:

*Proposal: Rename 'staging' to 'theory.'
"It works in theory, not on production."
—Najaf Ali*

There is no substitute for experimentation in your real production environment; containers are orthogonal to that, while cross-org communication is crucial to clarity of both purpose and intent. Observability being key is a fundamental tenet of the gospel according to Charity Majors. The conflicts inherent in misaligned incentives continue to manifest no matter where the lines of responsibilities are drawn. Andrew Clay Shafer calls the state of any running system "continuous partial failure;" good tooling is necessary



(but not sufficient) to operate a robust fault-tolerant system.

Relying on health checks in your continuous delivery is all well and good, until the health check is full of deceit and lies because it says everything is 200 OK, and all those instances are staying in the load balancer, and yet nothing is *working*. (My on-call PTSD may be showing.)

In a world of ever-increasing complexity, how do we evaluate our progress toward a Container Store utopia? How do we know when to course-correct? How do we react when it seems like there is always something new we should have done last month? Must I really orchestrate my containers? Could they maybe just do some improv jazz?

2. Good Team Interactions: Build, Because You Can't Buy

We hold in our heads the intricate composition of our complex distributed systems, and there is increasingly even

more state we cannot fit into those necessarily incomplete mental models. Microservices are not defined by lines of code so much as by the scope and breadth an individual service covers. And, no, microservices will not prevent your two-pizza teams from needing to have conversations with one another over that pizza. (Also, how hungry are these people, and how large are the pizzas? So many unanswered questions!)

Adrian Cockcroft points out that a monolith has as much complexity as microservices; it's just hidden. Okay, so we are going to deconstruct that dreaded monolith and keep on rockin' in the microservices world. That will solve everything! Clean abstractions and well-defined handoffs sound great, until you realize that you are moving the consequences of decisions (and the conflict inherent in any set of trade-offs) into another part of your stack, which Tim Gross calls "conservation of complexity."

Breaking into individuated teams does not change the fact that the teams have to agree where the boundaries lie at any given moment. Writing in the 1960s, Mel Conway could have been talking about today—except for the title, because "How Do Committees Invent?" very much buries the lede; today it would be a clickbait listicle.

Conway wrote that any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure. This came to be known as Conway's Law.

Contrary to popular belief, Conway's Law does not say your org chart has to look exactly like your death-star software architecture diagram, and a cursory inspection of either would lead us to believe that plan would never scale, anyway. No matter what design decisions you make around thermal exhaust ports, no amount of industrial-strength job scheduling makes your or-

ganization immune to Conway's Law.

The most important word in Conway's Law is *communication*. In your excitingly deconstructed world, how is communication about breaking changes handled? What about schema migrations, because state is real? (The pesky thing about storing state is that your value often exists there. The money types get awfully touchy about anything that could adversely affect systems of record.)

Creative problem solvers have a way of routing around process that we find inconvenient. If your heavyweight change control process applies *except* in case of emergencies, then (spoiler alert) you are going to see a surprisingly high rate of sorry-not-sorry "emergencies."

Dunbar's number, a cognitive limit on the number of people with whom an individual can maintain stable social relationships, is demonstrably valid. If working in a larger organization, you will need to communicate in smaller groups, but those groups should be cross-functional to eliminate bottlenecks and misunderstandings. Communication does not just mean talking with our human voices or replying to interminable email threads, either; much like Consul's gossip protocol, we need cross-talk in our orgs to keep communication flowing.


We have all heard "we only communicate through APIs," but technology alone does not solve all communication problems. If you launch a new version of the API, does that mean you will ever be able to deprecate the old one? Is well-labeled versioning sufficient for current needs of all your API's consumers? How about future, conflicting, overlapping needs? At some point, you will have to talk to each other. (Bring some pizza!)

3. Tech, Like Soylent Green, Is Made of People

Andrew Clay Shafer likes to opine that 90% of tech is tribalism and fashion. Tools are important, but people are an integral part of any human-designed complex system. We have all seen the ridiculously expensive migrations gone wrong, the years-long lift-and-shift projects that accomplish only a fraction of their goals because of the necessity of maintaining business continuity, the "DevOps initiatives" that



In a world of ever-increasing complexity, how do we evaluate our progress toward a Container Store utopia?



last only long enough for somebody's vice-presidency level-up to complete. Examining the motivations driving these decisions (even if reconstructed by observing consequences) can frequently reveal the probable genesis of suboptimal decisions.

Nobody is doing résumé-driven development with shell scripts; I'm willing to bet that all the janky bash ever written was meant to solve a real problem. When we start getting fancier, there are often motivations less pure than "Let's do this well," and even if there are not, intention alone does not create maintainable software. The trough of disillusionment is where we all land when dreams meet reality. Whatever slow-burning tire fire results from a given IT project, it's a sure bet it will burn for a good long while. Software is "done" when it's decommissioned; until that point, day one is short, while day two lasts until the heat death of the universe.

A good mental model is Simon Wardley's "Pioneers, Settlers, Town Planners." While a proof of concept can be "done" enough to ship it, operationalizing it takes longer, and keeping it running in production is an ongoing project. Entropy increases, as the second law of thermodynamics explains.

Obviously, striving for iterative IT improvement matters, but it's not an end state. We have all been in those meetings where people are not so much listening as just waiting for their turn to talk. Software is made of feelings, as Astrid Atkinson puts it. We need to consider our impact on each other. Certifying people in DevOps is like celebrating their graduation from kindergarten. "Congratulations! You learned not to eat the crayons and to play nicely with the other children!"

Does this mean that DevOps has failed in its promise of increased efficiency brought to you by collaboration? Not in the slightest. Talk to the fine folks at DORA (DevOps Research and Assessment)—a measurable impact shows up in the research when we center IT improvement. We cannot buy DevOps, despite what some in the ecosystem might promise that a given tool offers. We have to live it; change for the better is a choice we make every day through our actions of listening empa-

thetically and acting compassionately. Tools can and do help, but they can't make us care.

4. Good Fences Make Good Neighbors

Boundary objects and abstractions give needed structure, and containers make good boundary objects, but they do not eliminate the liminal space between the metaphorical (or all-too-real) dev and ops. When you implement microservices, how micro is micro? Even if you have a well-defined service that does one thing (somewhat) well, a good rubric is whether the service's health endpoint can answer unambiguously. If the answer to "Is this working?" is "Welllllll...", that service isn't micro enough.

Deciding what's *yours* and what's *theirs* is the basis of every sibling-rivalry détente. In Eric Brewer's CAP theorem you can pick two of consistency, availability, and partition tolerance as long as one of them is partition tolerance, because, as distributed systems expert Caitie McCaffrey puts it, "physics and math." In a distributed system that contains humans in multiple time zones, you're inevitably going to have partitions, and waiting 10 hours for headquarters to wake up and make a decision is nobody's idea of a good time. But decentralized decision making means distributing power to your human edge nodes (sometimes a hard sell).

Empowering developer choice is facilitated by containers; there is always a tension between what someone else dictates and what you are convinced you need. Making thoughtful decisions about tools and architecture can help; well-considered constraints can free us from the decisions that are not bringing us distinguishable benefit. Containers can help define scope and reach of a given tool or project, and deconstructing systems to human scale allows us to comprehend their complexity.

Being able to reproduce a build allows for separation of concerns. We want this to be effective and yet not introduce unnecessary barriers. The proverbial wall of confusion is all too real, built on the tension between having incentive to ship changes and being rewarded for stability. Building just the right abstractions that empower independent teams is worth taking the time to iterate on (and, no, nobody gets it right immediate-

ly, because "right" will evolve over time).

We want to empower people with as much agency as possible within the constraints that work for our organizations. To determine the right constraints for you, you need to talk to your teams. Think in terms of TCP instead of UDP; you will need to SYN/ACK to really understand what other humans want. Nonviolent communication, where you restate what you heard, is an effective way to checksum your human communications. (Bonus: techies will appreciate this logic!)

5. Avoiding Sadness as a Service

Hindsight being what it is, we can look back and recognize inflection points. It's more difficult to recognize change in the moment, but the days of operating your own data centers, where your unit of currency is the virtual machine, are coming to a definite middle. The hipsters among us will say that's over and sell you on serverless (which is just servers you cannot ssh into), but we are talking about the realities of enterprise adoption here, and they are about at the point of taking containers seriously. Application container clustering is better for utilization and flexibility of workload placement, and using containerized abstractions makes for better portability, including for those orgs looking toward public cloud.


W. Edwards Deming, a leader in the field of quality control, said, "It's not necessary to change. Survival is not mandatory." Change is difficult. Not changing is even worse. Tools are essential, but how we implement the tools and grow the culture and practices in our organizations needs even more attention. As it turns out, it's not mandatory to write a Markov bot to parse the front page of Hacker News, then yolo absolutely everything out to production instantly!

Whether you are just starting to implement technical and organizational change, or facing the prospect that you already have legacy microservices, it's worth considering the *why* and *how* of our behaviors, not just the *what*. If legacy were not important, you could just turn it off. But this is where your customers and money live. Glorifying exciting greenfield projects is all well and good, but the reality is that bimodal IT is a lie.

It's ludicrous to tell people that some of them have to stay in "sad mode" indefinitely, while others catapult ahead in "awesome mode." Change is on a continuum; absolutely every change ever doesn't happen at the same instant.

We succeed when we share responsibility and have agency, when we move past learned helplessness to active listening. Don't be a named pipe; you are not keyboard-as-a-service. Assuming we can all read code, putting detail in your commit messages can be a lot more useful than soon-to-be-outdated comments. Tell future-you why you did that thing; they can read but don't know what you intended. Oral tradition is like never writing state to disk; flush those buffers. There is no flowchart, no checklist, no shopping list of ticky boxes that will make everything better. "Anyone who says differently is selling something," as *The Princess Bride* teaches us. Orgs have "the way we do things" because process is the scar tissue of past failures.

You cannot take delivery of a shipping container with 800 units of DevOps, and have 600 of them go to the people in awesome mode, while the people in sad mode can look at the other 200 but not touch them. DevOps is something you do, not something a vendor implements for you with today's shiniest tools. Change for the better is a decision we make together.

Tools are necessary but not sufficient. To build a future we all can live with, we have to build it together. 

Editor's Note: To read this article complete with embedded hyperlinks, visit <https://queue.acm.org/detail.cfm?id=3185224>.

Related articles on queue.acm.org

The Verification of a Distributed System

Caitie McCaffrey

<https://queue.acm.org/detail.cfm?id=2889274>

Adopting DevOps Practices in Quality Assurance

James Roche

<https://queue.acm.org/detail.cfm?id=2540984>

Bridget Kromhout is a principal cloud developer advocate at Microsoft. She leads the devopsdays organization globally and the DevOps community at home in Minneapolis, MN, USA. She podcasts with Arrested DevOps, blogs at bridgetkromhout.com, and tweets at twitter.com/bridgetkromhout.

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

Article development led by [acmqueue](https://queue.acm.org)
queue.acm.org

**Your biggest mistake might
be collecting the wrong data.**

BY NICOLE FORSGREN AND MIK KERSTEN

DevOps Metrics

“Software is eating the world.”

— Marc Andreessen

“You can’t manage what you don’t measure.”

— Peter Drucker

ORGANIZATIONS FROM ALL industries are embracing software as a way of delivering value to their customers, and we are seeing software drive innovation and competitiveness from outside of the traditional tech sector.

For example, banks are no longer known for hiding gold bars in safes: instead, companies in the financial industry are harnessing software in a race to capture market share. Using innovative apps, banks are making it possible for their customers to do most of their daily banking in a few swipes, from depositing checks to transferring money securely between bank accounts. Moreover, the banks themselves can improve their service in a number of ways, such as using predictive analytics to detect fraudulent transactions. Other industries are seeing similar changes: cars are now computers on wheels, and even the U.S. Postal Service is in the middle of a massive DevOps transformation. Software is everywhere.

Leaders must embrace this new world or step aside. Gartner Inc. predicts that by 2020, half of the CIOs who have not transformed their teams’ capabilities will be displaced from their organizations’ leadership teams. And as every good leader knows, you cannot improve what you do not measure, so measuring the software development process and DevOps transformations is more important than ever.

Delivering value to the business through software requires processes and coordination that often span multiple teams across complex systems, and involves developing and delivering software with both quality and resiliency. As practitioners and professionals, we know that software development and delivery is an increasingly difficult art and practice, and that managing and improving any process or system requires insights into that system. Therefore, measurement is paramount to creating an effective software value stream. Yet accurate measurement is no easy feat.

Measuring DevOps. Collecting measurements that can provide insights across the software delivery pipeline is difficult. Data must be complete, comprehensive, and correct so that teams can correlate data to drive business decisions. For many organizations, adoption of the latest best-of-breed agile and DevOps tools has made the task even more difficult because of the proliferation of multiple systems of recordkeeping within the organization.

One of the leading sources of cross-organization software delivery data is the annual State of DevOps Report (found at <https://devops-research.com/research.html>).² This industry-wide survey provides evidence that software delivery plays an important role in high-performing technology-driven organizations. The report outlines key capabilities in technology, process, and cultural areas that contribute to software-delivery performance and how this, in turn, contrib-

utes to key outcomes such as employee well-being, product quality, and organizational performance.

Bolstered by this survey-based research, organizations are starting to measure their own DevOps “readiness” or “maturity” using survey data. While this type of data can provide a useful view of the potential role that DevOps can play in teams and organizations, the danger is that organizations may blindly apply the results of surveys without understanding the limitations of this methodology.

On the flip side, some organizations criticize survey-based data wholesale and instead attempt to measure or assess their DevOps readiness or maturity using system data alone. These organizations, which are creating metrics based on the system data stored in their repositories, may not understand the limitations of that methodology, either.

By understanding these limitations, practitioners and leaders can better leverage the benefits of each methodology. This article summarizes the two separate but complementary approaches to measuring the software value stream and shares some pitfalls of conflating the two. The two approaches are defined as follows:

- **Survey data.** Using survey measures and techniques that provide a holistic and periodic view of the value stream.

- **System data.** Using tool-based data that provides a continuous view of the value stream and is limited to what is automatically collected and correlated.

A Complementary Approach

Neither system data nor survey data alone can measure the effectiveness of a modern software delivery pipeline. Both are needed. A complementary approach to measurement can arm organizations with a more complete picture of their development and operations environment, address the key gaps of each approach, and provide organizations with the information they



need to develop and deliver software competitively.

As an analogy, consider how a manufacturer may track the effectiveness of a complex assembly line. Instrumentation at each step provides data on rates of flow and defects within each phase and across the end-to-end system. Augmenting that with survey data of the assembly line staff can prove invaluable—for example, discovering that a newly deployed cooperative robot is putting more physical strain on employees than was promised by the robot vendor.

Capturing that information before higher defect rates, lower employee survey scores, or even lawsuits arise can prove invaluable. In this example, the survey data provides leading indicators to system data, or provides insights that system data might not disclose at all. Whereas assembly line

manufacturing is extremely mature in terms of metrics and data collection, there is a severe lack of industry consensus on how to measure software delivery. This implies that this practice is still in its infancy. (Note that this is likely related to the relative maturity of the fields themselves: the manufacturing discipline has been around for a long time, so those who study and measure it have had several decades to perfect their craft; in contrast, software engineering is a relatively young field, making its measurement study much less mature.) As such, it is critical for organizations to understand what they can and cannot measure with which approach, and what steps they must take to gain visibility into their software delivery value streams.

Using the authors’ collective decades of research and experience in collecting both survey and system

data—confirmed by in-depth discussions with hundreds of experts at dozens of global organizations who make software value-stream measurement a key part of their digital transformation—this article outlines the measures necessary for understanding your ability to develop and deliver software.

Start Building a Baseline Now

There are several reasons why both system and survey data should be used to measure the value streams that define your software-delivery processes. One of the most important is that most organizations seem to have almost no visibility or reliable measurement of their software-delivery practices.

The earlier an organization starts measurement, the earlier a baseline is established and can be used for gauging relative improvement. For a small organization, applying system metrics as the initial baseline can be easy. For example, a 20-person startup can measure MTTR (mean time to repair) using just an issue tracker such as Jira. A large organization, however, will need to include service desks and potentially other planning systems in order to identify that baseline and may not have implemented a tool that provides cross-system visibility. We recommend getting started with baseline collection immediately, and for many organizations that will mean collecting survey data while efforts to capture and correlate system data are under way.

In the absence of complete system measurements, comprehensive surveys can provide a holistic view of your system relatively quickly (such as, within several weeks). Contrast that with full visibility of your system provided by system-based metrics. Getting end-to-end system data can be a long journey as you first must deploy a measurement solution across systems, and then make sure that cross-system integration is in place so the data can be properly correlated. Modern value-stream metrics are making this easier, but for many organizations this has been a multiyear project.

While it is important to start as early as possible to get the benefits of system data, deploying survey data provides an almost immediate value and



Leaders must embrace this new world or step aside. Gartner Inc. predicts that by 2020, half the CIOs who have not transformed their teams' capabilities will be displaced from their organizations' leadership teams.



source of baseline information. This is valuable both for baselining current and future survey data, and for comparing survey with system data once in place. Therefore, it is best to capture a system baseline with survey measures now while continuing to build out system-based metrics.

What happens once you are fully instrumented with system-based metrics? You can continue using your survey-based metrics for both augmentation and capturing additional data that's uniquely suited for survey methods.

There are still some measures that are important to software delivery, such as cultural measures, that survey-based measures will pick up and system-based metrics may miss. In addition, having both types of metrics provides opportunities for triangulation: if your survey measures provide data that is drastically different from the data coming from your systems, this can highlight gaps in the system.

Some might say such a gap is just an area where "people lie," but if all of the people working closely with the system are lying, you might want to consider their experience as a true data point. If your engineers consistently report long build times and the system data reports short build times, could it be a configuration error in the API? Or could it be that the system-based measure is capturing only a portion of the data? Without consistently collecting insights from the professionals working with your systems, you will miss opportunities to see the full picture. The rest of this article outlines the pros and cons of each measurement type.

System-Based Metrics

System-based metrics generally refer to data that comes from the various systems of record that make up an end-to-end software delivery value stream. Important aspects of this data include:

- *Completeness.* Is the data captured from a particular system of record, such as an agile tool, complete enough to provide the kind of visibility, metrics, and reports that are the goal of the initiative? For example, if demonstrating faster time to market is the goal, are enough historicals captured to derive the trend line of how quickly

new products and features are delivered?

► *Comprehensiveness.* Is enough data captured across all systems of record? For example, to measure time to market for a customer request, you may need data from a customer/support tracking system, the roadmapping/requirements system, the agile tool, and the deployment tool chain.

► *Correctness.* Is the data sufficiently correlated to be correct? For example, if a support ticket and a defect are actually the same item but exist in two different systems, should the two systems be integrated in a way to indicate that these are the same item, or do you risk double-counting defects in this scenario?

System Data Advantages

► *Precision.* Only system-generated data can accurately show minute, second, and millisecond response times.

► *Continuous visibility.* System-generated data is particularly well suited for continuous/streaming data and real-time reporting. You can just point it to the data store and gather everything for targeted analysis later.

► *Granularity.* Data from systems can provide very granular data, allowing you to report on subsystems and components. This is useful for identifying trends and bottlenecks, but requires additional effort to create a higher-level picture of the full system. The more granular the data, the more work is required to paint a full picture.

► *Scalability.* Once the integration and visibility infrastructure is implemented, it can be pointed at all systems. This means that the solution can be scaled from getting visibility on a single project to dozens or hundreds of projects with large amounts of data.

To use an analogy to illustrate: when building a house, a contractor may use concrete for the foundation; wood/nails/screws/drywall for the walls; wiring and plumbing; brick for the exterior; paint/carpet for the finish; plus any materials for the kitchen and bath. In order to track and monitor progress, you build in monitoring to track each piece of the construction and install it as the house is built. Once installed, each and every piece of this infrastructure (specific data) can continually provide reporting and

metrics (continuous data) at subsecond intervals (precision). You can then combine and correlate (volume and scale) these to create a full picture of what is happening in your house.

System Data Challenges

► *Capturing behavior outside of the system.* This may be the most important yet most overlooked limitation in system-based data. An example is version control: your system can tell you only what is inside of it. What portion of the work being done is *not* being checked into a version control system? Common culprits include system configuration and database configuration scripts.

► *Gaining a holistic view.* Eventually, system-level data can provide a relatively full view of your system, but this requires full instrumentation, plus correlation across measures and maturity in reporting and visualization techniques so that teams can understand system state. This is a nontrivial task, especially if undertaken without the right tooling and infrastructure in place. Additionally, the holistic view should include the human aspects of the process, such as the difficulty of deployments and software sprints, which are important for understanding the sustainability of the work.

► *Capturing drifts in the system.* If any part of your system stack changes and your data collectors are not updated, your view of the system will be inaccurate. Note that this is not a characteristic of a first-class data reporting solution, but it happens in some commercial systems and in many home-grown solutions, so it is worth mentioning as a condition to watch for.

► **Cultural or perceptual measures.** If you want to measure aspects of culture, these are perceptual and should be measured with surveys. Further, any measures that come from system databases (such as HR systems) are usually poor representations of the data you're trying to collect and will be lagging indicators. That is, they will be able to measure something only after it has happened (such as someone leaving a team or an organization). In contrast, survey measures can let you measure perceptions of culture in time to act on the information.

System-based metrics are useful,

but they cannot paint a complete picture of what is happening in your software-delivery work. Therefore, it is strongly recommended that you augment your metrics with complementary survey measures.

Survey-Based Metrics

Survey-based metrics generally refer to data about systems and people (such as culture) that comes from surveys. Ideally, these surveys are sent to the people who are working on the systems themselves and who are intimately familiar with the software-development and delivery system—that is, the doers. It is better for teams to avoid surveying management and executives, because, as a recent study by Forrester shows, executives tend to overestimate the maturity of their organizations.³

Important aspects of this data include:

► *Cohesiveness.* Survey-based data is particularly good at providing a complete and holistic view of systems. This is because it can capture information about systems, processes, and culture. Measure your system periodically and at regular intervals: every four to six months.

► *Correctness.* Survey design and measurement is a well-understood discipline and can be leveraged to provide good data and insights about systems and culture. By using carefully designed surveys with statistically valid and reliable survey questions that have been rigorously developed and tested, organizations can have confidence in their survey data.

Survey Data Advantages

► *Accuracy.* When collected correctly, survey data can provide accurate insights into systems, processes, and culture. For example, you can measure system capabilities by asking teams how often key tasks are done in automated or manual ways. When designed correctly, this provides a fast and accurate measurement that can be used to baseline and guide improvement efforts.

► *A holistic view of the system.* Surveys are particularly good at capturing holistic pictures of systems, because the answers that respondents provide synthesize data related to automation, processes, and culture.

► *Triangulation with system data.* Survey data provides an alternate view of your system, allowing you to identify problems or errors when there are two contrasting views. Do not automatically discount your survey measures when this happens: there can often be cases where changes in configurations or system behavior alter the way that system data is collected, while survey measures remain true—and it is only the delta in these two measures that calls attention to changes in the underlying system.

► *Capturing behavior outside of the system.* In the discussion of system data, version control was used as an example of data that will be incomplete if it is collected only from your system. You can gain a more complete view of what is happening both within and around your system by using surveys. For example, are there situations where version control is being bypassed?

► *Cultural or perceptual measures related to the system.* Survey data provides insights into what it's like to do the work: organizational culture, job satisfaction, and burnout are important as leading indicators of work tempo sustainability and hiring/retention. Research shows that good organizational cultures drive software delivery and organizational performance,² and job satisfaction drives revenues.¹ Monitoring these proactively (through survey data) and not just reactively (through turnover metrics in HR databases) should be a priority for all technical managers and executives.

Let's return to the house analogy. When using system data, you can get detailed information from each piece of the system that is reporting. This level of detail isn't possible (or realistic) when asking people through survey questions—but you can very quickly and easily get a holistic understanding of what your system or its components are doing. For example, you can reliably ascertain if the house is in a good state: anyone can report if the house is on fire, if a room is dirty or smoky, or if an event has caused damage. This data can be gathered much faster than the time needed to instrument and then correlate and synthesize hundreds or thousands of data points. If your survey

and system measures disagree, you have great cause to start debugging the system.

Survey Data Challenges

► *Precision.* While you can query practitioners about broad strokes, you should not rely on them for detailed or specific information. When you ask about deployment frequency, your survey options increase in log scale: people can generally tell you if they are deploying software on demand, weekly, monthly, quarterly, or yearly. Those frequencies are easy to confirm with system-based metrics (when available—though that is a nontrivial metric to get from systems, because it requires getting data from several systems along the deployment pipeline).


► *Continuity of data.* Asking people to fill out surveys at frequent intervals is exhausting, and survey fatigue is a real concern. It is better to limit the frequency of big data collection through surveys—say, every six months or so.

► *Volume.* The amount of data you collect is related to how often you collect it. Experience tells us that surveys should be kept to 20–25 minutes (or shorter) to maximize participation and completion rates. There are notable exceptions: Amazon's famous developer survey was rolled out on an annual basis and took about an hour to complete, but the engineers were very interested and invested in the results, so they took the time to complete it.

► *Measures in strained environments.* If management has made it very clear that it isn't safe to be honest, or that the results will be used to punish teams, then any survey responses will be suspect. To quote the late W. Edwards Deming: "Whenever there is fear, you will get wrong figures." But, to be fair, system-based metrics are equally suspect in unsafe and fearful environments, and possibly more so. Why? Because it only takes a single person with root access to slip a rogue metric into the system and a tired person on peer review or a CAB (change approval board) to miss it (as those of us who have seen the cult classic movie *Office Space* can attest). In contrast, it takes several or dozens or hundreds of people to skew survey results en masse.

Conclusion

Software is driving value in organizations across all industries and around the world. To help deliver value, quality, and sustainability more quickly, companies are undergoing DevOps transformations. To help guide these difficult transformations, leaders must understand the technology process.

This process can be illuminated through a good measurement program, which allows team members, leaders, and executives to understand technology and process work, plan initiatives, and track progress so the organization can demonstrate the value of investments to key stakeholders. System-based metrics and survey-based metrics each have inherent limitations, but by leveraging both types of metrics in a complementary measurement program, organizations can gain a better view of their software-delivery value chain and DevOps transformation work. 

Related articles on queue.acm.org

Adopting DevOps Practices in Quality Assurance

James Roche

<http://queue.acm.org/detail.cfm?id=2540984>

Statistics for Engineers

Heinrich Hartmann

<http://queue.acm.org/detail.cfm?id=2903468>

The Responsive Enterprise: Embracing the Hacker Way

Erik Meijer and Vikram Kapoor

<http://queue.acm.org/detail.cfm?id=2685692>

References

1. Azzarello, D., Debruyne, F. and Mottura, L. The chemistry of enthusiasm. Bain and Co., 2012; <http://www.bain.com/publications/articles/the-chemistry-of-enthusiasm.aspx>.
2. DevOps Research and Assessment. 2014, 2015, 2016, and 2017 State of DevOps Reports; <https://devops-research.com/research.html>.
3. Stroud, R., Klavens, E., Oehrlich, E., Kinch, A. and Lynch, D. A dangerous disconnect: executives overestimate DevOps maturity. Forrester, 2017; <http://bit.ly/2Fs6Wjo>

Nicole Forsgren is co-founder, CEO and Chief Scientist at DevOps Research and Assessment (DORA). She is best known for her work measuring the technology process and as the lead investigator on the largest DevOps studies to date.

Mik Kersten is the founder and CEO of Tasktop and drives the strategic direction of the company and a culture of customer-centric innovation. Previously, he launched a series of open source projects that changed how software developers collaborate.

Copyright held by owners/authors.
Publication rights licensed to ACM.

Introducing *ACM Transactions on Human-Robot Interaction*

Now accepting submissions to ACM THRI

As of January 2018, the *Journal of Human-Robot Interaction* (JHRI) has become an ACM publication and has been rebranded as the *ACM Transactions on Human-Robot Interaction* (THRI).

Founded in 2012, the *Journal of HRI* has been serving as the premier peer-reviewed interdisciplinary journal in the field.

Since that time, the human-robot interaction field has experienced substantial growth. Research findings at the intersection of robotics, human-computer interaction, artificial intelligence, haptics, and natural language processing have been responsible for important discoveries and breakthrough technologies across many industries.

THRI now joins the ACM portfolio of highly respected journals. It will continue to be open access, fostering the widest possible readership of HRI research and information. All issues will be available in the ACM Digital Library.

Co-Editors-in-Chief Odest Chadwicke Jenkins of the University of Michigan and Selma Šabanović of Indiana University plan to expand the scope of the publication, adding a new section on mechanical HRI to the existing sections on computational, social/behavioral, and design-related scholarship in HRI.

The inaugural issue of the rebranded *ACM Transactions on Human-Robot Interaction* is planned for March 2018.

To submit, go to <https://mc.manuscriptcentral.com/thri>



DOI:10.1145/3117800

Smart Internet-based infrastructure is one thing but will be ignored without the public's continuing engagement.

BY MILA GASCÓ-HERNANDEZ

Building a Smart City: Lessons from Barcelona

OVER THE PAST few decades, the challenges faced by local governments, like urban growth and migration, have become increasingly complex and interrelated. In addition to traditional land-use regulation, urban maintenance, production and management of services, governments must meet new demands from different actors regarding water supply, natural-resources sustainability, education, safety, and transportation.^{2,16} Moreover, cities today compete with one another for companies, tourists, and especially human talent¹⁸ while addressing unprecedented socioeconomic crises. Innovation, particularly technological innovation, can help local governments address the challenges of contemporary urban governance, improve the urban environment, increase their competitive edge, and cope with environmental



» key insights

- “Smart cities” is an umbrella term for how information and communication technology can help improve the efficiency of a city’s operations and its citizens’ quality of life while also promoting the local economy.
- Smart-city branding often produces better results in terms of external identity and image than implementation of specific smart-city initiatives alone.
- The active participation of city stakeholders and residents in smart-city development is a key success factor.



IMAGE BY ELOI ONELIA

risks. To prevent and manage them, cities must innovate and become smart.

Although current research regarding cities is rich in references to the smart city, it is also fragmented, as smart city is still a fuzzy term that is not used consistently, even by experts.¹³ This fragmentation is also reproduced in terms of the strategies that different cities follow to become smarter. There is no single route to being smart, and different cities have adopted different approaches that

reflect their own very local circumstances.

Barcelona, Spain, is viewed as being among the top most advanced smart cities in the world, according to several recent surveys and is thus often considered a model for other cities to follow. Exactly what makes Barcelona smart is a topic worth exploring and could help guide other cities in their own development processes. Barcelona is particularly interesting because it has reinvented itself over the past 30

years. Following an era of traditional manufacturing (mainly textile) and commerce, its economy was near collapse in the 1980s, with stagnation and widespread unemployment. The challenge for Barcelona's governmental leaders was to transform the economy and social profile, moving to a new economy based in knowledge industries, modern-city tourism, and quality infrastructure for residents, investors, and visitors alike. Technology has been an essential tool supporting the multi-

faceted innovation process at different times, facilitating an evolution from a 2.0 model^a based on e-government initiatives aimed at taking government to citizens through more flexible, straightforward, efficient service, to a 5.0 model, aiming to make the city more inclusive, productive, self-sufficient, innovative, and community-oriented.^{6,9}

Here, I assess Barcelona’s smart-city strategy from 2011 to 2014 when Mayor Xavier Trias of the Democratic Convergence of Catalonia Party, a liberal, regionalist Catalanian party, was elected, took office, and promoted a political strategy of government-driven innovation based on technology to tackle the city’s socioeconomic challenges.

Characterizing Smart Cities

Despite academic attempts to define and conceptually describe a smart city,^{1,8} there is thus far no universally accepted definition. However, several articles and reports have identified certain urban attributes that can help give us an idea.

For example, in 2007, Giffinger et al.¹⁰ ranked 70 European cities on six dimensions: smart economy (competitiveness); smart people (human and social capital); smart governance (participation); smart mobility (transport and ICT); smart environment (natural resources); and smart living (quality of life). As a result, they defined a smart city as “a city well performing in a forward-looking way in these six characteristics, built on the ‘smart’ combination of endowments and activities of

self-decisive, independent and aware citizens.” Likewise, in 2012, Cohen⁵ said smart cities could be understood and evaluated through a different set of six dimensions—environment, mobility, government, economy, society, and quality of life—to account for several working areas measured by one or more quantitative indicators.

Taking a simpler view, Nam and Pardo¹⁵ identified three conceptual dimensions of a smart city—technology (the key to transforming life and work in a city), people (human capital and education), and community (or support of government and policy)—concluding, “A city is smart when investments in human/social capital and IT infrastructure fuel sustainable growth and enhance a quality of life, through participatory governance.”

In 2014 and 2015, respectively, the IESE Cities in Motion project in Spain^{11,12} launched a benchmarking effort focusing on smart cities, producing a more complex model because it included 11 dimensions: human capital, social cohesion, economy, public management, governance, mobility, transportation, environment, urban planning, international outreach, and technology. Each one includes multiple different indicators.

Meanwhile, in 2012, Chourabi et al.⁴ presented one of the most comprehensive and integrative frameworks for analyzing smart-city progress, characterizing smart cities based on eight dimensions, both internal and external, affecting design, implementation, and use of smart-city initiatives (see Table 1). It is now used by the Smart Cities Smart Government Research Practice

Consortium at the Center for Technology in Government (at the University at Albany–SUNY; <https://www.ctg.albany.edu/projects/smartcitiesconsortium>) to assess “smartness” in cities worldwide, including Medellin in Colombia, Seattle and Philadelphia in the U.S., and Milan in Italy. I thus consider it to be a valuable tool for analyzing Barcelona’s own smart-city strategy.

Barcelona’s Smartness

Although technology has always been at the core of the Barcelona City Council’s modernization processes, a notable effort has sought to evolve from an e-government focus to a smart-city focus, gaining momentum after 2011, a year involving a change of the city’s government.

The new government proclaimed its desire to reinforce Barcelona’s smart-city brand as a promoter of a new economy of urban services. The goal was to promote Barcelona as an essential reference for all cities seeking to redirect their economies and external views of themselves following this paradigm.⁹ The Smart City Expo and World Congress, held for the first time in 2011, helped launch and promote this policy.

During the first two years under Mayor Trias, the Barcelona City Council had begun planning new projects, in addition to finishing ones that had already begun (such as the Smart City Campus at 22@ and development of the City Protocol^b). Different projects, with links to one another that were not spelled out explicitly, were individu-

a There was never a Barcelona 1.0 as such.

b <http://www.22barcelona.com/index.php?lang=en> and <http://cityprotocol.org/>

Table 1. Smart-city integrative framework.

Dimension	Description
Management and organization	A project is influenced by such managerial and organizational factors as project size, managers’ attitudes and behaviors, and organizational diversity.
Technology	A smart city relies on computing technologies applied to critical infrastructure components and services, but technology can either improve citizens’ quality of life or contribute to the digital divide.
Governance	Included are processes, norms, and practices that guide the exchange of information among the various stakeholders and their leadership, collaboration, communication, data exchange, partnership, and service integration.
Policy context	Included are the political and institutional components of the environment.
People and communities	Individuals and communities affecting and affected by implementation of a smart-city initiative can involve participation and partnership, accessibility, quality of life, and education.
Economy	Economic inputs to and economic outcomes from smart-city initiatives include innovation, productivity, and flexibility.
Built infrastructure	Availability and quality of technology infrastructure involve wireless infrastructure and service-oriented information systems.
Natural environment	Included are sustainability and good management of natural resources.

ally implemented and together made Barcelona smart. However, in 2013, the City Council recognized the importance of having a comprehensive yet explicit smart-city strategy and declared its willingness to become the first truly smart city in Spain. The City Council thus established this definition of a smart city: “a self-sufficient city of productive neighborhoods at human speed, inside a hyper-connected zero emissions metropolitan area.” Technology and built infrastructure, economy, people and communities, and natural environment were key components in this characterization.

Barcelona’s aim was twofold: use new technologies to foster economic growth and improve the well-being of its citizens. The strategy to achieve it included international positioning, international cooperation, and 22 smart local programs implemented primarily by public-private partnerships (see Table 2). The unit in charge of realizing it was Urban Habitat, which is responsible for the maintenance of the city and improving the urban landscape, including urban transformation and regeneration.

In terms of general results, Ferrer⁶ reported €85 million of added GDP impact in 2014, as well as 21,600 jobs, of which 1,870 were the direct result of smart-city programs. The City Council invested €53.7 million in smart projects in 2014; in return, for each invested euro from the municipal budget, an additional €0.53 euros were invested by third parties, including private businesses. As reported by Ferrer,⁶ the projects added €43 million to the city’s economic activity between 2011 and 2014. The smart-city projects were also expected at the time to save 9,700 tons of CO₂ and 600,000 liters of water consumption annually.⁶ Galvadà and Ribera⁹ were more skeptical of such projections, arguing that most of the initiatives did not make a clear contribution to environmental sustainability and lacked bottom-up approaches involving people and communities. They also argued that specific projects aimed at making the economy more dynamic had a socially negative impact, because they favored the concentration of talent and an influx of new types of residents in certain districts while displacing people already there, mostly of



Barcelona’s aim was twofold: use new technologies to foster economic growth and improve the well-being of its citizens.



low-middle socioeconomic status who were vulnerable to increased costs for housing.

Barcelona is still viewed worldwide as a leading smart city, with several studies ranking it among the smartest in Spain, Europe, and internationally. Additionally, in March 2014, the European Commission awarded it the European Capital of Innovation, or “iCapital,” prize for introducing new technologies to stay better connected to citizens (<http://ec.europa.eu/research/prizes/icapital/index.cfm?pg=2014>). This recognition helped make Barcelona the “Mobile World Capital” through 2023. Meanwhile, the United Nations established its international office for urban resilience in Barcelona and, along with IESE Business School, the Center of Excellence for PPP [public-private partnerships] in Smart Cities. And the World Bank identified Barcelona as a knowledge hub for exploring the use of ICT in city management.

Analysis. Table 2 outlines the city’s comprehensive strategy, including programs and projects aimed at developing the eight dimensions of the Chourabi et al. framework.⁴ However, many specific projects focused mainly on technology and built infrastructure, which was consistent with the city’s previous expertise in the use of technology. I analyzed the program’s specific contributions to the various dimensions by conducting 19 semistructured interviews with Barcelona city officials and stakeholders, including the former deputy mayor for urban development of the Barcelona City Council, former e-government and smart cities directors of the Barcelona City Council, managing director of 22@, and several academic experts.

Regarding management and organizational issues, Barcelona’s management and organizational structure were part of a broader management model based on new public management and thus on territorial decentralization, service externalization, and adoption of managerial tools (such as strategic plans), an approach with technology at its core. Castells and Ollé³ wrote that for a long time, Barcelona’s governance paradigm focused on deployment of the Internet to help increase internal efficiency in local government, improving public servic-

Table 2. Barcelona's 22 smart local programs following Ferrer⁶ and Gavaldà and Ribera.⁹

Program	Year launched	Main dimensions addressed ⁵
Telecommunications and networks	2013	Built infrastructure
Urban platform	2013	Technology
Smart data	2013	Technology
Smart lighting	2013	Technology, built infrastructure, natural environment
Energy self-sufficiency	2013	Technology, built infrastructure, natural environment
Smart water	2013	Technology, built infrastructure, natural environment
Smart mobility	2013 but most in 2014	Technology, built infrastructure, natural environment
Urban transformation	2013	Built infrastructure
Smart urban furniture	2014	Built infrastructure
Urban resilience	2013	Governance, built infrastructure, natural environment
Smart citizens	2013	People and communities
Open government	2013	Management and organization, technology, people and communities
Barcelona In Your Pocket	2013	Management, organization, technology, people and communities
Smart garbage collection	2013	Technology, built infrastructure, natural environment
Smart regulation	2014	Policy context
Smart innovation	2013	Economy
Health and social services	2013	Technology, governance, people and communities
Education	2013	Technology, governance, people and communities
Smart tourist destination	2013	Technology, governance, people and communities, economy
Infrastructure and logistics	2013	Economy
Leisure and culture	Not begun	Technology, governance, people and communities
Security	2014	Technology, governance, people and communities

a Results are available only for very specific projects reported up to 2014.

es, and rearticulating city governance processes. Meanwhile, Mayor Trias actively supported development of the smart city. So did the Urban Habitat department, politically and technically. The Computer Municipal Institute also helped implement the strategy, adapting to its evolution, from siloed to citywide effort.

Technology was, and still is, at the core of the Barcelona urban-development model and essential crosswise tool supporting the innovation process. Unlike many other cities, Barcelona's key smart projects reflected and continue to reflect the strategic use of technology in development of a smart city.⁸

Regarding governance, Barcelona involved several local and regional stakeholders in the definition and implementation of its smart-city strategy, particularly businesses and universities. Indeed, public-private partnerships and collaboration with other public administrations proved highly effective in the implementation of the various local programs, though each such program has been managed differently. In the case of Barcelona, a lot of support/cooperation thus came from the Autonomous Government of Catalonia, particularly in relation to wider projects that included other areas in Catalonia, in addition to Barcelona.

Cooperation with other cities in Spain, along with European Union support, was also important.

However, this involvement essentially took place by adopting a top-down perspective, with the Barcelona City Council leading efforts in the city: though other actors participated, the City Council provided explicit direction regarding strategy, programs, and projects in the smart-city effort.

In terms of policy context, the smart-city strategy clearly expanded while always keeping in mind the need to prioritize urban-transformational projects. However, the strategy's political and institutional components proved fragile

Strategic projects	Results ^a
New telecommunication networks, Antennas Plan, WiFi	New telecommunication Networks: >500 km of optical fibre
Sensor platform, CityOS, iCity, CityDB	NA
City Key Performance Indicators, Situation Room	NA
Master lighting plan, sensors	Master lighting plan: 1,155 urban lights with LED bulbs
Self-sufficient islands, smart grid, cooling and heating network, building protocol, smart-meter electricity distribution, corporate buildings	Cooling and heating network: 29km connecting >60 buildings
Remote irrigation, smart sewer system, water-table management, remote ornamental fountains, pilot water telemetry metering	NA
Zero emissions mobility, vehicle guideway system, public-space utilization, "sensorization," and identification of new services (such as Les Corts pilot), mobility plan, orthogonal bus network, smart parking	Orthogonal bus network: 17 vertical routes, eight horizontal routes, three diagonal routes Sensorization and identification of new services: 50 sensors installed Mobility plan: >500 hybrid taxis, >130 electric bikes to rent
Paseo de Gracia, Parallel Avenue, Paseo de San Juan, La Sagrera, New Museum Center of Montjuic	NA
Smartquesina, kiosks	Kiosks: 44 available
United Nations program, infrastructure table, urban services	NA
Fablabs, Citizen Sensors, Whabit	Fablabs: two fablabs open each year involving 5,400 citizens and 200 institutions
Open data, citizen virtual office, e-administration	Open data: 322 available datasets, 80% with >3 quality rating on Tim Berners-Lee's scale Citizen virtual office: >40% virtual procedures in one year, 34% users >50 years (no digital divide)
Barcelona Contactless, Digital Identity, Apps4BCN, Mobile Ecosystem	Barcelona Contactless: 8,000 pieces of urban furniture connected, with >15 actors public and private actors Apps4BCN: >650 apps (88.5% developed privately), >36% of apps downloaded, eight challenges, >350 developers Mobile ecosystem: >12,000 companies, >86,000 workers, €18 billion revenue
Optimized waste collection, green mobile point	NA
Tenders, legislation	NA
Smart City Campus, Smart City Tour, Smart City Cluster, Urban Lab, Competence Center mSmart City, Spark Lab, BIT Habitat	Urban Lab: 43 pilot proposals, 18 pilots
Catalan Health Plan, Strategic Plan SITIC, iSalut.cat, Vincles	Bloomberg Philanthropy Prize winner, 20,000 assistant networks, 100,000 users, 110,000 potential users
Educat, mSchools, Raspberry BCN, 4DLife, Smart Hort (vegetable garden)	NA
Geographical information system of tourism, Catalonia Experience Program	NA
Industrial sector	NA
—	NA
—	NA

after a new government took office in May 2015, replacing Mayor Trias. The new officials, including the new mayor and her appointees, offered no enthusiasm for or agreement with the smart-city strategy, resulting in cancellation of most of its projects and development of another vision for the city (called Barcelona Digital City <http://ajuntament.barcelona.cat/digital/en>).

Even under Mayor Trias, Barcelona had rarely implemented projects that required the participation of individuals and groups or communities. Generally speaking, there was a lack of bottom-up approaches. The failed electronic consultation on the 2010

transformation of Diagonal Avenue, the city's main street, forced the City Council to be cautious regarding any citizen participation. Additionally, citizens were not aware what it means to develop a smart city, with many not knowing what a smart city even is, thus further inhibiting their participation.

Barcelona has a dynamic economy. The city was awarded the European Capital of Innovation (iCapital designation) in March 2014 for promoting a broad-based innovation culture. Other economic factors also made it particularly competitive, including an entrepreneurial culture and promotion of technology-based economic

activities in specific districts. The economic boost Barcelona experienced followed the same pattern as its smart strategy, following both the City Council and the Autonomous Government of Catalonia.

Over the past 30 years, Barcelona has prioritized development of technological infrastructure. In particular, following the Barcelona 2.0 model framework and local elections in 2007, it implemented several projects, including Barcelona WiFi (a service allowing citizens to connect to the Internet through WiFi access points) and the WiFi mesh network (a municipal network for ubiquitous services). In 2011, it further



Torre Agbar (Agbar Tower) in Barcelona is covered with more than 4,500 luminous devices capable of creating 16 million colors.

invested in physical and technological infrastructure. The projects under the 22@ umbrella regarding urban transformation and urban innovation motivated development of a modern network of energy, telecommunications, district heating, and pneumatic garbage-collection systems.

Regarding the natural environment, many Barcelona smart-city projects were designed to help deliver environmental sustainability but achieved no clear goals. Gavalda and Ribera⁹ re-

ported, “The effects of the economic crisis have put in standby the achievement of higher levels of quality of life and have paralyzed investments in environmental sustainability at macro level.” It is a revealing statement for a city aiming to achieve sustainability, including self-sufficiency, eco-efficiency, and zero emissions.

Conclusion

Three main conclusions arise from the analysis I have presented here. First,

Barcelona’s smart-city strategy has evolved over time. Before 2011, but also during the first two years of Mayor Trias’s term, 2011 to 2013, the model the city pursued was intended to make Barcelona a smarter city by providing a local public administration that was simple, effective, closely connected to citizens, ubiquitous, and innovative. Barcelona was getting smart essentially through e-government/e-governance programs and projects, but there was no overall smart-city strategy that embraced these projects, linking them and giving them purpose. However, in 2013, such links and direction did start to attract public attention and support.


Second, two years of financial and political investment was not enough by itself to make the city smart. Although several projects were implemented in 2013 and 2014, by the end of 2014, there were still no clear results. Moreover, most smart-city-related accomplishments were reported in terms of output (such as number of things done and number of users) rather than outcomes (such as economic growth or environmental improvement). Also, although Barcelona’s approach to urban transformation followed a long-term vision, the change of government in May 2015 when Mayor Trias was voted out of office showed his vision was not shared, hindering its sustainability and the possibility to earn a greater return on the financial and political investment already made.

Even so, Barcelona’s reputation as a smart city remains positive. Indeed, Barcelona is viewed by urban planners and scholars, as well as progressive politicians worldwide, as a leading smart city to which many other cities turn for inspiration and real-world guidance. Barcelona invested financially and politically to build a smart city, with mixed results but successfully established a brand—the Barcelona Smart City—by following a city-management model that had already proved successful.

Finally, following Mintzberg,¹⁴ Barcelona’s smart-city strategy was driven more by deliberate components than by emergent components, or learning and interacting with stakeholders. It was the City Council that conceptualized the smart-city strategy and implemented it, involving other actors,


mainly businesses and research centers and universities, to pursue what it thought was best. However, ordinary citizens did not have a say nor did they understand the smart-city concept. The city's aim was to become smarter for the benefit of its citizens but without including or educating them.

What lessons can Barcelona teach other public managers and politicians? My research found that management and organization, governance, people, and communities are especially important dimensions when planning a smart-city initiative. On one hand, Barcelona's experience shows that real projects matter but that public image and marketing are also important. Barcelona's smart-city marketing strategy has indeed proved even more successful than the city's "real" strategy. Clear vision and strategy are also crucial. On the other hand, a smart city's sustainability depends on sharing the vision and goals with key stakeholders who might end up being responsible for continuing work begun by others. A smart city is not just another technological project but long-term urban transformation defining the type of city the public wants to live in, a decision that cannot be changed with each new election. Finally, no smart city can involve its citizens only as recipients of its interventions but include them as partners deciding the type of city they want to live in when designing, implementing, and evaluating related projects.


No matter how advanced a city is technologically, what drives smartness is the capacity of public organizations and public servants to plan, implement, and assess a strategy, as well as engage citizens and other stakeholders in its development. From a practical point of view, this entails developing a city model co-conceptualized and co-implemented with ordinary citizens and other stakeholders. 

References

1. AlAwadhi, S. and Scholl, H.J. Aspirations and realizations: The Smart City of Seattle. In *Proceedings of the 46th Hawaii International Conference on System Sciences* (Maui, HI, Jan. 7–10). IEEE Computer Society Press 2013, 1695–1703.
2. Albrechts, L. Shifts in strategic spatial planning? Some evidence from Europe and Australia. *Environment and Planning A* 38, 6 (2006), 1149–1170.
3. Castells, M. and Ollé, E. *El Model Barcelona II: L'Ajuntament de Barcelona a la Societat Xarxa*.



No smart city can involve its citizens only as recipients of its interventions but include them as partners deciding the type of city they want to live in when designing, implementing, and evaluating related projects.



Research Report. Universitat Oberta de Catalunya, Barcelona, Spain, 2004; http://www.uoc.edu/in3/pic/cat/pdf/PIC_Ajuntament_0_0.pdf

4. Chourabi, H., Nam, T., Walker, S., Gil-García, J.R., Mellouli, S., Nahon, K., Pardo, T., and Scholl, H.J. Understanding smart cities: An integrative framework. In *Proceedings of the 45th Hawaii International Conference on System Sciences* (Maui, HI, Jan. 4–7) IEEE Computer Society Press, 2012, 2289–2297.
5. Cohen, B. What exactly is a smart city? *Fast Company* (Sept. 9, 2012); <https://www.fastcodesign.com/1680538/what-exactly-is-a-smart-city>
6. Ferrer, J.R. *Barcelona 5.0. A Roman Village Transforming into a Smart City*. PowerPoint presentation. Smart Cities: Innovating in City Management (Apr. 15, 2015).
7. Gascó, M. Smart cities: Un fenómeno en auge. *ESADE Alumni Magazine* (Feb.-Mar. 2015), 20–22.
8. Gascó, M., Trivellato, B., and Cavenago, D. How do Southern European cities foster innovation? Lessons from the experience of the smart city approaches of Barcelona and Milan. Chapter in *Smarter As the New Urban Agenda: A Comprehensive View of the 21st Century City*, J.R. Gil-García, T. Pardo, and T. Nam, Eds. Springer, New York, 2015, 191–216.
9. Gavalda, J. and Ribera, R. *Barcelona 5.0: From Knowledge to Smartness?* Working Paper Series, WP12-002. Universitat Oberta de Catalunya, Barcelona, Spain 2012; <http://in3-working-paper-series.uoc.edu/in3/en/index.php/in3-working-paper-series/article/download/1590/1590-4557-1-PB.pdf>
10. Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanovic, N., and Meijers, E. *Smart Cities: Ranking of European Medium-Sized Cities*. Research Report. Center of Regional Science, Vienna UT, Vienna, Austria, 2007; http://www.smart-cities.eu/download/smart_cities_final_report.pdf
11. IESE Cities in Motion. Cities in motion – Index 2015. IESE Business School, Barcelona, Spain 2015; <http://www.iese.edu/research/pdfs/ST-0366-E.pdf>
12. IESE Cities in Motion. Cities in motion – Index 2014. IESE Business School, Barcelona, Spain, 2014; <http://www.iese.edu/research/pdfs/ST-0396-E.pdf>
13. Meijer, A. and Rodriguez-Bolivar, P. Governing the smart city: A review of the literature on smart urban governance. *International Review of Administrative Science* 82, 2 (2016), 392–408.
14. Mintzberg H. *The Rise and Fall of Strategic Planning*. Free Press, New York, 1994.
15. Nam, T. and Pardo, T. Smart city as urban innovation: Focusing on management, policy, and context. In *Proceedings of the Fifth International Conference on Theory and Practice of Electronic Governance* (Tallinn, Estonia, Sept. 26–28). ACM Press, New York, 2011, 185–194.
16. Naphade, M., Banavar, G., Harrison, C., Paraszczak, J., and Morris, R. Smarter cities and their innovation challenges. *Computer* 44, 6 (June 2011), 32–39.
17. Yin, R. Case study research. Chapter in *Design and Methods, Fourth Edition*. SAGE Publications, Thousand Oaks, CA, 2009.
18. Zenker, S., Eggers, F., and Farsky, M. Putting a price tag on cities: Insights into the competitive environment of places. *Cities* 30 (Feb. 2013), 133–139.

Mila Gascó-Hernandez (mgasco@ctg.albany.edu) is the Associate Research Director of the Center for Technology in Government and a research associate professor in the Department of Public Administration and Policy at Rockefeller College at the University at Albany, Albany, NY, USA.

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



Watch the author discuss her work in this exclusive *Communications* video. <https://cacm.acm.org/videos/building-a-smart-city>

DOI:10.1145/3188720

For a static analysis project to succeed, developers must feel they benefit from and enjoy using it.

BY CAITLIN SADOWSKI, EDWARD AFTANDILIAN, ALEX EAGLE, LIAM MILLER-CUSHON, AND CIERA JASPAN

Lessons from Building Static Analysis Tools at Google

SOFTWARE BUGS COST developers and software companies a great deal of time and money. For example, in 2014, a bug in a widely used SSL implementation (“goto fail”) caused it to accept invalid SSL certificates,³⁶ and a bug related to date formatting caused a large-scale Twitter outage.²³ Such bugs are often statically detectable and are, in fact, obvious upon reading the code or documentation yet still make it into production software.

Previous work has reported on experience applying bug-detection tools to production software.^{6,3,7,29} Although there are many such success stories for developers using static analysis tools, there are also reasons engineers do not always use static analysis tools or ignore their warnings,^{6,7,26,30} including:

Not integrated. The tool is not integrated into the developer’s workflow or takes too long to run;

Not actionable. The warnings are not actionable;

Not trustworthy. Users do not trust the results due to, say, false positives;

Not manifest in practice. The reported bug is theoretically possible, but the problem does not actually manifest in practice;

» key insights

- **Static analysis authors should focus on the developer and listen to their feedback.**
- **Careful developer workflow integration is key for static analysis tool adoption.**
- **Static analysis tools can scale by crowdsourcing analysis development.**

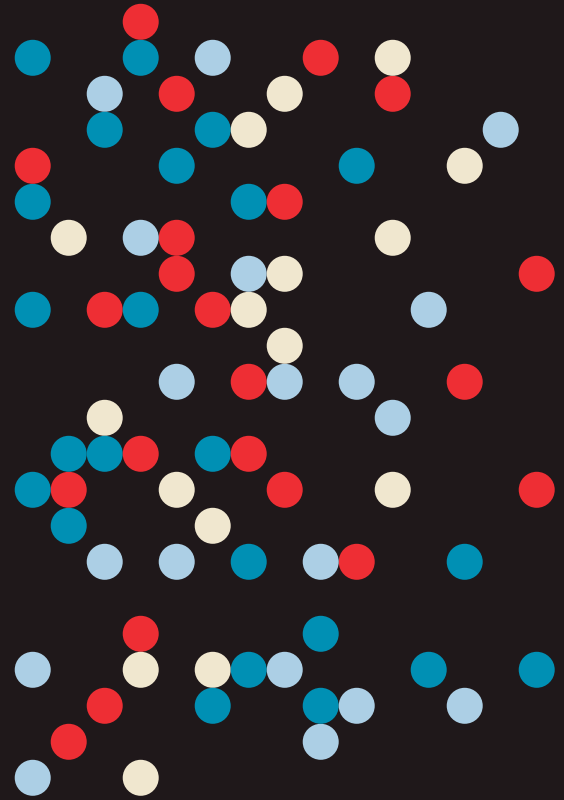
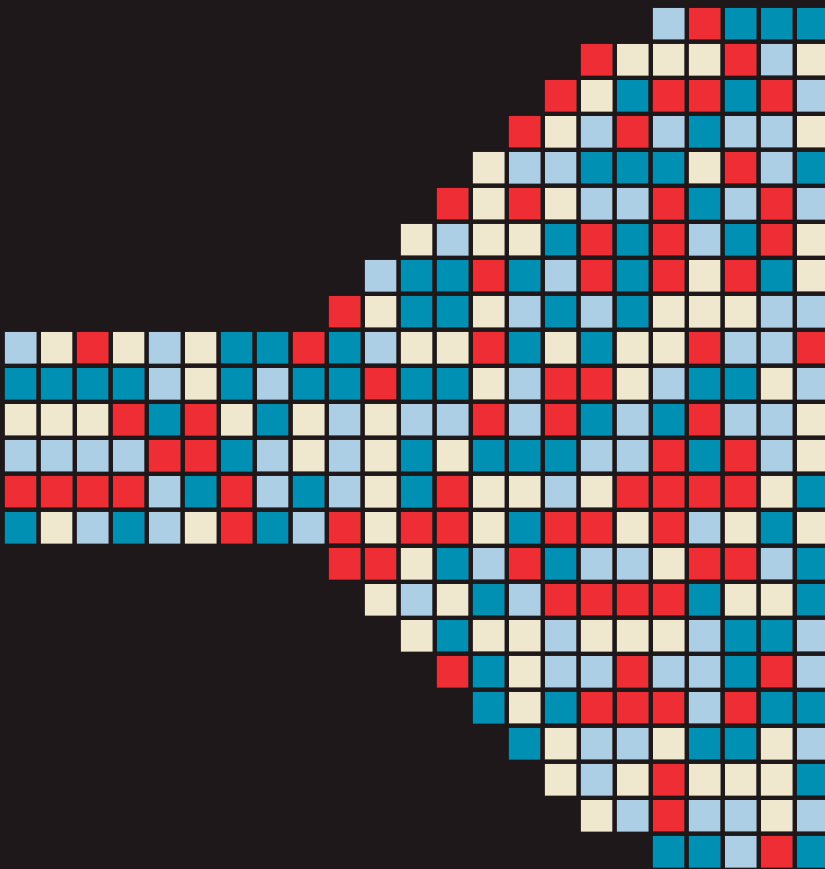


IMAGE BY IGOR KISSELEV

Too expensive to fix. Fixing the detected bug is too expensive or risky; and *Warnings not understood.* Users do not understand the warnings.

Here, we describe how we have applied the lessons from Google's previous experience with FindBugs Java analysis, as well as from the academic literature, to build a successful static analysis infrastructure used daily by most software engineers at Google. Google's tooling detects thousands of problems per day that are fixed by engineers, by their own choice, before the problematic code is checked into Google's companywide codebase.

Scope. We focus on static analysis tools that have become part of the core developer workflow at Google and used by a large fraction of Google's develop-

ers. Many of the static analysis tools deployed at the scale of Google's two-billion-line codebase³² are relatively simple; running more sophisticated analyses at scale is not yet considered a priority.

Note that developers outside of Google working in specialized fields (such as aerospace¹³ and medical devices²¹) may use additional static analysis tools and workflows. Likewise, developers working on specific types of projects (such as kernel code and device drivers⁴) may run ad hoc analyses. There has been lots of great work on static analysis, and we do not claim the lessons we report here are unique, but we do believe that collating and sharing what has worked to improve code quality and the devel-

oper experience at Google is valuable.

Terminology. We use the following terms: analysis tools run one or more "checks" over source code and identify "issues" that may or may not represent actual software faults. We consider an issue to be an "effective false positive" if developers did not take positive action after seeing the issue.³⁵ If an analysis incorrectly reports an issue, but developers make the fix anyway to improve code readability or maintainability, that is not an effective false positive. If an analysis reports an actual fault, but the developer did not understand the fault and therefore took no action, that is an effective false positive. We make this distinction to emphasize the importance of developer perception. Developers, not tool authors, will determine and act on a tool's perceived

false-positive rate.

How Google builds software. Here, we outline key aspects of Google's software-development process. At Google, nearly all developer tools (with the exception of the development environment) are centralized and standardized. Many parts of the infrastructure are built from scratch and owned by internal teams, giving the flexibility to experiment.

Source control and code ownership. Google has developed and uses a single-source control system and a single monolithic source code repository that holds (nearly) all Google proprietary source code.^a Developers use "trunk-based" development, with limited use of branches, typically for releases, not for features. Any engineer can change any piece of code, subject to approval by the code's owners. Code ownership is path-based; an owner of a directory implicitly owns all subdirectories as well.

Build system. All code in Google's repository builds with a customized version of the Bazel build system,⁵ requiring that builds be hermetic; that is, all inputs must be explicitly declared and stored in source control so the builds are easily distributed and parallelized. In Google's build system, Java rules depend on the Java Development Kit and Java compiler that are checked into source control, and such binaries can be updated for all users simply by checking-in new versions. Builds are generally from source (at head), with few binary artifacts checked into the repository. Since all developers use the same build system, it is the source of truth for whether any given piece of code compiles without errors.

Analysis tools. The static analysis tools Google uses are typically not complex. Google does not have infrastructure support to run interprocedural or whole-program analysis at Google scale, nor does it use advanced static analysis techniques (such as separation logic⁷) at scale. Even simple checks have required analysis infrastructure supporting workflow integration to make them successful. The types of analyses deployed as part of the general developer workflow include:

Style checkers (such as Checkstyle,¹⁰

a Google's large open source projects (such as Android and Chrome) use separate infrastructure and their own workflows.

Developers, not tool authors, will determine and act on a tool's perceived false-positive rate.

Pylint,³⁴ and Golint¹⁸);

Bug-finding tools that may extend the compiler (such as Error Prone,¹⁵ ClangTidy,¹² Clang Thread Safety Analysis,¹¹ Govet,¹⁷ and the Checker Framework⁹), including, but not limited to, abstract-syntax-tree pattern-match tools, type-based checks, and unused variable analysis;

Analyzers that make calls to production services (such as to check whether an employee mentioned in a code comment is still employed at Google); and

Analyzers that examine properties of build outputs (such as the size of binaries).

The "goto fail" bug³⁶ would have been caught by Google's C++ linter that checks whether `if` statements are followed by braces. The code that caused the Twitter outage²³ would not compile at Google because of an Error Prone compiler error, a pattern-based check that identifies date-formatting misuses. Google developers also use dynamic analysis tools (such as AddressSanitizer) to find buffer overruns and ThreadSanitizer to find data races.¹⁴ These tools are run during testing and sometimes also with production traffic.

Integrated Development Environments (IDEs). An obvious workflow integration point to show static analysis issues early in the development process is within an IDE. However, Google developers use a wide variety of editors, making it difficult to consistently detect bugs by all developers prior to invoking the build tool. Although Google does use analyses integrated with popular internal IDEs, requiring a particular IDE with analyses enabled is a non-starter.

Testing. Nearly all Google code includes corresponding tests, ranging from unit tests all the way to large-scale integration tests. Tests are integrated as a first-class concept in the build system and hermetic and distributed, just like builds. For most projects, developers write and maintain the tests for their code; projects typically have no separate testing or quality-assurance group. Google's continuous build-and-test system runs tests on every commit and notifies a developer if the developer's change broke the build or caused a test to fail. It also supports testing a change before committing to avoid

breaking downstream projects.

Code review. Every commit to Google's codebase goes through code review first. Although any developer can propose a change to any part of Google's code, an owner of the code must review and approve the change before submission. In addition, even owners must have their code reviewed before committing a change. Code review happens through a centralized, web-based tool that is tightly integrated with other development infrastructure. Static analysis results are surfaced in code review.

Releasing code. Google teams release frequently, with much of the release validation and deployment process automated through a "push on green" methodology,²⁷ meaning an arduous, manual-release-validation process is not possible. If Google engineers find a bug in a production service, a new release can be cut and deployed to production servers at relatively low cost compared with applications that must be shipped to users.

What We Learned from FindBugs

Earlier research, from 2008 to 2010, on static analysis at Google focused on Java analysis with FindBugs^{2,3}: a stand-alone tool created by William Pugh of the University of Maryland and David Hovemeyer of York College of Pennsylvania that analyzes compiled Java class files and identifies patterns of code that lead to bugs. As of January 2018, FindBugs was available at Google only as a command-line tool used by few engineers. A small Google team, called "BugBot," worked with Pugh on three failed attempts to integrate FindBugs into the Google developer workflow.

We have thus learned several lessons:

Attempt 1. Bug dashboard. Initially, in 2006, FindBugs was integrated as a centralized tool that ran nightly over the entire Google codebase, producing a database of findings engineers could examine through a dashboard. Although FindBugs found hundreds of bugs in Google's Java codebase, the dashboard saw little use because a bug dashboard was outside the developers' usual workflow, and distinguishing between new and existing static-analysis issues was distracting.

Attempt 2. Filing bugs. The BugBot team then began to manually triage new issues found by each nightly Find-

Bugs run, filing bug reports for the most important ones. In May 2009, hundreds of Google engineers participated in a companywide "Fixit" week, focusing on addressing FindBugs warnings.³ They reviewed a total of 3,954 such warnings (42% of 9,473 total), but only 16% (640) were actually fixed, despite the fact that 44% of reviewed issues (1,746) resulted in a bug report being filed. Although the Fixit validated that many issues found by FindBugs were actual bugs, a significant fraction were not important enough to fix in practice. Manually triaging issues and filing bug reports is not sustainable at a large scale.

Attempt 3. Code review integration. The BugBot team then implemented a system in which FindBugs automatically ran when a proposed change was sent for review, posting results as comments on the code-review thread, something the code-review team was already doing for style/formatting issues. Google developers could suppress false positives and apply FindBugs' confidence in the result to filter comments. The tooling further attempted to show only new FindBugs warnings but sometimes miscategorized issues as new. Such integration was discontinued when the code-review tool was replaced in 2011 for two main reasons: the presence of effective false positives caused developers to lose confidence in the tool, and developer customization resulted in an inconsistent view of analysis results.

Make It a Compiler Workflow

Concurrent with FindBugs experimentation, the C++ workflow at Google was improving with the addition of new checks to the Clang compiler. The Clang team implemented new compiler checks, along with suggested fixes, then used ClangMR³⁸ to run the updated compiler in a distributed way over the entire Google codebase, refine checks, and programmatically fix all existing instances of a problem in the codebase. Once the codebase was cleansed of an issue, the Clang team enabled the new diagnostic as a compiler error (not a warning, which the Clang team found Google developers ignored) to break the build, a report difficult to disregard. The Clang team was very successful improving the co-

debase through this strategy.

We followed this design and built a simple pattern-based static analysis for Java called Error Prone¹⁵ on top of the javac Java compiler.¹ The first check rolled out, called `PreconditionsCheckNotNull`,^b detects cases in which a runtime precondition check trivially succeeds because the arguments in the method call are transposed, as when, say, `checkNotNull("uid was null", uid)` instead of `checkNotNull(uid, "uid was null")`.

In order to launch checks like `PreconditionsCheckNotNull` without breaking any continuous builds, the Error Prone team runs such checks over the whole codebase using a javac-based MapReduce program, analogous to ClangMR, called JavacFlume built using FlumeJava.⁸ JavacFlume emits a collection of suggested fixes, represented as diffs, that are then applied to produce a whole-codebase change. The Error Prone team uses an internal tool, Rosie,³² to split the large-scale change into small changes that each affect a single project, test those changes, and send them for code review to the appropriate team. The team reviews only those fixes that apply to its code, and, when they approve them, Rosie commits the change. All changes are eventually approved, the existing issues are fixed, and the team enables the compiler error.

When we have surveyed developers who received these patches, 57% of them who received a proposed fix to checked-in code were happy to have received it, and 41% were neutral. Only 2% responded negatively, saying, "It just created busywork for me."

Value of compiler checks. Compiler errors are displayed early in the development process and integrated into the developer workflow. We have found expanding the set of compiler checks to be effective for improving code quality at Google. Because checks in Error Prone are self-contained and written against the javac abstract syntax tree, rather than bytecode (unlike FindBugs), it is relatively easy for developers outside the team to contribute checks. Leveraging these contributions is vital in increasing Error Prone's overall im-

b <http://errorprone.info/bugpattern/PreconditionsCheckNotNull>

pact. As of January 2018, 733 checks had been contributed by 162 authors.

Reporting issues sooner is better. Google's centralized build system logs all builds and build results, so we identified all users who had seen one of the error messages in a given time window. We sent a survey to developers who recently encountered a compiler error and developers who had received a patch with a fix for the same problem. Google developers perceive that issues flagged at compile time (as opposed to patches for checked-in code) catch more important bugs; for example, survey participants deemed 74% of the issues flagged at compile time as "real problems," compared to 21% of those found in checked-in code. In addition, survey participants deemed 6% of the issues found at compiletime (vs. 0% in checked-in code) "critical." This result is explained by the "survivor effect";³ that is, by the time code is submitted, the errors are likely to have been caught by more expensive means (such as testing and code review). Moving as many checks into the compiler as possible is one proven way to avoid those costs.

Criteria for compiler checks. To scale-up our work, we have defined criteria for enabling checks in the compiler, setting the bar high, since breaking the compile would be a significant disruption. A compiler check at Google should be easily understood; actionable and easy to fix (whenever possible, the error should include a suggested fix that can be applied mechanically); produce no effective false positives (the analysis should never stop the build for correct code); and report issues affecting only correctness rather than style or best practices.

The primary goal of an analyzer satisfying these criteria is not simply to detect faults but to automatically fix all instances of a prospective compiler error throughout the codebase. However, such criteria limit the scope of the checks the Error Prone team enables when compiling code; many issues that cannot always be detected correctly or mechanically fixed are still serious problems.

Warn During Code Review

Once the Error Prone team had built the infrastructure needed to detect issues at compile time, and had proved

the approach works, we wanted to show more high-impact bugs that do not meet the criteria we outlined earlier for compiler errors and provide results for languages other than Java and C++. The second integration point for static analysis results is Google's code review tool, Critique; static analysis results are exposed in Critique using Tricorder,³⁵ Google's program-analysis platform. As of January 2018, there was a compiler warnings-free default for C++ and Java builds at Google, with all analysis results either shown as compiler errors or in code review.

Criteria for code-review checks. Unlike compile-time checks, analysis results shown during code review are allowed to include up to 10% effective false positives. There is an expectation during code review that feedback is not always perfect and that authors evaluate proposed changes before applying them. A code review check at Google should fulfill several criteria:

Be understandable. Be easy for any engineer to understand;

Be actionable and easy to fix. The fix may require more time, thought, or effort than a compiler check, and the result should include guidance as to how the issue might indeed be fixed;

Produce less than 10% effective false positives. Developers should feel the check is pointing out an actual issue at least 90% of the time;^c and

Have the potential for significant impact on code quality. The issues may not affect correctness, but developers should take them seriously and deliberately choose to fix them.

Some issues are severe enough to be flagged in the compiler, but producing them or developing an automated fix is not feasible. For example, fixing an issue may require significant restructuring of the code. Enabling these checks as compiler errors would require manual cleanup of existing instances that is infeasible on the scale of Google's vast codebase. Analysis tools show these checks in code review prevent new occurrences of the issue, allowing the developer to decide how to

make an appropriate fix. Code review is also a good context for reporting relatively less-important issues like stylistic problems or opportunities to simplify code. In our experience, reporting them at compile-time is frustrating for developers and makes it more difficult to iterate and debug quickly; for example, an unreachable code detector might hinder attempts to temporarily disable a block of code for debugging. However, at code-review time, developers are preparing their code to be seen; they are already in a critical mindset and more receptive to seeing readability and stylistic details.

Tricorder. Tricorder is designed to be easily extensible and support many different kinds of program-analysis tools, including static and dynamic analyses. We showed a suite of Error Prone checks in Tricorder that cannot be enabled as compiler errors. Error Prone also inspired a new set of analyses for C++ that are integrated with Tricorder and called ClangTidy.¹² Tricorder analyzers report results for more than 30 languages, support simple syntactic analyses like style checkers, leverage compiler information for Java, JavaScript, and C++, and are straightforward to integrate with production data (such as about jobs that are currently running). Tricorder continues to be successful at Google because it is a plug-in model supporting an ecosystem of analysis writers, actionable issues are highlighted during the code-review process, and it provides feedback channels to improve analyzers and ensure analyzer developers act on the feedback.

Empower users to contribute. As of January 2018, Tricorder included 146 analyzers, with 125 contributed from outside the Tricorder team and seven plug-in systems for hundreds of additional checks (such as ErrorProne and ClangTidy, which comprise two of the seven analyzers plug-in systems).

Provide fixes and involve reviewers. Tricorder checks can provide suggested fixes that can be directly applied from the code-review tool. They are seen by both the reviewer and the author, and the reviewer can ask the author to fix the problematic code simply by clicking a "Please fix" button on the analysis result. Reviewers typically withhold approval of a change until

^c Although this number was initially chosen by the first author somewhat arbitrarily, it seems to be a sweet spot for developer satisfaction and matches the cutoff for similar systems in other companies.


all their comments, manual and automated, have been addressed.

Iterate on feedback from users. In addition to the “Please fix” button, Tricorder also provides a “Not useful” button that reviewers or proposers can click to express that they do not like the analysis finding. Clicking automatically files a bug in the issue tracker, routing it to the team that owns the analyzer. The Tricorder team tracks such not-useful clicks, computing the ratio of “Please fix” vs. “Not useful” clicks. If the ratio for an analyzer goes above 10%, the Tricorder team disables the analyzer until the author(s) improve it. While the Tricorder team has rarely had to permanently disable an analyzer, it has disabled an analyzer (on several occasions) while the analyzer author is removing and revising sub-checks that were particularly noisy.


The bugs being filed often lead to improvement in the analyzers that in turn greatly improves developers’ satisfaction with those analyzers; for example, the Error Prone team developed, in 2014, an Error Prone check that flagged when too many arguments were being passed to a `printf`-like function in Guava.¹⁹ The `printf`-like function did not actually accept all `printf` specifiers, accepting only `%s`. About once per week the Error Prone team would receive a “Not useful” bug claiming the analysis was incorrect because the number of format specifiers in the bug filers’ code matched the number of arguments passed. In every case, the analysis was correct, and the user was trying to pass specifiers other than `%s`. The team thus changed the diagnostic text to state directly that the function accepts only the `%s` placeholder and stopped getting bugs filed about that check.

Scale of Tricorder. As of January 2018, Tricorder had analyzed approximately 50,000 code review changes per day. During peak hours, there were three analysis runs per second. Reviewers clicked “Please Fix” more than 5,000 times per day, and authors applied the automated fixes approximately 3,000 times per day. And Tricorder analyzers received “Not useful” clicks 250 times per day.

The success of code-review analysis suggests it occupies a “sweet spot” in the developer workflow at Google.



Even in a mature codebase with full test coverage and a rigorous code-review process, bugs slip by.



Analysis results that are shown at compilation time must reach a much higher bar for quality and accuracy that is not possible to meet for some analyses that can still identify serious faults. After the review and code are checked in, the friction confronting developers for making changes increases. Developers are thus hesitant to make additional changes to code that has already been tested and released, and lower severity and less-important issues are unlikely to be addressed. Other analysis projects among major software-development organizations (such as Facebook Infer analysis for Android/iOS apps⁷) have also highlighted code review as a key point for reporting analysis results.

Expand Analyzer Reach

As Google developer-users have gained trust in the results from Tricorder analyzers, they continue to request further analyses. Tricorder addresses this in two ways: allowing project-level customization and adding analysis results at additional points in the developer workflow. In this section, we also touch on the reasons Google does not yet leverage more sophisticated analysis techniques as part of its core developer workflow.

Project-level customization. Not all requested analyzers are equally valuable throughout the Google codebase; for example, some analyzers are associated with higher false-positive rates and so would have correspondingly high effective false-positive rates or require specific project configuration to be useful. These analyzers all have value but only for the right team.

To satisfy these requests, we aimed to make Tricorder customizable. Our previous experience with customization for FindBugs did not end well; user-specific customization caused discrepancies within and across teams and resulted in declining use of tools. Because each user could see a different view of issues, there was no way to ensure a particular issue was seen by everyone working on a project. If developers removed all unused imports from their team’s code, the fix would quickly backslide if even a single other developer was not consistent about removing unused imports.

To avoid such problems, Tricorder allows configuration only at the proj-

ect level, ensuring that anyone making a change to a particular project sees a consistent view of the analysis results relevant to that project. Maintaining a consistent view has enabled several types of analyzers to do the following:

Produce dichotomous results. For example, Tricorder includes an analyzer for protocol buffer definitions³³ that identifies changes that are not backward compatible. It is used by developer teams that ensure persistent information from protocol buffers in their serialized form but is annoying for teams that do not store data in this form. Another example is an analyzer that suggests using Guava³⁷ or Java 7 idioms that do not make sense for projects that cannot use these libraries or language features;


Need a particular setup or in-code annotations. For example, teams can only use the Checker Framework's nullness analysis⁹ if their code is annotated appropriately. Another analysis, when configured, will check the increase in binary size and method count for a particular Android binary and warn developers if there is a significant increase or if they are approaching a hard limit;

Support custom domain-specific languages (DSLs) and team-specific coding guidelines. Some Google software development teams have developed small DSLs with associated validators they wish to run. Other teams have developed their own best practices for readability and maintainability and would like to enforce those checks; and


Are highly resource-intensive. An example is hybrid analyses that incorporate results from dynamic analysis. Such analyses provide high value for some teams but are too costly or slow for all.

As of January 2018, there were approximately 70 optional analyses available within Google, and 2,500 projects had enabled at least one of them. Dozens of teams across the company are actively developing a new analyzer, most outside the developer-tools group.

Additional workflow integration points. As developers have gained trust in the tools, they have also requested further integration into their workflow. Tricorder now provides analysis results through a command-line tool, a continuous integration system, and a code-browsing tool.



Engineers working on static analysis must demonstrate impact through hard data.



Command line support. The Tricorder team added command-line support for developers who are, in effect, code janitors, regularly going through and scrubbing their team's codebase of various analysis warnings. These developers are also very familiar with the types of fixes each analysis will generate and have high trust in specific analyzers. Developers can thus use a command-line tool to automatically apply all fixes from a given analysis and generate cleanup changes;

Gating commits. Some teams want specific analyzers to actually block commits, rather than just appear in the code-review tool. The ability to block commits is commonly requested by teams that have highly specific custom checks with no false positives, usually for a custom DSL or library; and

Results in code browsing. Code browsing works best for showing the scale of a problem across a large project (or an entire codebase). For example, analysis results when browsing code about a deprecated API can show how much work a migration entails; or some security and privacy analyses are global in scope and require specialized teams to vet the results before determining whether there is indeed a problem. Since analysis results are not displayed by default, the code browser allows specific teams to enable an analysis layer and then scan the entire codebase and vet the results without disrupting other developers with distractions from these analyzers. If an analysis result has an associated fix, then developers can apply the fix with a single click from the code-browsing tool. The code browser is also ideal for displaying results from analyses that utilize production data, as this data is not available until code is committed and running.

Sophisticated analyses. All of the static analyses deployed widely at Google are relatively simple, although some teams work on project-specific analysis frameworks for limited domains (such as Android apps) that do interprocedural analysis. Interprocedural analysis at Google scale is technically feasible. However, implementing such an analysis is very challenging. All of Google's code resides in a single monolithic source code repository, as discussed, so, conceptually, any code in the repository can be part of any binary. It is thus possible to imagine

a scenario in which analysis results for a particular code review would require analyzing the entire repository. Although Facebook's Infer^{7,25} focuses on compositional analysis in order to scale separation-logic-based analysis to multimillion-line repositories, scaling such analysis to Google's multibillion-line repository would still take significant engineering effort.

As of January 2018, implementing a system to do more sophisticated analyses has not been a priority for Google since:

Large investment. The up-front infrastructure investment would be prohibitive;

Work needed to reduce false-positive rates. Analysis teams would have to develop techniques to dramatically reduce false-positive rates for many research analyzers and/or severely restrict which errors are displayed, as with Infer;

Still more to implement. Analysis teams still have plenty more "simple" analyzers to implement and integrate; and

High upfront cost. We have found the utility of such "simple" analyzers to be high, a core motivation of FindBugs.²⁴ In contrast, even determining the cost-benefit ratio for more complicated checks has a high up-front cost.

Note this cost-benefit analysis may be very different for developers outside of Google working in specialized fields (such as aerospace¹³ and medical devices²¹) or on specific projects (such as device drivers⁴ and phone apps⁷).

Lessons

Our experience attempting to integrate static analysis into Google's workflow taught us valuable lessons:

Finding bugs is easy. When a codebase is large enough, it will contain practically any imaginable code pattern. Even in a mature codebase with full test coverage and a rigorous code-review process, bugs slip by. Sometimes the problem is not obvious from local inspection, and sometimes bugs are introduced by seemingly harmless refactorings. For example, consider the following code snippet hashing a field `f` of type `long`

```
result =
    31 * result
    + (int) (f ^ (f >>> 32));
```

Now consider what happens if the developer changes the type of `f` to `int`. The code continues to compile, but the right shift by 32 becomes a no-op, the field is XORed with itself, and the hash for the field becomes a constant 0. The result is `f` no longer affects the value produced by the `hashCode` method. The right shift by more than 31 is statically detectable by any tool able to compute the type of `f`, yet we fixed 31 occurrences of this bug in Google's codebase while enabling the check as a compiler error in Error Prone.

Since finding bugs is easy,²⁴ Google uses simple tooling to detect bug patterns. Analysis writers then tune the checks based on results from running over Google code.

Most developers will not go out of their way to use static analysis tools. Following in the footsteps of many commercial tools, Google's initial implementation of FindBugs relied on engineers choosing to visit a central dashboard to see the issues found in their projects, though few of them actually made such a visit. Finding bugs in checked-in code (that may already be deployed and running without user-visible problems) is too late. To ensure that most or all engineers see static-analysis warnings, analysis tools must be integrated into the workflow and enabled by default for everyone. Instead of providing bug dashboards, projects like Error Prone extend the compiler with additional checks, and surface analysis results in code review.

Developer happiness is key. In our experience and in the literature, many attempts to integrate static analysis into a software-development organization fail. At Google, there is typically no mandate from management that engineers use static analysis tools. Engineers working on static analysis must demonstrate impact through hard data. For a static analysis project to succeed, developers must feel they benefit from and enjoy using it.

To build a successful analysis platform, we have built tools that deliver high value for developers. The Tricorder team keeps careful accounting of issues fixed, performs surveys to understand developer sentiment, makes it easy to file bugs against the analysis tools, and uses all this data to justify continued investment. Developers need to build trust in analysis tools. If

a tool wastes developer time with false positives and low-priority issues, developers will lose faith and ignore results.

Do not just find bugs, fix them. To sell a static analysis tool, a typical approach is to enumerate a significant number of issues that are present in a codebase. The intent is to influence decision makers by indicating a potential ability to correct the underlying bugs or prevent them in the future. However, that potential will remain unrealized if developers are not incentivized to act. This is a fundamental flaw: analysis tools measure their utility by the number of issues they identify, while integration attempts fail due to the low number of bugs actually fixed or prevented. Instead, Google static analysis teams take responsibility for fixing, as well as finding, bugs, and measure success accordingly. Focusing on fixing bugs has ensured that tools provide actionable advice³⁰ and minimize false positives. In many cases, fixing bugs is as easy as finding them through automated tooling. Even for difficult-to-fix issues, research over the past five years has highlighted new techniques for automatically creating fixes for static analysis issues.^{22,28,31}

Crowdsource analysis development. Although typical static analysis tools require expert developers to write the analyses, experts may be scarce and not actually know what checks will have the greatest impact. Moreover, analysis experts are typically not domain experts (such as those working with APIs, languages, and security). With FindBugs integration, only a small number of Googlers understood how to write new checks, so the small BugBot team had to do all the work themselves. This limited the velocity of adding new checks and prevented others from contributing their domain knowledge. Teams like Tricorder now focus on lowering the bar to developer-contributed checks, without requiring prior static analysis experience. For example, the Google tool Refaster³⁷ allows developers to write checks by specifying example before and after code snippets. Since contributors are frequently motivated to contribute after debugging faulty code themselves, new checks are biased toward those that save developer time.

Conclusion

Our most important insight is that careful developer workflow integration is key for static analysis tool adoption. While tool authors may believe developers should be delighted by a list of probable defects in code they have written, in practice we did not find such a list motivates developers to fix the defects. As analysis-tool developers, we must measure our success in terms of defects corrected, not the number presented to developers. This means our responsibility extends far beyond the analysis tool itself.

We advocate for a system focused on pushing workflow integration as early as possible. When possible, checks are enabled as compiler errors. To avoid breaking builds, tool writers take on the task of first fixing all the existing issues in the codebase, allowing us to “ratchet” the quality of Google’s codebase one small step at a time, without regressions. Since we present the errors in the compiler, developers encounter them immediately after writing code, while they are still amenable to making changes. To enable this, we have developed infrastructure for running analyses and producing fixes over the whole vast Google codebase. We also benefit from code review and submission automation that allows a change to hundreds of files, as well as an engineering culture in which changes to legacy code are typically approved because improving the code wins over risk aversion.

Code review is a sweet spot for displaying analysis warnings before code is committed. In order to ensure developers are receptive to analysis results, Tricorder presents issues only when a developer is changing the code in question, before the change is committed, and the Tricorder team applies a set of criteria to selecting what warnings to display. Tricorder further gathers user data in the code-review tool that is used to detect any analyses that produce unacceptable numbers of negative reactions. The Tricorder team minimizes effective false positives by disabling misbehaving analyses.

To overcome warning blindness, we have worked to regain the trust of Google engineers, finding Google developers have a strong bias to ignore static analysis, and any false positives

or poor reporting give them a justification for inaction. Analysis teams are quite careful to enable a check as an error or warning only after vetting it against the criteria described here, so developers are rarely inundated, confused, or annoyed by analysis results. Surveys and feedback channels are an important quality control for this process. Now that developers have gained trust in analysis results, the Tricorder team is fulfilling requests for more analyses surfaced in more locations in the Google developer workflow.

We have built a successful static analysis infrastructure at Google that prevents hundreds of bugs per day from entering the Google codebase, both at compiletime and during code reviews. We hope others can benefit from our experience to successfully integrate static analyses into their own workflows. C

References

1. Aftandilian, E., Sauciu, R., Priya, S., and Krishnan, S. Building useful program analysis tools using an extensible compiler. In *Proceedings of the International Working Conference on Source Code Analysis and Manipulation* (Riva del Garda, Italy, Sept. 23–24). IEEE Computer Society Press, 2012, 14–23.
2. Ayewah, N., Hovemeyer, D., Morgenthaler, J.D., Penix, J., and Pugh, W. Using static analysis to find bugs. *IEEE Software* 25, 5 (Sept.–Oct. 2008), 22–29.
3. Ayewah, N. and Pugh, W. The Google FindBugs fixit. In *Proceedings of the International Symposium on Software Testing and Analysis* (Trento, Italy, July 12–16). ACM Press, New York, 2010.
4. Ball, T., Bounimova, E., Cook, B., Levin, V., Lichtenberg, J., McGarvey, C., Ondrusek, B., Rajamani, S.K., and Ustuner, A. Thorough static analysis of device drivers *ACM SIGOPS Operating Systems Review* 40, 4 (Oct. 2006), 73–85.
5. Bazel; <http://www.bazel.io>
6. Bessey, A., Block, K., Chelf, B., Chou, A., Fulton, B., Hallem, S., Henri-Gros, C., Kamsky, A., McPeak, S., and Engler, D. A few billion lines of code later. *Commun. ACM* 53, 2 (Feb. 2010), 66–75.
7. Calcagno, C., Distefano, D., Dubreil, J., Gabi, D., Hooimeijer, P., Luca, M., O’Hearn, P.W., Papakonstantinou, I., Purbrick, J., and Rodriguez, D. Moving fast with software verification. In *Proceedings of the NASA Formal Method Symposium* (Pasadena, CA, Apr. 27–29). Springer, 2015.
8. Chambers, C., Raniwala, A., Perry, F., Adams, S., Henry, R., Bradshaw, R., and Weizenbaum, N. FlumeJava: Easy, efficient data-parallel pipelines. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation* (Toronto, Canada, June 5–10). ACM Press, New York, 2010.
9. The Checker Framework; <https://checkerframework.org>
10. Checkstyle Java Linter; <http://checkstyle.sourceforge.net/>
11. Clang Thread Safety Analysis; <http://clang.lvm.org/docs/ThreadSafetyAnalysis.html>
12. ClangTidy; <http://clang.lvm.org/extra/clang-tidy.html>
13. Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Monniaux, D., and Rival, X. The ASTRÉE analyzer. In *Proceedings of the European Symposium on Programming* (Edinburgh, Scotland, Apr. 2–10). Springer, Berlin, Heidelberg, 2005.
14. Dynamic Sanitizer Tools; <https://github.com/google/sanitizers>
15. Error Prone; <http://errorprone.info>
16. FindBugs; <http://findbugs.sourceforge.net/>
17. Go vet; <https://golang.org/cmd/vet>
18. Golint; <https://github.com/golang/lint>

19. Grammatech; <https://resources.grammatech.com/medical>
20. Griesmayer, A., Bloem, R., Cook, B. Repair of Boolean programs with an application to C. In *Proceedings of the 18th International Conference on Computer Aided Verification* (Seattle, WA, Aug. 17–20). Springer, Berlin, New York, 2006.
21. Guava: Google Core Libraries for Java 1.6+; <https://code.google.com/p/guava-libraries/>
22. Gupta, P., Ivey, M., and Penix, J. Testing at the speed and scale of Google. *Google Engineering Tools Blog*, 2011; <http://google-engtools.blogspot.com/2011/06/testing-at-speed-and-scale-of-google.html>
23. Hacker News. Twitter outage report, 2016; <https://news.ycombinator.com/item?id=8810157>
24. Hovemeyer, D. and Pugh, W. Finding bugs is easy. *ACM SIGPLAN Notices* 39, 12 (Dec. 2004), 92–106.
25. Infer; <http://fbinfer.com/>
26. Johnson, B., Song, Y., Murphy-Hill, E.R., and Bowdidge, R.W. Why don’t software developers use static analysis tools to find bugs? In *Proceedings of the 35th International Conference on Software Engineering* (San Francisco, CA, May 18–26). ACM Press, New York, 2013.
27. Klein, D.V., Betser, D.M., and Monroe, M.G. Making ‘push on green’ a reality: Issues and actions involved in maintaining a production service. *Jagim*, 39, 5 (2014), 26–32.
28. Kneuss, E., Koukoutos, M., and Kuncak, V. Deductive program repair. In *Proceedings of the 27th International Conference on Computer Aided Verification* (San Francisco, CA, July 18–24). Springer, 2015.
29. Larus, J.R., Ball, T., Das, M., DeLine, R., Fahndrich, M., Pincus, J., Rajamani, S.K., and Venkatapathy, R. Righting software. *IEEE Software* 21, 3 (May 2004), 92–100.
30. Lewis, C., Lin, Z., Sadowski, C., Zhu, X., Ou, R., and Whitehead, Jr., E. J. Does bug prediction support human developers’ findings?: From a Google case study. In *Proceedings of the 35th International Conference on Software Engineering* (San Francisco, CA, May 18–26). ACM Press, New York, 2013.
31. Logozzo, F. and Ball, T. Modular and verified automatic program repair. *ACM SIGPLAN Notices* 46, 10 (Oct. 19, 2012), 133–146.
32. Potvin, R. and Levenberg, J. Why Google stores billions of lines of code in a single repository. *Commun. ACM* 59, 7 (July 2016), 78–87.
33. Protocol buffers; <http://code.google.com/p/protobuf/>
34. Pylint Python Linter; <http://www.pylint.org/>
35. Sadowski, C., van Gogh, J., Jaspan, C., Söderberg, E., and Winter, C. Tricorder: Building a program analysis ecosystem. In *Proceedings of the 37th International Conference on Software Engineering* (Firenze, Italy, May 16–24). ACM Press, New York, 2015.
36. Synopsys Editorial Team. *Coverity Report on the ‘Goto Fail’ Bug*. Blog post, Synopsys, Mountain View, CA, Feb. 25, 2014; <http://security.coverity.com/blog/2014/Feb/a-quick-post-on-apple-security-55471-aka-goto-fail.html>
37. Wasserman, L. Scalable, example-based refactorings with Refaster. In *Proceedings of the Workshop on Refactoring Tools* (Indianapolis, IN, Oct. 26). ACM Press, New York, 2013.
38. Wright, H., Jasper, D., Klimek, M., Carruth, C., and Wan, Z. Large-scale automated refactoring using ClangMR. In *Proceedings of the 29th IEEE International Conference on Software Maintenance* (Eindhoven, the Netherlands, Sept. 22–28). IEEE Computer Society Press, 2013.

Caitlin Sadowski (supertri@google.com) is a software engineer at Google Inc., Mountain View, CA, USA.

Edward Aftandilian (eaftan@google.com) leads the Java compiler and static analysis team at Google, Inc., Mountain View, CA, USA.

Alex Eagle (alexeagle@google.com) is a software engineer at Google Inc., Mountain View, CA, USA.

Liam Miller-Cushon (cushon@google.com) is a software engineer at Google Inc., Mountain View, CA, USA.

Ciera Jaspan (ciera@google.com) is a software engineer at Google Inc., Mountain View, CA, USA.

Data science promises new insights, helping transform information into knowledge that can drive science and industry.

BY FRANCINE BERMAN, ROB RUTENBAR, BRENT HAILPERN, HENRIK CHRISTENSEN, SUSAN DAVIDSON, DEBORAH ESTRIN, MICHAEL FRANKLIN, MARGARET MARTONOSI, PADMA RAGHAVAN, VICTORIA STODDEN, AND ALEXANDER S. SZALAY

Realizing the Potential of Data Science

THE ABILITY TO manipulate and understand data is increasingly critical to discovery and innovation. As a result, we see the emergence of a new field—data science—that focuses on the processes and systems that enable us to extract knowledge or insight from data in various forms and translate it into action. In practice, data science has evolved as an interdisciplinary field

that integrates approaches from such data-analysis fields as statistics, data mining, and predictive analytics and incorporates advances in scalable computing and data management. But as a discipline, data science is only in its infancy.

The challenge of developing data science in a way that achieves its full potential raises important questions for the research and education community: How can we evolve the field of data science so it supports the increasing role of data in all spheres? How do we train a workforce of professionals who can use data to its best advantage? What should we teach them? What can government agencies do to help maximize the potential of data science to drive discovery and address current and fu-

ture needs for a workforce with data science expertise? Convened by the Computer and Information Science and Engineering (CISE) Directorate of

» insights

- **Data science can help connect previously disparate disciplines, communities, and users to provide richer and deeper insights into current and future challenges.**
- **Data science encompasses a broad set of areas, including data-focused algorithmic innovation and machine learning; data mining and the use of data for discovery; collection, organization, stewardship and preservation of data; privacy challenges and policy associated with data; and pedagogy to support the education and training of data-savvy professionals.**
- **There is a growing gap between commercial and academic research practice for data systems that needs to be addressed.**

the U.S. National Science Foundation as a Working Group on the Emergence of Data Science (<https://www.nsf.gov/dir/index.jsp?org=CISE>), we present a perspective on these questions with a particular focus on the challenges and opportunities for R&D agencies to support and nurture the growth and impact of data science. For the full report on which this article is based, see Berman et al.²

The importance and opportunities inherent in data science are clear (see <http://cra.org/data-science/>). If the National Science Foundation, working with other agencies, foundations, and industry can help foster the evolution and development of data science and data scientists over the next decade, our research community will be better able to meet the potential of data science to drive new discovery and innovation and help transform the information age into the knowledge age. We hope this article serves as a basis for dialogue within the academic community, the industrial research community, and ACM and relevant ACM special interest groups (such as SIGKDD and SIGHPC).

The Data Life Cycle

Data never exists in a vacuum. Like a biological organism, data has a life cycle, from birth through an active life to “immortality” or some form of expiration. Also like a living and intelligent organism, it survives in an environment that provides physical support, social

context, and existential meaning. The data life cycle is critical to understanding the opportunities and challenges of making the most of digital data; see the figure here for the essential components of the data life cycle.

As an example of the data life cycle, consider data representing experimental outputs of the Large Hadron Collider (LHC), an instrument of tremendous importance to the physics community and supported by researchers and nations worldwide. LHC experiments collide particles to test the predictions of various theories of particle physics and high-energy physics. In 2012, data on LHC experiments provided strong evidence for the Higgs Boson, supporting the veracity of the Standard Model of Physics. This scientific discovery was *Science Magazine’s* 2012 “Breakthrough of the Year”³ and Nobel Prize for Physics in 2013.

The life cycle of LHC data is fascinating. At “birth,” data represents the results of collisions within an instrument carried out in a 17-mile tunnel on the France-Switzerland border. Most of the data generated is technically “uninteresting” and disposed of, but a tremendous amount of “interesting” data remains to be analyzed and preserved. Estimates are that by 2040, there will be from 10 exabytes to 100 exabytes (billion trillion bytes) of “interesting” data produced by the LHC. Retained LHC data is annotated, prepared for preservation, and archived at more than a dozen physical sites. It is published

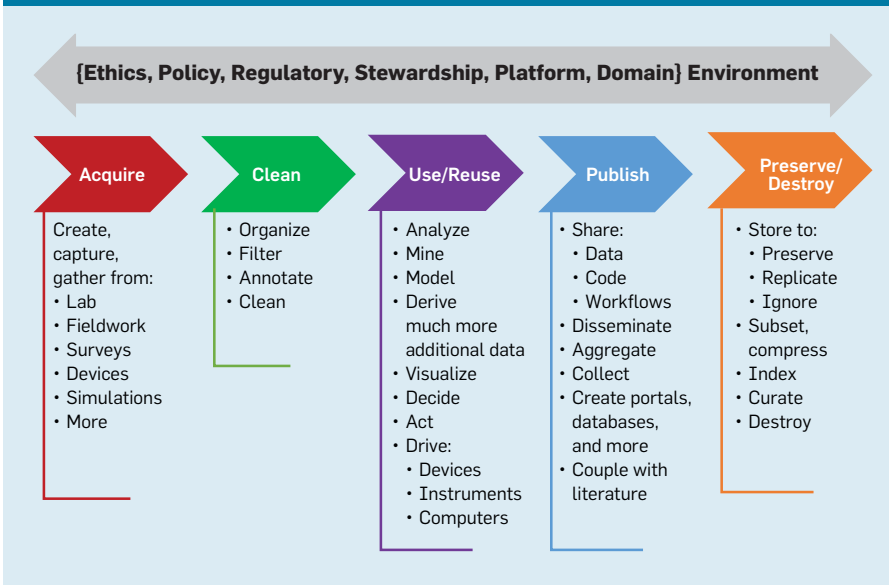
and disseminated to the community for analysis and use at more than 100 other research sites. Critical attention to stewardship, use, and dissemination of LHC data throughout its life cycle has played a key role in enabling the scientific breakthroughs that have come from the experiments.

In addition to development of data stewardship, dissemination, and use protocols, the LHC data ecosystem also provides an economic model that sustainably supports the data and its infrastructure. It is the combination of this greater ecosystem, community agreements about how the data is organized, and political and economic support that allow LHC data to meet its potential to transform our knowledge of physics and enable scientists to make the most of the tremendous investment being made in the LHC’s physical instruments and facilities.

The data life cycle diagram outlined in the figure and the LHC example suggest a seamless set of actions and transformations on data, but in many scientific communities and disciplines today these steps are isolated. Domain scientists focus on generating and using data. Computer scientists often focus on platform and performance issues, including mining, organizing, modeling, and visualizing, as well as the mechanisms for eliciting meaning from the data through machine learning and other approaches. The physical processes of acquisition and instrument control are often the focus of engineering, or data as “dirty signals” or as control inputs for other equipment. Statisticians may focus on the mathematics of models for risk and inference. Information scientists and library scientists may focus on stewardship and preservation of data and the “back-end” of the pipeline, following acquisition, decisions, and action in the realm of publishing, archiving, and curation.

There is a significant opportunity for bridging gaps in development of effective life cycles for valuable data within and among the computer science, information science, domain, and physical science and engineering communities, for a start. There is also an opportunity for bridging gaps among machine learning, data analytics, and related disciplines (such as statistics

The data life cycle and surrounding data ecosystem from the *Realizing the Potential of Data Science Report*.²



and operations research). Here we focus on some opportunities.

National Data Science Research

Almost every stage of the data lifecycle, as outlined in the figure, provides deep research opportunities. Moreover, an overarching area of opportunity for a national data science agenda is to bridge the gaps in the life cycle, building stronger connections among the computer science, information science, statistics, domain, and physical science and engineering communities, as outlined earlier. That is, a business-as-usual research agenda is likely to strengthen individual technologies behind discrete steps in the data life cycle but unlikely to nurture broader breakthroughs or paradigm shifts that cut across existing disciplinary silos. It is an essential and defining attribute of data (“big” and otherwise) that it can connect previously disparate disciplines, communities, and users to provide richer and deeper insight into current and future challenges.

It is vital to encourage a broader and more holistic view of data as integrating research opportunities across the sciences, engineering, and range of application domains. One such opportunity is to invest in the full data life cycle and surrounding environment—as a central outcome itself, not as a side effect or intermediate step to another desirable outcome. In parallel with development of data science in depth as a core component of computer science, data science should also evolve in breadth to address the needs of domains outside computer science. Our community has a unique opportunity to advance data science, with respect to applying data-driven strategies to individual domain research and cross-domain research opportunities.

A second opportunity involves what might be called “embodied intelligence” scenarios that big data is enabling for the first time. Recent breakthroughs in a range of foundational artificial intelligence and “deep learning” technologies¹ have made it possible to create sophisticated software artifacts that “act intelligently.” The key innovations are in mathematical-pattern-recognition techniques that take input from millions of training examples of correct responses to create software systems (soon likely hardware systems as well) able to better recog-

Teaching Data Science: Many Flowers Blooming

To support research and workforce development for data science, we must determine how—and, interestingly, *where* in the institution—it should be taught. Much as the emergence of computer science in the 1960s created the first organizational units and degrees dedicated to computing in modern universities, the rise of data science is driving a range of interesting curricular experiments. For a sense of the rapidly evolving landscape, consider these five:

University of California. The University of California, Berkeley, Data Science Education Program⁸ is part of its recently established Division of Data Sciences, at the same level as Berkeley’s colleges and schools, integrating with them. The introductory class provides a foundation for students in all fields to engage with data and creates pathways to advanced-level work. The foundational course combines instruction in core computational and statistics concepts while enabling students to work with real data in a range of fields. It is designed to be accessible to undergraduates of any intended major without prior experience. A set of connector courses (mostly taken simultaneously with the foundational course) enable them to apply core skills from the foundational course to real-world issues that relate to their areas of interest. There are also advanced courses, including an upper-division integrative course called “Data 100 Principles and Techniques of Data Science.”

University of Michigan. The University of Michigan Undergraduate Program in Data Science¹¹ is a new major offered as a joint program by the Electrical Engineering and Computer Science and Statistics Departments. The data science major is a rigorous program focusing on aspects of computer science, statistics, and mathematics relevant for analyzing and manipulating large datasets. It can be entered from either the College of Engineering or the College of Literature, Science, and Arts.

Columbia University. The Columbia University Data Science Institute’s Master of Science in Data Science⁴ offers a professional master’s degree to students with any undergraduate degree that includes suitable quantitative prior coursework. It starts from a set of four foundational courses (that can be taken independently to yield a data science certificate), focusing on algorithms, probability/statistics, machine learning, and visualization.

University of Illinois. The University of Illinois at Urbana-Champaign Master of Computer Science in Data Science degree¹⁰ is offered as an online professional master’s in computer science available on the Coursera massive open online course platform.⁵ The degree seeks to create a global gateway into the discipline. The program builds expertise in four core areas of computer science—data visualization, machine learning, data mining, and cloud computing—and also offers courses in collaboration with the university’s Statistics Department and School of Information Sciences. This collaboration specifically strives to cover the full data life cycle, including its mathematical, computational, and curation and stewardship components, in an integrated and comprehensive fashion.

University of Chicago. The University of Chicago Master of Science in Computational Analysis and Public Policy program⁹ is offered jointly by the Department of Computer Science and the Harris School of Public Policy. As government decision making is increasingly data-driven, data use, data sharing, transparency, and accountability become increasingly important issues from both a public policy and a technological perspective. The program focuses on the intersection of policy and computer science. Students take courses across both areas, preparing them to make meaningful contributions to the design, implementation, and rigorous analysis of policies in the public sector.

nize images, decode human speech, discover critical patterns in legal and business documents, and more. As engineered artifacts, these artificial intelligence systems are embodied as complex mathematical formulae that are customized to purpose, or “trained,” by a truly astounding volume of numerical parameters (such as 10 million for a decent image-classification system today).

These trained decision-oriented models are becoming core components in a range of novel software solutions to

complex problems, creating cross-disciplinary challenges.⁶ For example, what does it mean for such a component to be “correct” when it is perhaps only 70% accurate? What should the life cycle be for the data used to train and update these models? What are the policy implications (and designation of responsibility) for embodied intelligent agents trained on such data that behave with negative consequences (such as when blamed for an autonomous vehicle that crashes, or by a customer whose account is suspend-

ed inappropriately based on an automatic inference)? Software engineering, as a discipline, is challenged by such imprecision and with versioning and testing of the enormous data components—giga-byte-to-terabyte scale training data—for these systems. Existing notions of model verification/validation seem woefully insufficient. And the policy, stewardship, and curation questions go largely unasked and unanswered.

Note that the existence of predictive models is not unique to machine learning; for example, statistical models have been used in epidemiology, and physical models are common in weather prediction and nuclear simulations. The “training” aspect for data science may be novel in the context of the software engineering of solutions, in that the resulting models may lack the guarantees associated with statistical power and sample-size calculations.

Yet another opportunity is to address the growing gap between commercial and academic research practice for data systems at the edge of the state of the art. Much has been made of the increasing “reverse migration” of strong academic researchers into data-rich enterprises (such as Facebook, Google, and Microsoft). While this is likely good for the U.S. national economy in the near term, it is worrisome for the future of discovery-based open research, education, and training in the academic sector. In addition to the challenges of attracting sponsored research funding, another reason for the “brain drain” from the research community into the private sector may be declining infrastructure-support environments, including the sparsity of large datasets and adequate infrastructure in academia that support data science research at scale. When the best infrastructure environment for cutting-edge research is consistently in the private sector, the opportunity for innovation in the public sector deteriorates. Government support for strategic and committed public-private partnerships that build adequate and representative at-scale infrastructure in the academic community for researchers can unlock innovation in academic research and ultimately support the private sector through development of a more sophisticated, educated, better-trained workforce.

National Data Science Education and Training

Higher-education institutions across the U.S. recognize that data science is a critical skill for 21st-century research and a 21st-century workforce. In higher education, data science curricula have two audiences: new professionals in data science, and scientists and professionals who need data science skills to contribute to other fields. Data science curricula in higher education often focus on both, the same way curricula in computer science departments educate computer science students and provide training in computer skills to students from other disciplines to promote computer literacy.

It is important to note that, at present, there is no single model of which department, school, or cross-unit collaboration within higher-education institutions should have the responsibility for data science education and training. Data science programs are being sited in departments and schools of computer science, information science, statistics, and management. Many of the most successful, particularly at the undergraduate level, represent university-wide coalitions frequently sponsored by interdisciplinary institutes, rather than by a particular department or school. There is thus no common agreement as to where data science should “live” in the institution, though there is much interesting experimentation at this point (see the sidebar “Teaching Data Science” for several programmatic configurations). Note that when a university chooses to house “data science” in an existing department or college, it implicitly adopts the standards and culture of that existing organization. In contrast, when a university introduces “data science” as an interdisciplinary function, it confronts the heterogeneity of the new field up front but will likely deal with additional administrative overhead associated with a cross-organizational entity. We focus on trends in both data science education and training in the following paragraphs.

Educational curricula in data science have yet to “standardize” and appear today with many interesting course configurations. In general, data scientists are expected to be able to analyze large datasets using statistical techniques, so statistics and modeling are typically

part of required coursework. Moreover, a comprehensive data science curriculum is more than machine learning and statistics, possibly including courses on programming, data stewardship, and ethics, in addition to other areas. Data scientists must be able to find meaning in unstructured data, so classes on programming, data mining, and machine learning are often part of the core. Data scientists must also be able to communicate their findings effectively, so courses on visualization may be offered, at least as an elective. In recognition of the challenges that arise from misuse of data and incorrect conclusions drawn from data, ethics is also becoming a part of responsible curricula for the field.

Other courses that appear either in the core or as an elective in various programs include research design, databases, algorithms, parallel computing, and cloud computing, all of which reflect skills an employer might expect from a data scientist. Many programs also require a capstone project that gives students experience in working through real-world problems in teams in a particular domain. Data science courses are also becoming a staple of quality online programs.

A strong data science curriculum requires faculty with appropriate expertise and engagement with the field. The pull of faculty with expertise in data science and related fields away from academia and toward industry creates a challenge for educational institutions in mounting such programs. It also presents a potential challenge to development of data science as a formal discipline.

To combat this trend, the Moore and Sloan Foundations in 2013 created a joint \$38 million project, the Moore-Sloan Data Science Environments, to fund initiatives to create “data science environments,”⁷ addressing challenges in academic careers, education and training, tools and software, reproducibility and open science, physical and intellectual space, and data science studies. This funding has been transformational, providing critical “worked examples” of data science programs useful for current and future efforts.

From the current diversity of curricula and programs, data science is going through an important and healthy period of experimentation. It is important


that we do not “standardize” data science too quickly, continuing to explore configurations of courses, areas, projects, faculty, and partnerships to gain critical experience in how to best educate new generations of data scientists.

In addition to “data science” programs and majors that serve to evolve data science as a discipline, data science skills are increasingly critical as training for other disciplines and professions as they become more and more data-enabled. Effective training will empower data-enabled professionals and domain scientists to utilize data effectively and operate within a broader data-driven environment, develop an appreciation of what data can tell us and what it cannot, acquire appropriate technical knowledge about how data should be handled, gain awareness that correlation in data does not necessarily imply causality, and begin to develop a sense of responsible methodologies and ethical principles in the use of data.


More specific training in the nuts and bolts of dealing with data is also critical for various data-driven professions. Training in programming and software engineering is useful for students who will be using data-driven simulations and models in their research. Training in version control and the subtleties of stewardship, including working with repositories for data and software, should be taught to computational researchers. And training in best practices for digital scholarship and reproducibility should be integrated into research-methodology curricula. The ethics of using (and misusing) data should be incorporated into all training programs to promote effective and responsible data use. Courses teaching these skills can be made available in a variety of venues, from university courses and modules to online courses to professional courses that could be developed by scientific societies and communities.

Data Science Research and Education Infrastructure

Any innovative agenda in data science research and education will depend on a foundation of enabling data infrastructure and useful datasets. Research in data science needs access to sufficiently large and numerous datasets to illuminate and validate results.



At present, there is no single model of which department, school, or cross-unit collaboration within higher-education institutions should have the responsibility for data science education and training.



The datasets must be available for reproducible research and hosted by reliable infrastructure.

Lack of such infrastructure and datasets will inhibit success. Education and training in data science is most authentic in a setting where students can work on data that represents the datasets and environments they will see in the professional arena; that is, data that is both at-scale and embedded in a stewardship infrastructure that enables it to be a useful tool in analysis, modeling, and mining.

In the best case, data infrastructure should support access to data for research and education that is equivalent to access to any other key utility; it must be “always on,” it must be robust enough to support extensive use, and the quality must be good. In the world of data, this comes down to responsible stewardship, meaning there must be actors, plans, and both “social” and technical infrastructure to ensure the following:

Data is appropriately tracked, monitored, and identified. Who created, curated, and used the data? Can it be persistently identified? Are there adequate privacy and security controls?;

Data is well cared for. Who is committed to keeping it, in what formats, and for how long? Who is committed to funding data stewardship? And how will it be stored and migrated to next-generation media?;

Data is discoverable and useful. How is data made available and to whom? What services are needed to make good use of it? And what metadata and other information is needed to promote reproducibility?; and

Data stewardship is compliant with policy and good practice. Does stewardship comply with community standards and appropriate policy regarding reporting, intellectual property, and other concerns? Are the rights, licenses, and other properties that will determine appropriate use clear? And what data and metadata are to be kept, who owns it and its by-products, and who has access to it and its metadata or parts of it?

Since data will become the core for research and insight for a broad set of academic disciplines, access to it in a usable form on a reasonable time scale becomes the entry point for any effective research and education agenda.

Government R&D agencies (such as the National Science Foundation) have an opportunity to ensure the lack of adequate data infrastructure does not present a roadblock to innovative research and educational programs.

Developing and sustaining the infrastructure that ensures that research data is available to the public and accessible for reuse and reproducibility requires stable economic models. While there is much support for the development of tools, technologies, building blocks, and data-commons approaches, few U.S. federal programs directly address the resource challenges for data stewardship or provide help for libraries, domain repositories, and other stewardship environments to become self-sustaining and address the need for public access.

While the U.S. federal government cannot take on the entire responsibility for stewardship of sponsored research data and its infrastructure, neither should it shy away from providing seed or transition funding for institutions and organizations to develop sustainable stewardship options for the national community. We encourage the community, inside and outside of government, to support the development and piloting of sustainable data stewardship models for data-driven research and data science education through strategic programs, guidance, and cross-agency and public-private partnerships. Science-centric government agencies like the National Science Foundation should coordinate with peer agencies like the National Institutes of Health that focus on similar issues to leverage investments and provide economies of scope and scale.

Realizing the Potential

The research, education, and infrastructure discussions here focus on developing a foundation that can increase the pool of data scientists and data-literate professionals to meet the current and near-term challenges of data-driven efforts in all sectors, as well as the need to evolve data science as a discipline that can meet the challenges of future data-driven scenarios.

Data is everywhere, providing an increasingly important tool for a broad spectrum of endeavors. As systems grow “smarter” and take on more autonomous and decision-making capabilities,

we will increasingly face data science technical challenges and the social challenges of governance, ethics, policy, and privacy. Addressing them will be critical to rendering data-driven systems useful, effective, and productive, rather than intrusive, limiting, and destructive. Such solutions will be particularly important in highly data-driven environments like the Internet of Things. Moreover, as fundamental computational platforms change in response to the looming end of Moore’s Law scaling of semiconductors,¹² there will be tremendous opportunities to reimagine the entire hardware/software enterprise in the light of future data needs.

Conclusion

Our community must be prepared to deal with future scenarios by encouraging the initial research that lays the groundwork for innovative uses of data, well-functioning data-focused systems, useful policy and protections, and effective governance of data-driven environments. With both programmatic resources and a platform for community leadership, federal R&D agencies like the National Science Foundation play an important role in guiding the community toward innovation. Attention to deep efforts needed to expand the field and its impact, as well as broad efforts to help data science reach its potential for transforming 21st-century research, education, commerce, and life, are needed.

Acknowledgments

We would like to thank the National Science Foundation for convening this group and the institutions and organizations of the co-authors for their support for this work. C

References

1. Bengio, Y., LeCun, Y., and Hinton, G. Deep Learning. *Nature* 521 (May 28, 2015), 436–444.
2. Berman, F. (co-chair), Rutenbar, R. (co-chair), Christensen, H., Davidson, S., Estrin, D., Franklin, M., Hailpern, B., Martonosi, M., Raghavan, P., Stodden, V., and Szalay, A. *Realizing the Potential of Data Science: Final Report from the National Science Foundation Computer and Information Science and Engineering Advisory Committee Data Science Working Group*. National Science Foundation Computer and Information Science and Engineering Advisory Committee Report, Dec. 2016; <https://www.nsf.gov/cise/ac-data-science-report/CISEACDataScienceReport1.19.17.pdf>
3. Cho, A. The discovery of the Higgs Boson. *Science* 338, 6114 (Dec. 21, 2012), 1524–1525.
4. Columbia University Data Science Institute. Master of Science in Data Science; <http://datascience.columbia.edu/master-of-science-in-data-science>
5. Coursera. Master of Computer Science in Data

- Science; <https://www.coursera.org/university-programs/masters-in-computer-data-science>
6. Dhar, V. When to trust robots with decisions, and when not to. *Harvard Business Review* (May 17, 2006); <https://hbr.org/2016/05/when-to-trust-robots-with-decisions-and-when-not-to>
7. Moore-Sloan Data Science Program; <http://msdse.org/>
8. University of California, Berkeley. Data Science Education Program; <http://data.berkeley.edu/data-science-education-program>
9. University of Chicago. Master of Science in Computational Analysis & Public Policy; <https://capp.uchicago.edu/>
10. University of Illinois, Urbana-Champaign, CS@ILLINOIS. Master of Computer Science in Data Science, Data Science Track; <http://www.cs.uiuc.edu/academics/graduate/professional-mcs-program/mcs-data-science-track>
11. University of Michigan. Undergraduate Program in Data Science; <https://www.eecs.umich.edu/eecs/undergraduate/data-science/>
12. Waldrop, M.M. The chips are down for Moore’s Law. *Nature* 530, 7589 (Feb. 11, 2016), 144–146.

Francine Berman (bermanf@rpi.edu) is the Edward P. Hamilton Distinguished Professor in Computer Science at Rensselaer Polytechnic Institute, Troy, NY, USA, and Chair of the Research Data Alliance / U.S. She served as Co-Chair of the Data Science Working Group of the NSF CISE Advisory Committee.

Rob Rutenbar (rutenbar@pitt.edu) is a professor of computer science and electrical and computer engineering and Senior Vice Chancellor for Research at the University of Pittsburgh, Pittsburgh, PA, USA. He served as Co-Chair of the Data Science Working Group of the NSF CISE Advisory Committee.

Henrik Christensen (hichristensen@ucsd.edu) is a professor of computer science and Director of the Institute for Contextual Robotics at the University of California at San Diego, USA.

Susan Davidson (susan@cis.upenn.edu) is the Weiss Professor of Computer and Information Science at the University of Pennsylvania, Philadelphia, PA, USA.

Deborah Estrin (destrin@cs.cornell.edu) is Associate Dean and professor of computer science at Cornell Tech in New York City and a professor of public health at Weill Cornell Medical College, New York, USA.

Michael Franklin (mjfranklin@uchicago.edu) is the Liew Family Chairman of Computer Science and Senior Advisor to the Provost for Data and Computing at the University of Chicago, USA.

Brent Hailpern (bth@us.ibm.com) is a Distinguished Research Staff Member, Science Director of the IBM Cognitive Horizons Network, and Head of Computer Science for IBM Research, San Jose, CA, USA.

Margaret Martonosi (mrm@princeton.edu) is the Hugh Trumbull Adams ‘35 Professor of Computer Science at Princeton University, Princeton, NJ, USA.

Padma Raghavan (padma.raghavan@vanderbilt.edu) is a professor of computer science and computer engineering and Vice President of Research at Vanderbilt University, Nashville, TN, USA.

Victoria Stodden (vcs@illinois.edu) is an associate professor in the School of Information Sciences at the University of Illinois at Urbana-Champaign, USA.

Alex Szalay (szalay@jhu.edu) is Bloomberg Distinguished Professor in the Departments of Physics and Astronomy and Computer Science at the Johns Hopkins University, Baltimore, MD, USA.

© 2018 ACM 0001-0782/18/4 \$15.00



Watch the authors discuss their work in this exclusive *Communications* video. <https://cacm.acm.org/videos/realizing-the-potential-of-data-science>

ACM Welcomes the Colleges and Universities Participating in ACM's Academic Department Membership Program

ACM now offers an Academic Department Membership option, which allows universities and colleges to provide ACM Professional Membership to their faculty at a greatly reduced collective cost.

The following institutions currently participate in ACM's Academic Department Membership program:

- Appalachian State University
- Armstrong State University
- Ball State University
- Berea College
- Bryant University
- Calvin College
- Colgate University
- Colorado School of Mines
- Edgewood College
- Franklin University
- Georgia Institute of Technology
- Governors State University
- Harding University
- Hofstra University
- Howard Payne University
- Indiana University Bloomington
- Mount Holyoke College
- Northeastern University
- Ohio State University
- Old Dominion University
- Pacific Lutheran University
- Pennsylvania State University
- Regis University
- Roosevelt University
- Rutgers University
- Saint Louis University
- San José State University
- Shippensburg University
- St. John's University
- Trine University
- Trinity University
- Union College
- Union University
- University of California, Riverside
- University of Colorado Denver
- University of Connecticut
- University of Illinois at Chicago
- University of Jamestown
- University of Memphis
- University of Nebraska at Kearney
- University of Nebraska Omaha
- University of North Dakota
- University of Puget Sound
- University of the Fraser Valley
- University of Wyoming
- Virginia Commonwealth University
- Wake Forest University
- Wayne State University
- Western New England University
- Worcester State University

Through this program, each faculty member receives all the benefits of individual professional membership, including *Communications of the ACM*, member rates to attend ACM Special Interest Group conferences, member subscription rates to ACM journals, and much more.

The challenge of combatting malware designed to breach air-gap isolation in order to leak data.

BY MORDECHAI GURI AND YUVAL ELOVICI

Bridgeware: The Air-Gap Malware

MANY ORGANIZATIONS STORE and process sensitive information within their computer networks. Naturally, such networks are the preferred targets of adversaries due to the valuable information they hold. Securing computer networks is a complex task involving the installation of endpoint protection, maintaining firewalls, configuring intrusion detection and intrusion prevention systems (IDSs and IPSs), and so on. However, regardless of the level of protection, a persistent attacker will eventually find a way to breach a computer network connected to the Internet. Consequently, if a network stores sensitive or classified information, an ‘air-gap’ approach is often used to prevent such a breach.

Air-gapped networks have no physical or logical connection to public networks (such as, the Internet). Such networks are often used in cases where the information stored in, or generated by, the system is too sensitive to risk data leaks, for example, military networks such as the Joint Worldwide Intelligence

Communications System (JWICS).¹² Air-gapped networks are also commonly used in critical infrastructure and control systems where breaching incidents can have catastrophic results, however such networks are not limited to military or critical infrastructures. Stock exchanges, insurance companies, biomedical manufacturers, and a wide range of industries use isolated networks in their IT environments.³⁰ These networks maintain intellectual property, financial data, trade secrets, confidential documents, and personal information, and air-gap isolation is aimed at protecting this data.

Breaching the air-gap vs. bridging the air-gap. Despite the physical isolation and lack of external connectivity, attackers have successfully compromised such networks in the past. The most famous cases are Stuxnet and Agent.btz,³⁸ although other incidents have been reported from time to time.⁴³ Motivated attackers can breach air-gapped networks in different ways. In recent years, some of the tactics attackers have used in order to achieve this goal have been exposed. A supply chain attack is a method in which attackers load malware onto computer systems in the supply network. Other tactics include infecting a USB drive, which is then used within the targeted network by a deceitful or malicious insider with the appropriate credentials. Several recent incidents have shown that these

» key insights

- “Air-gap” in cyber security refers to a situation in which a sensitive computer, classified network, or critical infrastructure is intentionally isolated from public networks such as the Internet.
- While breaching air-gapped networks has been proven feasible in recent years, data exfiltration from air-gapped networks is a challenging phase of an advanced cyber attack.
- We focus on a type of malware that allows attackers to overcome air-gap isolation in order to leak data. We survey various covert channels proposed over the years, examine their characteristics and limitations, and discuss the relevance of these threats and the likelihood of related cyber attacks in the modern IT environment.

Air-gap research page: <https://cyber.bgu.ac.il/advanced-cyber/airgap>



types of breaches are feasible.¹⁴

Breaching the internal network is only the first phase of an attack. After infiltrating the network, the attacker must maintain a communication channel with the malware in order to receive data. To that end, the attacker must move beyond breaching the air-gap and bridge the air gap that separates the attacker from the targeted network in which the malware is operating or is present. Although a one-time breach into an air-gapped network is evidently possible, continuous bridging of the air-gap in order to facilitate the exfiltration of data is a significantly more challenging task.

TEAPOT, TEMPEST, and EMSEC

The threat of exfiltrating data through the air-gap has been the subject of public research since the 1990s, but investigation actually began much earlier with governmental research conducted by the U.S. Department of Defense. The basic idea behind this research is based on the fact that computer systems are electronic devices that emanate electromagnetic radiation at various wavelengths and strengths. By intentionally manipulating the emanated radiation, information can be modulated and leaked out of the system despite the physical isolation of an air-gap.

The fact that the U.S. defense research community has considered the malicious potential of such techniques for a long period of time is reflected by the presence of a definition for a code word for this field of study in an NSA document that was partially declassified in 1999:

“TEAPOT: A short name referring to the investigation, study, and control of intentional compromising emanations (i.e., those that are hostilely induced or provoked) from telecommunications and automated information systems equipment.”³⁴ Also related are the terms ‘TEMPEST’ and ‘compromising emanation’ which refer to the threat posed by emissions that unintentionally leak from electronic devices:

“Compromising Emanations: Unintentional data-related or intelligence-bearing signals that, if intercepted and analyzed, disclose the information transmitted, received, handled, or otherwise processed by any information processing equipment. See TEMPEST.”³³

The term TEMPEST is now commonly used in modern academic literature and publications to generally describe any threat or defense related to compromising emanation. EMSEC (Emanation/Emission Security) refers to countermeasures used to defend against TEAPOT and TEMPEST threats.

Issues related to TEMPEST, TEAPOT, and EMSEC began to attract public attention in 1985 when Dutch computer researcher Wim van Eck published the first paper on the topic.⁴² Van Eck successfully eavesdropped content from a CRT screen at a range of tens of meters, using just \$15 worth of equipment. Since the beginning of this century, various academic research and publications have introduced new techniques to compromise emanation and data leakage from air-gapped facilities. Interestingly, in 2014 documents leaked by NSA contractor, Edward

Snowden, mentioned a product code-named Cottonmouth (CM)—hidden radio transmitters physically installed in USB equipment in order to maintain a bi-directional communication channel to malicious software running on air-gapped facilities.³⁷

Leakage Scenarios

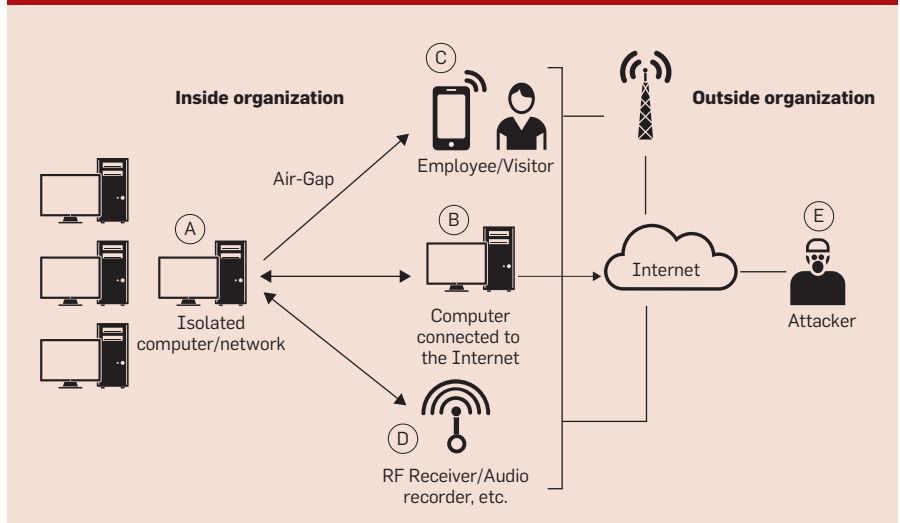
In attacks that involve data leakage, malware is typically used to gather sensitive data (for example, passwords, documents, keystrokes, and personal records) and send it back to the attacker. The data is sent over the Internet, usually in an encrypted form. To evade detection by firewalls and anti-virus software, the malware may also hide the information within so-called *covert channels*. For example, malware may leak a password file within an innocent looking HTTP request. Many types of covert channels have been investigated over the years, including email protocols, DNS requests, and VoIP traffic.⁴⁴ More recently, covert channels for devices such as smartphones and smartwatches,⁹ 3D printers,¹⁰ and IoT devices¹¹ have been also studied.

In an air-gapped network, a malware does not have access to an Internet connection, and hence special types of covert channels must be used. In order to leak out data, the malware exploits the emanations from different components of the computer to establish an out-of-band covert communication channel with the outer world. Similar to any other communication channel, these covert channels have two communicating ends, the *transmitter* and the *receiver*. In the context of air-gap bridging, there are three different scenarios for air-gap communication, each involving different types of transmitters and receivers (depicted in Figure 1):

1. *Computer-to-computer*. In this scenario the malware is capable of transferring data between two closely positioned air-gapped computers, one of which has Internet connectivity. As illustrated, a computer (A) leaks data to a nearby computer (B). (B), in turn, transfers the information to the attacker (E) over the Internet. Such a scenario is likely in modern offices where computers from different networks may be positioned alongside one another, for practical purposes or due to space limitations.

2. *Computer-to-mobile*. In this scenario the malware is capable of transmitting

Figure 1. Three scenarios of bridging the air-gap between an isolated network and an attacker.



data from an air-gapped computer to a nearby mobile phone. As illustrated, a computer (A) leaks data to a nearby mobile phone carried by an employee/visitor (C). (C), in turn, transfers the information to the attacker (E) over the Internet via cellular data or Wi-Fi. This scenario has become more realistic with the new bring-your-own-device trend, where employers commonly bring their personal mobile devices (smartphones, tablets, and mini-tablets) to the workplace.

3. *Computer-to-equipment.* In this scenario the malware is capable of transmitting data from an air-gapped computer to dedicated equipment such as a sound recorder, RF receiver, or remote camera. As illustrated, a computer (A) leaks data to a receiver located some distance away (D). (D), in turn, transfers the information to the attacker (E) over the Internet.

Non air-gapped networks. In addition to the aforementioned air-gap scenarios, the covert channels discussed in this article are relevant to regular, non air-gapped networks as well. In this scenario, the target network—despite having a connection to the Internet—is highly secured, with heavy monitoring of inbound and outbound traffic. As a result, an attacker may choose not to use Internet traffic for exfiltration, but instead resort to another type of out-of-band covert communication. By doing so, the attacker can bypass security measures such as firewalls, traffic analyzers, and network monitors, remain stealthy, and evade detection.

Covert channels vs. side channels. In light of the leakage scenarios, air-gap covert channels are correlated with TEAPOT attacks in which a malware *intentionally* generates comprising emanation from computer components in order to leak data. Side channels, on the other hand, are correlated with TEMPEST attacks in which attackers make use of the emissions that *unintentionally* leak from a computer. By using side channels, an adversary may be able to obtain knowledge about the data being processed by some devices including CRT displays²⁵ or communication devices.²⁸

Types of Covert Channels

Over the years, various techniques have been proposed, enabling covert communication over an air-gap separation. These techniques can be classified into four main groups: acoustic/ultrasonic, electromagnetic, thermal,

and optical methods. Here, we review these techniques and describe their primary characteristics.

An extensive amount of research has been conducted on a wide range of covert communication channels.⁴⁴ More related to this work, Carrara provides a thoughtful analysis of non-conventional, out-of-band covert channels.⁷ Our work differs from other work in the field (for example, Carrara⁷) in terms of focus, coverage, and discussion.

Focus. We focus on covert channels used by attackers to exfiltrate data from highly secure, air-gap networks. Therefore, we don't discuss topics such as side channels and mobile-to-mobile communication that are less relevant to the attack model.

Coverage. We comprehensively survey both new and existing air-gap covert channels. Notably, a major segment of these covert channels have been developed in recent years and have not yet been covered in previous work (for example, methods found in the following references:^{3,4,6,15,16,18,20-23,32,36,39})

Discussion. Our discussion takes place in the context of cybersecurity. We discuss an attack model and various leakage scenarios in air-gap environments. We also examine its relevance in the modern IT environment, considering such areas as hardware availability, virtualized environments, and the required credentials.

Acoustic methods are based on leaking data over sound waves at sonic and ultrasonic frequencies. Madhavapeddy et al.²⁹ first introduced data transmission over audio in 2005 when they discussed audio-based communication between two computers. Two computers, equipped with speakers and a microphone, can exchange data over audio waves in the same way an old dial-up modem works. Obviously, in its original form, acoustic communication is not covert, since people in the room can easily hear the transmission noise. To prevent this, attackers may resort to ultrasonic communication.

Ultrasonic. The main idea in ultrasonic covert channels is to use a computer speaker to produce audio waves at frequencies that are beyond or at the limit of human hearing capabilities. Humans with perfect hearing can perceive sound frequencies within the range of 20Hz to 20,000Hz. However, sometime around the age of eight, sensitivity to the upper

frequency limit begins to decrease, and most adults cannot hear frequencies above 17,000Hz.⁵

In 2013, Hanspach et al. showed how to construct a covert channel between isolated computers over ultrasonic sound waves.²⁴ They observed that an ordinary speaker and microphone can produce and sense sound waves at up to approximately 24,000Hz, well above the range of human hearing. Consequentially, two computers equipped with speakers and microphones can communicate covertly over ultrasonic sound. They extended the idea and established multi-hop communications to create a wireless network over an air-gap. Each computer in the network receives the data through the microphone, and in turn, broadcasts it to the next computer in the hop. Once a computer with an Internet connection receives the packet, it sends the data to the attacker. Their method could maintain communication between computers at distance of 19.7 meters with a bandwidth of 20 bit per second (bit/sec). In the same way, O'Malley and Choo examined different exfiltration scenarios using laptop speakers and microphones at high-frequency sounds up to approximately 23kHz.³⁵ The concept of an air-gap communication over inaudible sounds has been comprehensively examined by Lee et al.²⁷ and also in Deshotels.⁸

Interestingly, in 2013, security researcher, Dragos Ruiu, claimed to find a malware that he dubbed, "BadBIOS." This malware can communicate between instances of itself across air-gaps using ultrasonic communication between a laptop's speakers and microphone. This is the first reported instance of air-gap malware reported in the wild.¹³

Speakerless computers. Acoustic covert channels rely on the presence of audio hardware and a speaker in the transmitter computer. To that end, common practices and security policies prohibit the use of speakers and microphones in a secure computer, in order to create a so-called "audio-gap."¹ Motherboard audio support may also be disabled to cope with the accidental attachment of speakers to the line out connectors. Obviously, disabling audio hardware and keeping speakers disconnected from sensitive computers can effectively mitigate the acoustic and ultrasonic covert channels presented thus far.⁴⁰

Fansmitter is an acoustic covert channel introduced in 2016 that does not re-

quire speakers or audio hardware.²¹ This method utilizes the noise emitted from the CPU and chassis fans that are present in virtually every computer. A malware can regulate the internal fans' speed in order to control the acoustic waveform emitted from a computer. Binary data can be modulated and transmitted over these audio signals to a nearby mobile phone at eight meters away. A video demonstrating Fansmitter can be viewed online.^a DiskFiltration, also introduced in 2016, is a covert channel that allows leaking data from speakerless air-gapped computers via acoustic signals emitted from the hard disk drive (HDD).²² A malware installed on a compromised machine can generate acoustic emissions at certain audio frequencies by controlling the movements of the HDD's actuator arm. Digital information can be modulated over the acoustic signals and then be picked up by a nearby receiver located a distance of two meters away. A demonstration video of DiskFiltration can also be viewed online.^b Table 1 provides details about the various acoustic covert channels discussed.

Electromagnetic radiation (EMR) is a form of energy that is emitted from certain electronic components. EMR consists of electromagnetic (EM) waves that propagate through space in a radiant manner. Put very simply, wireless communication is based on the transmission and reception of these electromagnetic waves between a transmitter and receiver, where the waves are modulated to carry information. In many cases, electronics, such as wiring, computer monitors, video cards, and communication cables, emit EMR in the radio frequency spectrum. In some cases, these casual emissions can be modulated to carry information to other nearby receivers.

AM and FM radio frequencies. Computer screens receive images from the graphics card continuously through the video cable. The signal strength passed through the video cable is determined by the image presented on the screen. The current flow through the metal wires causes the video cable to emit electromagnetic radiation where the cable acts like an antenna. In 1998, Kuhn and Anderson released the first publications related to TEMPEST,²⁶ demonstrating that

EMR originating from a graphics card of a desktop computer can be manipulated by appropriate software to produce controllable AM radio transmissions. They showed that whenever specially generated images are displayed on the screen, AM radio signals are emitted from the video cable. The basic idea is the pattern of pixels on the screen influences the frequency and amplitude of the electromagnetic waves. By intentionally generating images with specially calculated patterns and displaying it on the screen, the required AM signals are emitted from the video cable.

In 2001, Thiele provided an open source program dubbed "TEMPEST for Eliza,"⁴¹ utilizing the computer monitor to transmit radio signals at AM radio frequencies modulated with specific audio tones. In his demonstration, the basic music of Mozart's "Letter for Alice" was modulated over AM radio. The signals generated can be heard by listening to a cheap radio receiver placed in the same room.

More than a decade later, in 2014, Guri et al. introduced a new type of attack utilizing TEMPEST to exfiltrate data from air-gapped computers.^{17,19} The malware, called, "AirHopper," bridges the air-gap between an isolated network and nearby infected mobile phones using FM signals. During the attack, the malware within the air-gapped network starts the exfiltration of sensitive data such as keylogging, passwords, and encryption keys. This sensitive data is transferred to a nearby mobile phone over FM radio signals intentionally emitted from the screen cable. AirHopper is capable of transmitting up to 60 bytes per second to a mobile phone located seven meters away from the leaking computer. A demonstration video of AirHopper can be viewed online.^c

Cellular frequencies. Smartphones with Wi-Fi, Bluetooth, and FM receivers might be physically banned from classified or sensitive areas of an organization. However, in many cases simple mobile devices with limited capabilities are not considered a threat and are hence permitted in secured facilities. In 2015, researchers introduced a malware that can turn an ordinary PC into a cellular transmitting antenna.¹⁶ The malware, codenamed "GSMem," transmits electromagnetic

signals at cellular (GSM, UMTS, and LTE) frequencies by invoking specific memory-related CPU instructions. The researchers showed that transmitted signals can be intercepted by a nearby low-end, GSM mobile phone. A demonstration video of GSMem can be found online.^d

Other techniques. SAVAT (Signal Available to the Attacker) presents a new metric that measures the electromagnetic signal created during execution of a program.⁶ The researchers observed that each basic instruction consumes a slightly different amount of voltage when executed in the CPU. The voltage fluctuations create EMR that can be captured some distance away from the computer. Programs on the computer can generate electromagnetic signals from the CPU by alternating between pairs of instructions. The attacker can measure the EMR levels over an air-gap and utilize the SAVAT metrics in order to distinguish between "0" and "1" and decode the exfiltrated data.

Funtenna, introduced in 2015, is malware that intentionally causes compromising emanation from embedded devices.⁴ Its researchers describe it as a software payload that intentionally causes its host hardware to act as an improvised RF transmitter using existing hardware that is typically not designed for electromagnetic emanation. The method exploits the output pins (GPIO) commonly seen in embedded systems in order to create EMR at a range of 10Mhz to 5Mhz. Data encoded over the emission can be intercepted remotely by an attacker with an RF receiver and antenna. A demonstration video of Funtenna can be viewed online.^e

In 2013, the NSA catalog leaked by Edward Snowden, exposed Cottonmouth, a tool that allows air-gap communication with a host software, over a USB dongle implanted with an RF transmitter and receiver.³⁷ The USBee malware¹⁸ presented in 2016 can be seen as an improvement of the Cottonmouth tool. USBee can render an unmodified USB connector into a RF transmitter utilizing just software, by the generation of controlled electromagnetic emissions from its data bus. Using this technique, one can leak information from an air-gapped computer to a simple receiver located over nine meters away.

a https://www.youtube.com/watch?v=v2_sZifZkDQ

b <https://www.youtube.com/watch?v=H7lQXmSLiP8>

c <https://www.youtube.com/watch?v=2OzTWiG1rM>

d <https://www.youtube.com/watch?v=RChj7Mg3rC4>

e <https://www.youtube.com/watch?v=1H1Lv9DAJPg>

A demonstration video of USBee can be viewed online.^f Table 2 provides details about the various electromagnetic covert channels discussed.

Thermal. More recently, heat emission for air-gap communication has also been proposed. BitWhisper, introduced in 2015, uses heat to transfer data between two adjacent computers.²⁰ A typical scenario consists of adjacent computers from two different networks, one of which is connected to the Internet, while the other is air-gapped. The basic idea is to establish a bi-directional communication channel over thermal manipulation (Figure 2). The transmitter computer intentionally emits heat for a specified amount of time (for example, by performing intensive calculations). The receiver computer uses the standard motherboard temperature sensors to measure the environmental temperature changes. Binary data can be modulated over the heat fluctuation to establish a link between the two air-gapped computers. A video demonstrating BitWhisper can be viewed online.^g

Using the same techniques, researchers have shown that attackers can broadcast messages to a group of computers located in the same room or building, by infecting the air conditioning control systems with malicious code.³² These systems are commonly connected to the Internet for purposes of remote control and monitoring. The attacker can regulate the temperature, while encoding binary information over the temperature changes. Computers can monitor the temperature changes in the room and decode the covert broadcast messages.³ Covert thermal channels between two isolated cores in the same computer case are discussed in Bartoloni.³ It has been shown that two neighboring cores on the same server platform can communicate at a speed of 12.5bit/sec. Table 3 provides details about the various thermal covert channels discussed.

Optical. Using optical emanation as a covert communication channel was also proposed in various forms. In the general form, two components are involved in the covert channel: a light-emitting source that exfiltrates the information and a remote camera that records the optic signals.

f <https://www.youtube.com/watch?v=E28V1t-k8Hk>
g <https://www.youtube.com/watch?v=EWRk51oB-1Y>

Table 1. Acoustic/ultrasonic air-gap covert channels.

Method	Transmitter	Receiver	Distance (max)	Bandwidth
Sonic ²⁹	Speaker		~30 m	20 bit/sec
Ultrasonic ^{8,24,27,36}	Speaker	PC microphone, smartphone, laptop,	~20 m	20 bit/sec
Fansmitter ²¹	Computer fans (CPU, chassis)	recording device, among others.	8 m	0.3 bit/sec
DiskFiltration ²²	HDD actuator arm		2 m	3 bit/sec

Table 2. Electromagnetic air-gap covert channels

Method	Transmitter	Receiver	Distance	Bandwidth
TEMPEST (AM) ^{26,41}	Video card, VGA cable		~30 m	?
AirHopper ^{17,19}	Video card, video cable	RF receiver, smartphone,	~8 m	480 bit/sec
GSMem ¹⁶	CPU-RAM Bus	baseband processor	5 m+	1-2 bit/sec
FUNTENNA ⁴	GPIO	FM radio	a few meters	?
SAVAT ⁶	CPU		1 m	?
USBee ¹⁸	USB data Bus		9 m+	640 bit/sec

Figure 2. An exchange of 'thermal pings' between two air-gapped computers.

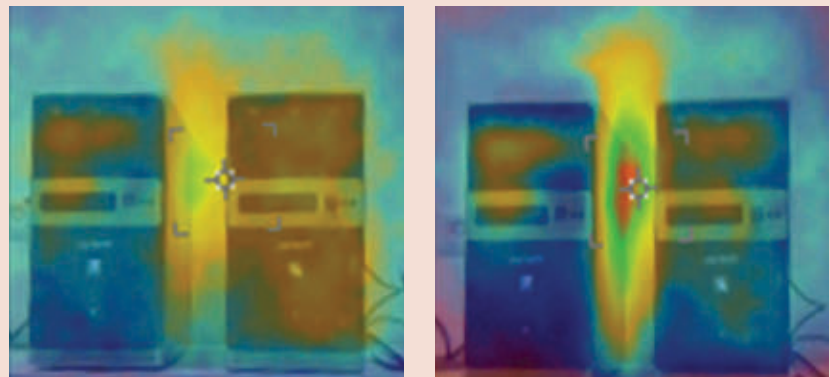


Table 3. Thermal air-gap covert channels.

Method	Transmitter	Receiver	Distance	Bandwidth
BitWhisper ²⁰	CPU/GPU	Thermal sensors	~40 cm	0.13 bit/sec
Air Conditioning ³²	HVAC	in motherboard, chassis,	room	0.83 bit/sec
Multi-Core ³	CPU	CPU and HDD	< 1 cm	12.5 bit/sec

LEDs. Loughry and Umphress²⁸ built a malware that manipulates the LED indicators of keyboards to encode sensitive information. They found that activity on a single keyboard's LED can take place at a speed of 150bit/sec. Alternatively, two or even three LEDs could be used in parallel, increasing the bandwidth of the covert channel to approximately 450bit/sec. An attacker with a line of sight to the keyboard can record the LED activity using a high-speed video camera. In another at-

tack proposed by Stepansky et al, the on/off LED indicator of a computer screen is used to exfiltrate information.³⁹ Using this technique, the data can be leaked at a bandwidth of 25bit/sec. In 2017, Guri et al. demonstrated how data can be leaked from air-gapped computers by controlling the blinks of the hard drive activity LED.^{23,h} They achieved a maximum bit rate of 4000 bits per second—a blinking

h <https://www.youtube.com/watch?v=4vlu8ld68fc>

rate that exceeds the visual perception capabilities of humans. Note that some LEDs (for example, routers and hard drive LEDs) routinely flicker, and therefore the user may not be suspicious of changes in their behavior.


Covert optical methods. A unique infiltration attack proposed in 2015 by Shamir et al. demonstrated how to establish a covert channel with a malware over the air-gap using a standard all-in-one printer.³⁶ In this case, a remote beam of blue laser blinked information in binary code; the laser was sent to the target building (aimed at a room in the building housing an all-in-one-printer) from a distance greater than one kilometer away. Malware located within the air-gapped network utilized the scanner sensors to receive the signals. The malware could also send out signals by turning the scanner lamp on and off to encode binary data. The researchers demonstrated how a drone with a laser beam and camera positioned outside a window could perform the transmission and reception tasks successfully.

VisiSploit, which was introduced in 2016, is a stealthy optical covert channel.¹⁵ This method exploits the limitations of human visual perception in order to leak sensitive information through the computer's LCD screen. A malware in the compromised computer conceals sensitive information and embeds it on the screen image in a covert manner (for example, by fast blinking), invisible and unbeknownst to the user. This research further demonstrated that an attacker was able to reconstruct the concealed data using a photo taken by a hidden camera located a distance of eight meters away. Table 4 provides details about the various optical covert channels discussed.


Attack Metrics

Most of the air-gap related covert channels have been demonstrated in experimental environments in research laboratories. However, in addition to considering a method's theoretical feasibility, it is important to examine its practical applicability and the likelihood that it may occur in a real cyberattack. Here, we examine six characteristics related to the relevance of such covert channels in realistic attack scenarios.

Stealth. There are two aspects regarding the stealth of the attack, and both



In addition to considering a method's theoretical feasibility, it is important to examine its practical applicability and the likelihood that it may occur in a real cyber attack.



are related to detection. The first is the method's ability to evade detection by *software*. If the malicious code consumes CPU at high levels, uses special system calls, or utilizes unique resources, it might be easier to detect by anti-virus or host intrusion detection products. The second aspect is the ability to evade detection by *humans* (for example, people in the room). Naturally, some optical and thermal methods can be sensed by people, and hence are more likely to be noticed during the workday, while electromagnetic and ultrasonic methods are considered more covert.

Channel availability. Another characteristic is the communication channel's level of availability during the day. In some methods, transmission or reception is available only when the computer is idle or the workload is low. This is particularly relevant to EMc methods such as AirHopper and SAVAT. Optical attacks might, in practice, only be used when there is no user in the room (for example, data exfiltration via blinking LEDs).

Virtualization and cloud environment. Modern IT environments may consist of personal workstations and servers running on top of virtual machines (VMs). Electromagnetic methods depend on precise timing of the CPU and GPU, which may be disrupted when multiple VMs are running on the same physical machine. In addition, in situations in which malware is executed in a VM, it may have no access to the system resources enabling the covert-channel. For instance, acoustic methods require access to the audio system that can be disabled in the VMs.

Hardware availability. Methods for bridging the air-gap have been proposed since the 1990s. Given this, some of the attacks have been conducted on hardware that has since become outdated. The TEMPEST-AM relay attack was conducted on CRT monitors and VGA connectors and is less relevant in today's environment. In contrast, other attacks such as GSMem, Funtenna, and USBee utilize components that are an indispensable component of modern systems. Ultrasonic channels require speakers and microphones, which might not be available in all setups. Notably, thermal sensors, and CPUs and GPUs (heat emitters) exist in every system, making thermal attacks relevant to nearly all off-the-shelf computers.

Channel quality. Another measure is the quality of the communication channel. Most electromagnetic methods suffer from erratic or low signal quality, which directly affects the bandwidth and effective distance. Attacks such as AirHopper and GSMem are prone to interruptions due to the transitional nature of the receiver: whenever the user carrying the mobile phone moves, signals may be interrupted. In the same manner, the acoustic channel is interrupted by environmental noises. Finally, thermal channels are affected by changes in the environmental temperature.

Required privileges. The covert channel may operate on the system with ordinary user privileges, or it might require an administrator or root privileges. Attacks that require root privileges are more challenging to conduct, since gaining high privileges requires exploiting special vulnerabilities in the system (for example, zero-day) without triggering IDS and AV systems.

Table 5 presents the six characteristics of covert channels (rows), and the four main types of channels (columns), and indicates for each pair, the level of feasibility and challenge they pose as real threats.

Countermeasures

Defensive countermeasures for air-gap covert-channel threats can be categorized as follows: physical insulation and red/black separation; hardware based countermeasures; and, software-based countermeasures.

Physical insulation and red/black separation. Many of the U.S. and NATO standards concerning air-gap and TEMPEST related threats are classified. Over the years, some of the standards have been declassified, but even these were released in a heavily redacted form.³¹ The prevailing standards such as NATO SDIP-27 are aimed at limiting the level of information-bearing signals and maintaining a certain distance from possible eavesdroppers. Red/black terminology, adopted from NSA jargon, refers to a physical separation between systems that may carry classified information in plain form (red) and encrypted form (black). In this way, certified equipment is classified by *zones* that refer to the perimeter that must be controlled to prevent signal leakage. For example, as a countermeasure against the electromagnetic and

acoustic attacks examined in this article, the zones approach would be used to define the physical areas inside the organization in which carrying a mobile phone or other type of radio and audio receiver is prohibited.

Hardware-based countermeasures. An essential hardware-based countermeasure scheme involves shielding devices and wires with metallic materials to prevent electromagnetic radiation from leaking out of the shielded equipment.²

Shielding can limit the effective range of many electromagnetic-based attacks. However, shielding is less suitable for internal computer components (for example, CPU and memory). Another approach is to limit the emitted signals by inserting signal filters into communication and interface cables. Such filters can block signals outside a specified frequency range, preventing unwanted electromagnetic radiation. Signal jamming is another countermeasure technique aim

Table 4. Optical air-gap covert channels.

Method	Transmitter	Receiver	Distance	Bandwidth
LEDs ^{23,28,39}	Keyboard, Hard-drive and screen LEDs	cameras (remote camera hidden camera, wearable camera, smartphone camera)	line of sight	150 bit/sec
VisiSploit ¹⁵	LCD Screen	wearable camera, smartphone camera)	~8 m	20 bit/sec
Laser, scanners, and drones ³⁶	Laser	scanner, cameras	1.2 km	20 bit/sec

Table 5. Main characteristics of covert channels by channel type.

Channel Characteristic	Channel Type			
	Acoustic	Electro	Distance	Bandwidth
Stealth	High	High	Medium (sensible)	Low/High (for example, Guri et al. ²³ or during the night)
Channel Availability	High	High	Low (overnight attack)	Low (user absence)
Feasibility in Virtualization	Medium	Medium	Medium	Medium
Hardware Availability	Medium-low	High	High	High
Quality	Medium	Medium/Low	Low	Medium
Required Privileges	Regular	Regular/Root	Regular/Root	Regular

Table 6. Types of countermeasures.

Method	Type	Relevancy to	Cost
Physical insulation, Zoning	Physical countermeasures	Acoustic, Electromagnetic, Thermal, Optical	High
Red/Black separation	Physical countermeasures	Acoustic, Electromagnetic, Thermal, Optical	High
NATO TEMPEST standards	Physical/Hardware countermeasures	Acoustic, Electromagnetic, Thermal, Optical	High
Wires and equipment shielding	Hardware countermeasures	Electromagnetic (partial)	Low-Medium
Signal filtering	Hardware countermeasures	Acoustic, Electromagnetic (partial)	Medium
Signal jamming	Hardware countermeasures	Electromagnetic, Acoustic	Medium
Activity detection	Software countermeasures	Acoustic, Electromagnetic, Thermal, Optical	Low-Medium
Soft tempest	Software countermeasures	Electromagnetic	Low

at overriding electromagnetic or acoustic signals at specified frequencies. In this method, a specialized hardware transmitter continuously generates random electromagnetic or acoustic noises that overlay other transmissions in the area.

Software-based countermeasures.

Anti-virus and behavioral detection techniques may be used to detect and block covert channel activities. For example, it is possible to monitor the program running in order to identify intentional electromagnetic, acoustic, thermal, or optic transmissions. In this case, behavioral analysis, machine learning, or anomaly detection may be used to alert on suspicious processes. Kuhn and Anderson proposed the “soft tempest” technique, an interesting software-based solution for electromagnetic attacks. The general idea is to filter out, at a software level, the information that is causing the component (for example, video cable) to emanate RF signals. The different types of countermeasures along with their relevancy to different types of covert channels and cost are provided in Table 6.

Conclusion and Outlook

Air-gap isolation is currently used in a wide range of industries and organizations. Although the exfiltration of information from air-gapped networks is still considered a challenging task, it is no longer dismissed as a sensational anecdote, as the last decade has shown that nothing is keddible for hackers. Over the years, a wide range of covert channels have been revealed that demonstrate the feasibility of data leakage by malware, despite a lack of network connection. These methods exploit the electromagnetic, acoustic, thermal, and optical emanation from various system components.

Three factors make air-gap isolation vulnerable to attacks. RF technologies have dramatically improved, allowing attackers to acquire high-quality RF receivers, audio recording devices, and remote cameras at affordable prices. This, coupled with emerging trends of multisensors, smartphones, HD cameras, versatile drones, and wearable devices, make the modern IT environment a source rich in potential covert communication channels. Finally, cyber security threats continuously develop, with hackers constantly raising the bar with sophisticated attack campaigns and innovative ways of achieving their goals. In the

future, we expect to see the emergence of new types of covert channels that challenge air-gap security, making this threat an interesting topic for academia and the cyber security community. ■

References

- Air Gap Computer Network Security; <http://abclegaldocs.com/blog-Colorado-Notary/air-gap-computer-network-security/>.
- Anderson, R.J. Emission security. *Security Engineering*, 2nd Ed. Wiley Publishing, 2008, 523–546.
- Bartolini, D.B., Miedt, P. and Thiele, L. On the capacity of thermal covert channels in multicores. *EuroSys*, 2016.
- Black-Hat. Emanate like a boss: Generalized covert data exfiltration with Funtenna. (2015); <https://www.blackhat.com/us15/briefings.html#emanate-like-a-boss-generalized-covert-data-exfiltration-with-funtenna>.
- Bornstein, M.H. and Lamb, M.E. *Cognitive Development: An Advanced Textbook*. Psychology Press, 2011.
- Callan, R., Zajic, A. and Prvulovic, M. A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 2014, 242–254.
- Carrara, B. and Adams, C. Out-of-band covert channels—A survey. *ACM Computing Surveys* 49, 2, (2016).
- Deshotels, L. Inaudible sound as a covert channel in mobile devices. In *Proceedings of the USENIX Workshop for Offensive Technologies*, 2014.
- Do, Q., Martini, B. and Choo, K-K.R. Exfiltrating data from Android devices. *Computers & Security* 48 (2015), 74–91.
- Do, Q., Martini, B. and Choo, K-K.R. A data exfiltration and remote exploitation attack on consumer 3D printers. *IEEE Trans. Information Forensics and Security* 11, 10 (2016), 2174–2186.
- D’Orazio, C.J., Choo, K-K.R. and Yang, L.T. Data exfiltration from Internet of Things devices: iOS devices as case studies. *IEEE Internet of Things J.* 99, 2327–4662.
- Federation of American Scientists. Joint Worldwide Intelligence Communications System, 1999; <http://fas.org/irp/program/dissminate/jwics.htm>.
- Goodin, D. Meet ‘badBIOS’, the mysterious Mac and PC malware that jumps airgaps. 2013; <http://arstechnica.com/security/2013/10/meet-badbios-the-mysterious-mac-and-pc-malware-that-jumps-airgaps/>.
- Goodin, D. How ‘omnipotent’ hackers tied to NSA hid for 14 years—and were found at last. 2015; <https://arstechnica.com/information-technology/2015/02/how-omnipotent-hackers-tied-to-the-nsa-hid-for-14-years-and-were-found-at-last/>.
- Guri, M., Hasson, O., Kedma, G. and Elovici, Y. An optical covert-channel to leak data through an air-gap. In *Proceedings of the 14th Annual Conference on Privacy, Security and Trust* (Auckland, 2016).
- Guri, M., Kachlon, A., Hasson, O., Kedma, G., Mirsky, Y. and Elovici, Y. GSMem: Data exfiltration from air-gapped computers over GSM frequencies. In *Proceedings of the USENIX Security Symposium*, (Washington, D.C., 2015).
- Guri, M., Kedma, G., Kachlon, A. and Elovici, Y. AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies. In *Proceedings of the 9th International Conference on Malicious and Unwanted Software: The Americas*. IEEE, 2014, 58–67.
- Guri, M., Monitz, M. and Elovici, Y. USBee: Air-gap covert-channel via electromagnetic emission from USB. In *Proceedings of the 14th Annual Conference on Privacy, Security and Trust*. (Auckland, 2016).
- Guri, M., Monitz, M. and Elovici, Y. Bridging the air gap between isolated networks and mobile phones in a practical cyber-attack. *ACM Trans. Intelligent Systems and Technology* 8, 4 (2017), 50.
- Guri, M., Monitz, Mirski, M. and Elovici, Y. BitWhisper: Covert signaling channel between air-gapped computers using thermal manipulations. In *Proceedings of the 28th IEEE Computer Security Foundations Symposium*, (Verona, 2015).
- Guri, M., Solewicz, Y., Daidakulov, A. and Elovici, Y. Fansmitter: Acoustic data exfiltration from (speakerless) air-gapped computers. 2016, arXiv:1606.05915.
- Guri, M., Solewicz, Y., Daidakulov, A. and Elovici, Y. Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise (‘DiskFiltration’). In *Proceedings of the European Symposium on Research in Computer Security*, (Oslo, 2017).
- Guri, M., Zadov, B. and Elovici, Y. LED-it-GO: Leaking (a lot of) data from air-gapped computers via the (small) hard drive LED. In *Proceedings of the 14th International Conference on Detection of Intrusions and Malware and Vulnerability Assessment*. (Bonn, 2017).
- Hanspach, M. and Goetz, M. On covert acoustical mesh networks in air. 2014; arXiv:1406.1213, 2014.
- Kuhn, M. Optical time-domain eavesdropping risks of CRT displays. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.
- Kuhn, M.G. and Anderson, R.J. Soft TEMPEST: Hidden data transmission using electromagnetic emanations. *Information Hiding*, Springer-Verlag, 1998, 124–142.
- Lee, E., Kim, H. and Yoon, J.W. Attack, various threat models to circumvent air-gapped systems for preventing network. *Information Security Applications 9503* (2015), 187–199.
- Loughry, J. and Umphress, D.A. Information leakage from optical emanations. *ACM Trans. Information and System Security* (2002), 262–289.
- Madhavapeddy, A., Sharp, R., Scott, D. and Tse, A. Audio networking: The forgotten wireless technology. *IEEE Pervasive Computing* 4, 3 (2005), 55–60.
- McAfee. Defending critical infrastructure without air gaps and stopgap security, 2015; <https://blogs.mcafee.com/executeive-perspectives/defending-critical-infrastructure-without-air-gaps-stopgap-security/>.
- McNamara, J. The complete, unofficial TEMPEST information page, 1999; <http://www.jammed.com/~jwa/tempest.html>.
- Mirsky, Y., Guri, M. and Elovici, Y. HVACKer: Bridging the air-gap by manipulating the environment temperature. deepsec, 2015.
- National Computer Security Center. NCSC-TG-004 Glossary of Computer Security Terms, 1988; <http://fas.org/irp/nsa/rainbow/tg004.htm>.
- NSA/CSS. NSA/CSS Regulation 90-6: Technical Security Program. Fort George G. Meade, MD. Partially declassified transcript, 1999; <http://cryptome.org/nsa-reg90-6.htm>.
- O’Malley, S.J. and Choo, K-K.R. Bridging the air gap: Inaudible data exfiltration by insiders. In *Proceedings of the Americas Conference on Information Systems*, 2014.
- SC Magazine. Light-based printer attack overcomes air-gapped computer security, 2014; <http://www.scmagazineuk.com/light-based-printer-attack-overcomes-air-gapped-computer-security/article/377837/>.
- Schneier, B. Schneier on Security: COTTONMOUTH-III: NSA exploit of the day; <https://www.schneier.com/blog/archives/2014/03/cottonmouth-iii.html>.
- Securelist. Agent.btz: A Source of inspiration? 2014; <https://securelist.com/blog/virus-watch/58551/agent-btz-a-source-of-inspiration/>.
- Sepetintsky, V., Guri, M. and Elovici, Y. Exfiltration of information from air-gapped machines using monitor’s LED indicator. In *Proceedings of the Intelligence and Security Informatics Conference*, (The Hague, The Netherlands, 2014).
- Symantec. Mind the gap: Are air-gapped systems safe from breaches? 2014; <http://www.symantec.com/connect/blogs/mind-gap-are-air-gapped-systems-safe-breaches>.
- Tempest for Eliza; <http://www.erikyyy.de/tempest/>.
- van Eck, W. Electromagnetic radiation from video display units, 1985; <https://cryptome.org/emr.pdf>.
- The Washington Post. Powerful NSA hacking tools have been revealed online; https://www.washingtonpost.com/world/national-security/powerful-nsa-hacking-tools-have-been-revealed-online/2016/08/16/bce4f974-63c7-11e6-96c0-37533479f3f5_story.html.
- Zander, S., Armitage, G. and Branch, P. A survey of covert channels and countermeasures in computer network protocols. *IEEE Communications Surveys & Tutorials* 9, 3 (2007), 44–57.

Mordechai Guri (gurim@post.bgu.ac.il) is head of R&D of the Cyber Security Research Labs at Ben-Gurion University of the Negev, Beer-Sheva, Israel.

Yuval Elovici (elovici@bgu.ac.il) is a professor in the Department of Information Systems Engineering and director of Deutsche Telekom Laboratories at Ben-Gurion University of the Negev, Beer-Sheva, Israel.

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

research highlights

P. 84

**Technical
Perspective
Expressive
Probabilistic
Models and Scalable
Method of Moments**

By David M. Blei

P. 85

Learning Topic Models — Provably and Efficiently

By Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno,
Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu

Technical Perspective

Expressive Probabilistic Models and Scalable Method of Moments

By David M. Blei

ACROSS DIVERSE FIELDS, investigators face problems and opportunities involving data. Scientists, scholars, engineers, and other analysts seek new methods to ingest data, extract salient patterns, and then use the results for prediction and understanding. These methods come from machine learning (ML), which is quickly becoming core to modern technological systems, modern scientific workflow, and modern approaches to understanding data.

The classical approach to solving a problem with ML follows the “cookbook” approach, one where the scientist shoehorns her data and problem to match the inputs and outputs of a reliable ML method. This strategy has been successful in many domains—examples include spam filtering, speech recognition, and movie recommendation—but it can only take us so far. The cookbook focuses on prediction at the expense of explanation, and thus values generic and flexible methods. In contrast, many modern ML applications require interpretable methods that both form good predictions and suggest good reasons for them. Further, as data becomes more complex and ML problems become more varied, it becomes more difficult to shoehorn our diverse problems into a simple ML set-up.

An alternative to the cookbook is probabilistic modeling, an approach to ML with roots in Bayesian statistics. Probabilistic modeling gives an expressive language for the researcher to express assumptions about the data and goals in data analysis. It provides a suite of algorithms for computing with data under those assumptions and a framework with which to use the results of that computation. Probabilistic modeling allows researchers to marry their knowledge and their data, developing ML methods tailored to their specific goals.

The following paper is about probabilistic topic models, a class of probabilistic models used to analyze text data. Topic modeling algorithms ingest large


collections of documents and seek to uncover the hidden thematic structures that pervade them. What is special about topic modeling is it uncovers the structure without pre-labeled documents. For example, when applied to a large collection of news articles, a topic-modeling algorithm will discover interpretable topics—represented as patterns of vocabulary words—such as sports, health, or arts. These discovered topics have many applications: summarizing the collection, forming predictions about new documents, extending search engines, organizing an interface into the collection, or augmenting recommendation systems. Topic models have further been adapted to other domains, such as computer vision, user behavior data, and population genetics, and have been extended in many other ways. There is a deluge of unlabeled text data in many fields; topic models have seen wide application in academia and industry.

A topic model assumes a random process by which unknown topics combine to generate documents. When we fit a topic model, we try to discover the particular topics that combined to form an observed collection. I emphasize that a topic model is a special case of a probabilistic model. Generally, probabilistic modeling specifies a random process that uses unobserved variables (such as topics) to generate data; the central algorithmic problem for probabilistic models is to find the hidden quantities that were likely to have generated the observations under study. What makes this problem hard, for topic models and other models, is that the models that accurately express our domain knowledge are complicated and the data sets we want to fit them to are large. The authors developed a new method for fitting topic models and at large scale.

The typical approach to solving the topic-modeling problem is to fit the topics with approximate Bayesian methods or maximum likelihood methods. (The authors here call these “likelihood-

based” methods.) The solution here is different in that the authors use what is called the method of moments. What this means is that they derive average functions of the data that a topic model would generate if it were the true model. They then calculate these average quantities on the observed documents and derive an algorithm to find the particular topics that produce them. Their algorithms scale to large datasets.

The authors prove theoretical guarantees about their algorithm. They make realistic assumptions about text (the “anchor word” assumption of topics) and assume that the data comes from a topic model. They show that, with enough documents, their algorithm—which involves their selection of the quantities to match and the algorithm to match them—finds the topics that generated the data. This is a significant result. Such guarantees have not been proved for likelihood-based methods, like Markov chain Monte Carlo (MCMC), variational Bayes, or variational expectation maximization. More generally, the paper represents an elegant blend of theoretical computer science and probabilistic machine learning.

Finally, I will posit the main question that came to me as I read the paper. The traditional methods in probabilistic machine learning, MCMC and variational Bayes (VB), provide convenient recipes for fitting a wide class of models. In contrast, much of the analysis and mathematical work that goes into method-of-moments solutions is model-specific. Is it possible to generalize method-of-moments for latent variable models so that it is as easy to derive and use as MCMC and VB? Can we generalize to other topic models? How about other graphical models? Are there guidelines for proving theoretical guarantees for other models? 

David M. Blei is a professor of statistics and computer science at Columbia University, New York City, NY, USA.

Copyright held by author.

Learning Topic Models — Provably and Efficiently

By Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu

1. INTRODUCTION

Today, we have both the blessing and the curse of being overloaded with information. Never before has text been more important to how we communicate, or more easily available. But massive text streams far outstrip anyone’s ability to read. We need automated tools that can help make sense of their thematic structure, and find strands of meaning that connect documents, all without human supervision. Such methods can also help us organize and navigate large text corpora. Popular tools for this task range from Latent Semantic Analysis (LSA)⁸ which uses standard linear algebra, to *deep learning* which relies on non-convex optimization. This paper concerns *topic modeling* which posits a simple probabilistic model of how a document is generated. We give a formal description of the generative model at the end of the section, but next we will outline its important features.

Topic modeling represents each document as a *bag of words* whereby all notions of grammar and syntax are discarded, and each document is associated with its vector of word counts. The central assumption is that there is a fixed set of *topics*—numbering, say, a couple hundred—that are shared and recur in different proportions in each document. For example, a news article about legislation related to retirement accounts might be represented as a mixture of 0.7 of the topic *politics* and 0.3 of the topic *personal finance*. Furthermore, each topic induces a distribution on words in the vocabulary. Note that a word like *account* can occur in several topics: it could refer to a financial product (a bank account) or a story (a fictional account), but the probability that it is assigned would likely vary across topics. Finally, the model specifies that each document is generated by first picking its topic proportions from some distribution, and then sampling each word from the document-specific distribution on words. In the above example, each word would be picked independently from *politics* with probability 0.7 and from *personal finance* with probability 0.3. The goal in topic modeling is, when given a large enough collection of documents, to discover the underlying set of topics used to generate them. Moreover, we want algorithms that are both fast and accurate.

This generative model is a simplistic account of how documents are created. Nevertheless, for a wide range of applications in text analysis, methods based on this model do indeed recover meaningful topics. We give an example of a randomly chosen set of topics recovered by our algorithm, when run on a collection of *New York Times* articles, as shown in Figure 1. These tools have also found many applications in summarization and exploratory data analysis. In fact, the models described above are not just limited to text analysis and have been used to recover semantic structure in various biological datasets, including fMRI images of brain activity.

Figure 1. Examples of topics automatically extracted from a collection of *New York Times* articles. Each row contains words from one topic in descending order by probability.

anthrax, official, mail, letter, worker, attack
president, clinton, white_house, bush, official, bill_clinton
father, family, elian , boy, court, miami
oil, prices, percent, million, market, united_states
microsoft , company, computer, system, window, software
government, election, mexico, political, vicente_fox, president
fight, mike_tyson , round, right, million, champion
right, law, president, george_bush, senate, john_ashcroft

Variants of this model have also been used in linguistic and humanities applications. See Ref. Blei⁵ for a thorough survey.

Traditional methods for learning the parameters of a topic model are based on maximizing a *likelihood objective*. Such approaches are popular when learning the parameters of various other probabilistic models, too. However even in the case of topic models with just *two* topics, this optimization problem is *NP-hard*.⁴ At best, the approaches used in practice are known to converge to the true solution eventually but we know of no good guarantees on the running time needed to fit the parameters up to some desired precision. These gaps in our understanding are not only a theoretical issue but also a practical one: the seemingly large running times of these algorithms means that learning 1000 topics from 20 mn news articles requires a distributed algorithm and 100 dedicated computers.¹

Recently, several groups of researchers have designed new algorithms that have provable guarantees. These algorithms run in times that scale as a fixed polynomial in the number of documents and the inverse of the desired precision.^{2,4} Our primary focus is on the algorithm of Arora et al.⁴ which is based on a seemingly realistic assumption—termed *separability*—about the structure of topics. The subsequent work of Anandkumar et al.² removes this assumption, but requires that the topics are essentially uncorrelated and seems to be quite sensitive to violations of this assumption. The contribution of the present article is to show that some of these new theoretical algorithms can be adapted to yield highly practical tools for topic modeling, that compete with state of the art approximate likelihood approaches in terms of the solution quality and run in a fraction of the time. At the same time, the provable guarantees continue to hold for our simplified algorithms.

This work appeared as “A Practical Algorithm for Topic Modeling with Provable Guarantees” (Arora, Ge, Halpern, Mimno, Moitra, Sontag, Wu, Zhu) ICML 2013.

1.1. The model

Here we formally state the model we will be interested in. We will rely on these definitions for much of the discussion that follows. Let V denote the number of words in the vocabulary. Let K denote the number of topics. And let M denote the number of documents, and D denote their length. (In general, one can allow documents to be of varying lengths and one could even specify a distribution from which their length is drawn). Each of the K topics is identified with a distribution over words. We will represent these distributions as V -dimensional vectors A_1, A_2, \dots, A_K whose entries are non-negative and sum to one.

Each document d is generated by picking its topic proportions W_d from a distribution τ . The topic proportions can also be viewed as a vector, but in K -dimensions where the value in coordinate i represents the proportion of topic i present in document d . Finally, each word is independently sampled by choosing its topic $z_j \in \{1, 2, \dots, K\}$ according to W_d , and then sampling it from that topic's distribution over words $w_j \sim A_{z_j}$. We remark that this formulation is very general and includes most widely used probabilistic topic models, such as the Latent Dirichlet Allocation Model (LDA)⁷ where τ is a Dirichlet distribution, as well as subsequent extensions that allow topics to be positively or negatively correlated such as the Correlated Topic Model (CTM)⁶ where τ is a logistic Normal distribution. See Figure 2.

1.2. Likelihood-based methods

Here we expand upon some of the computational difficulties of working with likelihood-based methods. The traditional approach to fitting the parameters of topic models is via *maximum likelihood estimation*, whereby we seek a set of K topics, $\{A_1, A_2, \dots, A_K\}$, as well as a description of the distribution τ , that maximize the likelihood of the entire collection having been generated by the model. This is a difficult optimization problem because the likelihood objective is non-convex, with many local maxima. Optimizing non-convex functions is notoriously difficult, and standard local-search based techniques like Expectation-Maximization⁹ or gradient ascent are only guaranteed to converge to a local maximum, which may be much worse in terms of the objective value than the global optimum. Even worse, evaluating the likelihood function is itself difficult due to the large number of latent variables, namely the topic proportions of each document, W_d , and the topic assignments of each word, z_j . To evaluate the likelihood of even a single document requires integrating over all possible topic-proportions, a high-dimensional integral with no closed form, as well as summing over an exponential

number of possible topic assignments for the words in the document.

Other previous works attempt to solve approximate versions of the maximum likelihood problem. For example, the variational-EM approach^{7,14} maximizes an objective that lower bounds the likelihood objective, but cannot guarantee that the solution is close to the optimum of the likelihood objective itself. The Markov Chain Monte Carlo (MCMC) approach¹³ uses Markov chains tailored to generate samples from the posterior distribution of the parameters conditioned on the observed collection of documents, but suffers from well-known drawbacks: It is difficult to assess convergence, and no polynomial bounds on its mixing time are known in settings of interest. These approximations to the maximum likelihood objective are, in a sense, necessary, since recent work has shown that even for just two topics finding the maximum likelihood solution is *NP-hard*.⁴ Another reason that these methods tend to be slow in practice, is that they contain an inner loop in which they perform approximate inference, determining which topics are likely present in each document in the collection. This is also known to be *NP-hard*.²⁵ Thus, we seek a principled new approach that can circumvent both the hardness of maximum likelihood estimation and inference.

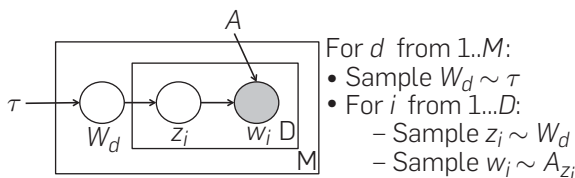
1.3. The method of moments

The challenges of working with the maximum likelihood estimator motivate us to investigate other consistent estimators, ones that hopefully can be computed more efficiently. As we mentioned earlier, we build on recent algorithms^{2,4} that provably recover the parameters of a topic model in polynomial time. These approaches are based on the method of moments—which is originally due to Pearson²³—but has fallen out of favor in the statistics community in large part because it seems to require more samples than the likelihood-based approaches championed by Fisher. However, in the modern age of big data, statistical efficiency is not as pressing an issue as computational efficiency. With this in mind, it seems time to revisit the method of moments.

The key concept behind the method of moments is to set up a system of equations relating quantities that can be easily estimated from data (such as means or averages) and the parameters of model. One has to choose the set of equations carefully so as to guarantee the identifiability of model parameters and to ensure that the system of equations can be solved efficiently. In recent years, the method of moments has been used to give computationally efficient algorithms for a variety of fundamental statistical estimation problems such as learning mixtures of Gaussians.¹⁵

Let us describe the approach in the context of topic modeling, working with second order moments. Let Q be a $V \times V$ matrix where the entry $Q_{j,j'}$ denotes the probability that the first and second words in a randomly generated document are word j and word j' respectively. It turns out that this matrix can be expressed as the product of three entry-wise nonnegative matrices. Let R denote a $K \times K$ matrix where the entry $R_{i,i'}$ represents the probability that the first and second word are sampled from topic i and topic i' respectively.

Figure 2. Generative model used in topic modeling.



Finally let A denote the $V \times K$ matrix whose columns are A_1, A_2, \dots, A_K . Then it can be shown that $Q = ARA^T$. Suppose for the moment that we could accurately estimate the entries of Q .

A naive attempt to apply the method of moments runs into its own computational difficulties when one attempts to solve the system of non-linear equations. In particular, we are faced with a matrix decomposition problem where our goal is to express Q as a product of entry-wise non-negative matrices as above. This is closely related to the *nonnegative matrix factorization* problem and is known to be *NP-hard*.^{3,26} The approach of Arora et al.⁴ is to make use of algorithms that solve nonnegative matrix factorization under a certain assumption that seems natural in the context of topic modeling. We describe this assumption and its rationale next.

1.4. Separability

The guiding assumption behind the algorithm of Arora et al.⁴ is a notion called *separability*.¹¹ More precisely, this assumption stipulates that topics can be reliably distinguished from one another via *anchor words*—which, in the context of topic models, are specialized words that are specific to a single topic. For example, if the word *401k* occurs in a document then it is a strong indicator that the document is at least partially about *personal finance*. Natural language seems to contain many such unambiguous words. The condition of separability requires that each topic contains at least one (unknown) anchor word. We provide various experimental evidence showing that models fit to real-world data sets contain many anchor words.

Arora et al.³ gave an algorithm for solving nonnegative matrix factorization under the separability assumption. In a subsequent paper, Arora et al.⁴ showed that such an algorithm can be used to provably learn the parameters of a separable topic model. While theoretically important, these algorithms (as stated) were far from practical: the runtime is a large polynomial, and the algorithm itself is sensitive to violations of the modeling assumptions, learning poor quality topics when run on real-world data collections. The current paper addresses these issues, presenting a variant of the above algorithm that achieves state of the art performance and runs orders of magnitude faster than approximate likelihood based approaches. Along the way, we also give a faster algorithm for solving separable nonnegative matrix factorization.

We remark that separability is not the only assumption that allows for polynomial time recovery of topic-models. Anandkumar et al.² give a provable algorithm for topic modeling based on third-order moments and tensor decomposition that does not require separability but instead requires that topics are essentially uncorrelated. Although standard topic models like LDA⁷ assume this property, there is strong evidence that real-world topics are dependent.^{6,19} For example, the topics *economics* and *politics* are more likely to co-occur than *economics* and *cooking*.

2. THE ANCHOR WORDS ALGORITHM

2.1. From probability to geometry

Separable topic models have various important probabilistic and geometric properties. These properties will form the foundation for our algorithm. We will work with simple

statistics measuring how often various pairs of words co-occur in a document. Recall that the matrix Q denotes the co-occurrence probabilities of pairs of words. In this section it is more convenient to consider the *conditional probabilities* \bar{Q} , where $\bar{Q}_{i,j}$ is the probability of the second word being j conditioned on the first word being i . The matrix \bar{Q} is just a row normalized version of Q whose rows sum up to 1.

It is useful to consider this data geometrically. We can view the rows of \bar{Q} as points in V -dimensional space. Moreover we will call a row of \bar{Q} an *anchor row* if it corresponds to an anchor word. A simplified illustration of anchor and non-anchor rows is given in Figure 3. The key insight behind our algorithm is the following fact. Recall that a vector u is said to be in the convex hull of vectors v_1, v_2, \dots, v_d if it can be written as $u = \sum_i \lambda_i v_i$ where the λ_i 's are nonnegative and sum to one.

LEMMA 1. *If the topic matrix is separable, then each row of \bar{Q} is in the convex hull of the anchor rows.*

This geometric property motivates our simple, greedy algorithm for identifying the anchor words. First we sketch a proof of this lemma through elementary manipulations on various conditional probabilities.

Algorithm 1. FindAnchors

1: **Compute co-occurrences.** Let N_d be the length of document d , and $N_d(i)$ be the number of occurrences of word i in document d .

$$\hat{Q}_{i,j} = \frac{1}{M} \sum_d \frac{2}{N_d(N_d-1)} N_d(i)N_d(j)$$

$$\hat{Q}_{i,i} = \frac{1}{M} \sum_d \frac{2}{N_d(N_d-1)} (N_d^2(i) - N_d(i))$$

2: Let $\hat{\bar{Q}}$ be a row normalization of \hat{Q} . Rows of $\hat{\bar{Q}}$ sum up to 1.

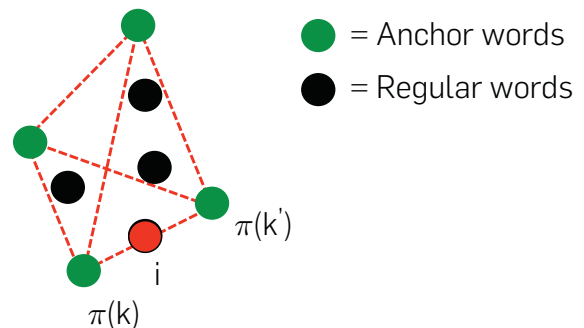
3: **for** $k = 1$ TO K **do**

4: Choose the row in $\hat{\bar{Q}}$ furthest from the affine span of the anchor rows chosen so far.

5: **end for**

6: Return the chosen anchor words

Figure 3. The rows of Q are vectors in V -dimensions, and their convex hull is a K -simplex whose vertices are anchor rows. Here $Q_i = \frac{1}{2}Q_{\pi(k)} + \frac{1}{2}Q_{\pi(k')}$ and this implies that the posterior distribution $P(z_1 = * | w_1 = i)$ assigns $\frac{1}{2}$ to $z_1 = k$ and $\frac{1}{2}$ to $z_1 = k'$ and zero to every other topic.



Consider a randomly generated document, and let w_1 and w_2 be random variables that denote its first and second words respectively. Furthermore let z_1 and z_2 denote their latent topic assignments. We will think of the generative procedure as first picking W_d from τ and then picking $z_1, z_2 \in [K]$ independently according to W_d . Once these topic assignments are fixed, the words w_1 and w_2 are independently sampled from A_{z_1} and A_{z_2} respectively. We will use the notation $\pi(k)$ to denote an anchor word for topic k . Then the definition of an anchor word gives us:

$$P(z_1 = k' | w_1 = \pi(k)) = \begin{cases} 1 & k' = k \\ 0 & \text{else} \end{cases}$$

This follows because when an anchor word is observed, there is only one topic that could have generated it! Moreover let \bar{Q}_i denote the i th row of \bar{Q} . Then the j th coordinate of \bar{Q}_i is $P(w_2 = j | w_1 = i)$. We will use the shorthand $\bar{Q}_i = P(w_2 = * | w_1 = i)$. It follows that,

$$\bar{Q}_{\pi(k)} = P(w_2 = * | w_1 = \pi(k)) = P(w_2 = * | z_1 = k)$$

And finally we can write,

$$\begin{aligned} \bar{Q}_i &= \sum_{k'} P(w_2 = * | w_1 = i, z_1 = k') P(z_1 = k' | w_1 = i) \\ &= \sum_{k'} P(w_2 = * | w_1 = \pi(k')) P(z_1 = k' | w_1 = i) \end{aligned}$$

This formula explicitly represents \bar{Q}_i as a convex combination of the anchor rows, but moreover we see that the convex combination is given by the conditional probabilities $P(z_1 = k' | w_1 = i)$ of which topic generated word $w_1 = i$. Thus our strategy is to find the anchor rows, and then solve a low-dimensional convex program to represent every non-anchor row as a convex combination of the anchor rows to find $P(z_1 = k' | w_1 = i)$. From there, we can use Bayes' rule to compute $P(w_1 = i | z_1 = k')$ which are exactly the parameters (except for the hyperparameters) of our topic model.

2.2. Finding the anchor words

We give a simple, greedy algorithm called `FindAnchors` that provably finds the K anchor words (one for each topic) given the empirical estimate \bar{Q} of the matrix Q defined in the previous subsection. We will analyze this algorithm in the noiseless setting where $\bar{Q} = Q$, but what is important about this algorithm is its behavior in the presence of noise. In that setting, it can be shown that `FindAnchors` recovers *near anchor words*—that is, words whose row in \bar{Q} is close in ℓ_1 distance to some anchor word. We need these latter types of guarantees to quantify how much data we need to get estimates that are provable close to the true parameters of the topic model.

The algorithm builds up a set of anchor words greedily, and starts by choosing the row farthest from the origin. Then it iteratively add points that maximize distance from the affine span of the previously collected points. This procedure can also be seen as iteratively growing the simplex, adding vertices that greedily maximize the enclosed volume. While the general problem of choosing K rows of a matrix Q to maximize the enclosed volume is NP -hard, it becomes

easy when the points are known to lie in a simplex and the vertices of the simplex are themselves among the input points.

For the purpose of improving the noise tolerance, we add a second “clean up” stage that iteratively removes each vertex and adds back the point farthest from the span of the remaining vertices. While this additional round of cleanup has been previously suggested as a heuristic to improve quality, in the full version of our paper we show that it also improves the theoretical guarantees of the algorithm.

Finally, the running time of this algorithm can be further improved by using random projection. Randomly projecting a collection of vectors in high dimensions onto a random low-dimensional subspace is well-known to approximately preserve the pairwise distance between each pair of vectors. And since our algorithm iteratively finding the farthest point from a subspace, its behavior is preserved after a random projection. But this refinement of the algorithm allows it to work with low-dimensional points, and improves its efficiency. The final running time is $\tilde{O}(V^2 + VK/\epsilon^2)$.

3. TOPIC RECOVERY

Here we give an algorithm called `Recover-Topics (L2)` that provably recovers the parameters of the topic model when given the anchor words. The algorithm exploits the same probabilistic and geometric properties of separable topic models, which we described earlier. Recall that every row of \bar{Q} can be (approximately) written as a convex combination of the anchor rows. Moreover, the mixing weights are very close to the probabilities $P(w_1 = i | z_1 = k)$.

Algorithm 2. Recover-Topics (L2)

- 1: **for** $i = 1$ TO V **do**
 - 2: Project row i of \bar{Q} into the convex hull of the anchor rows, and interpret the resulting convex combination as $p(z_1 = * | w_1 = i)$
 - 3: **end for**
 - 4: Solve for A using Bayes' rule, as given in Equation (1)
 - 5: Solve the linear system $\hat{Q} = ARA^T$ for R
 - 6: Return A, R
-

For each non-anchor row, our algorithm finds the point in the convex hull of the anchor rows that is closest (in Euclidean distance). This is a minimization problem that can be solved effectively using the Exponentiated Gradient algorithm.¹⁶ The resulting point can be expressed as a convex combination of the anchor rows, which yields the conditional probabilities $P(w_1 = i | z_1 = k')$ as described earlier. These values differ slightly from what we want. Ultimately, we can recover the entries of A through Bayes' rule

$$P(w_1 = i | z_1 = k) = \frac{P(z_1 = k | w_1 = i)P(w_1 = i)}{\sum_{i'} P(z_1 = k | w_1 = i')P(w_1 = i')} \quad (1)$$

Recall that $Q = ARA^T$, and since A has full column rank (because it is separable), we can solve for R by solving this linear system. Furthermore, it turns out that in the special case of the LDA

model, we can additionally recover the Dirichlet hyperparameters directly from R . We defer the details to the full version of our paper. Finally, we remark that in our algorithm, when we find the closest point in Euclidean distance this step can be “kernelized,” making the running time of each iteration of exponentiated gradient independent of the vocabulary size, V . We can solve the resulting minimization problem with a tolerance of ϵ^2 requires $K \log K/\epsilon^2$ iterations of the Exponentiated Gradient¹⁶ algorithm. The running time of Recover-Topics (L2) is $\tilde{O}(V^2K + VK^3/\epsilon^2)$ and the for-loop which constitutes the main computational bottleneck can be trivially parallelized.

Recall that in implementing Bayes’ rule, we compute for $k \in [K]$, the denominator $\sum_i p(z_1 = k | w_1 = i)p(w_1 = i) = p(z_k)$ (this is done implicitly when normalizing the columns of A' in Algorithm 2), which gives us, up to a constant scaling, the Dirichlet hyperparameters. This scaling constant can be recovered from the R matrix as described in Ref. Arora et al.,⁴ but in practice we find it better to choose this single parameter using a grid search to maximize the likelihood of the data.

3.1. Theoretical guarantees

Here we state rigorous guarantees on the sample complexity and running time of our overall algorithm. We defer the guarantees for FindAnchors and Recover-Topics (L2) themselves to later in this section. When we are given a finite set of samples, our empirical statistics—which we denote by \hat{Q} —will be a good, but imperfect approximation to \bar{Q} . In order to bound how many samples we need to obtain some target accuracy in recovering the true parameters of the topic model, we need to track the various sources of error through our algorithm.

Moreover, we need that certain parameters are bounded in reasonable ranges to guarantee that the inverse problem we are trying to solve is well-posed. Recall that the existence of anchors implies that we are trying to solve a *separable* non-negative matrix factorization problem. We characterize the separability of the problem as follows:

DEFINITION 1. *The word-topic matrix A is p -separable for $p > 0$ if for each topic k , there is some word i such that $A_{i,k} \geq p$ and $A_{i,k'} = 0$ for $k' \neq k$.*

Thus, not only should each topic have an anchor word but it should also have one that has non-negligible probability. We will require a lower bound on p , and the running time and sample complexity of our algorithm will depend polynomially on $1/p$. We will also need a second measure γ that we will use to denote the smallest singular value of R . When γ is too small, the problem of recovering A and R from $Q = ARA^T$ becomes unstable. Note that this measure also implies that no topic can have very low probability since for any topic i , it can be shown that $\gamma \leq P(z_1 = k)$. When the problem is well-behaved with respect to these two measures, our algorithm achieves the following guarantee:

THEOREM 1. *There is a polynomial time algorithm that learns the parameters of a topic model if the number of documents is at least*

$$M = \max \left\{ O \left(\frac{\log V \cdot K^6}{\epsilon^2 p^6 \gamma^6 D} \right), O \left(\frac{\log K \cdot K^4}{\gamma^4} \right) \right\},$$

where p and γ are the two non-degeneracy measures defined above and $D \geq 2$ is the length of the shortest document. The algorithm learns the word-topic matrix A and the topic-topic covariance matrix R up to additive error ϵ .

To prove this theorem, we show that the FindAnchors algorithm successfully recovers near-anchor words, and the Recover-Topics (L2) algorithm accurately estimates the desired parameters given near-anchor words. Before stating the guarantee for FindAnchors algorithm, we first introduce the following notion of α -covering. We will say

Let $\{v_1, v_2, \dots, v_K\}$ and $\{v'_1, v'_2, \dots, v'_K\}$ be two sets of points. We say that these sets of points α -cover each other

DEFINITION 2. *We say that a set of points $\{v'_1, v'_2, \dots, v'_K\}$ α -covers another set of points $\{v_1, v_2, \dots, v_K\}$, if when representing each v'_i as a convex combination $v = \sum_{k'=1}^K c_{k'} v_{k'}$, we have that $c_i \geq 1 - \alpha$.*

Clearly, we would like the anchor points to be α -covered by the set of near-anchors found by FindAnchors algorithm. Let δ be the largest perturbation between the rows of \hat{Q} and \bar{Q} , $\max_i \|\hat{Q}_i - \bar{Q}_i\| \leq \delta$. Lemma 2 connects the indices found by FindAnchors and the true anchors.

LEMMA 2. *If $\delta < (\gamma p)^3/20K$, then FindAnchors will output a set of rows that $O(\delta/\gamma p)$ -covers the true anchor rows.*

Next we show the Recover-Topics algorithm is robust to perturbations in the vertices and the internal points, making it possible to bound the error in the reconstruction coefficients in Lemma 3.

LEMMA 3. *When Recover-Topics (L2) is provided with rows which $O(\delta/\gamma p)$ -cover the true anchor rows, the element-wise error on the returned matrix A is at most $O(\delta K/\gamma^3 p^2)$.*

Combining these two lemmas, and standard concentration bounds for the empirical correlation matrix \bar{Q} , we get the guarantees in the main Theorem 1.

4. EXPERIMENTAL RESULTS

The proposed method in this article, anchor finding followed by convex optimization for topic recovery, is both faster than standard probabilistic approaches and more robust to violations of model assumptions than previous provable approaches. We compare two parameter recovery methods and a standard, probabilistically motivated algorithm. The first method is a simple matrix inversion presented in Ref. Arora et al.,⁴ which we call Recover. This inversion method is theoretically optimal, but fails in practice. The second is the constrained recovery method using a squared ℓ_2 loss, which we call RecoverL2 as shorthand for Recover-Topics (L2). As a comparison, we also consider a state-of-the-art Gibbs sampling implementation.²⁰ We would like an algorithm to be fast, accurate, and robust to noisy data. We find that the anchor-based algorithm is substantially faster than the standard algorithm, especially for large corpora. To evaluate accuracy we test the algorithms on semi-synthetic data (with known

topic distributions) and real documents. In addition, we measure the effect of different sources of error and model mismatch.

4.1. Methodology

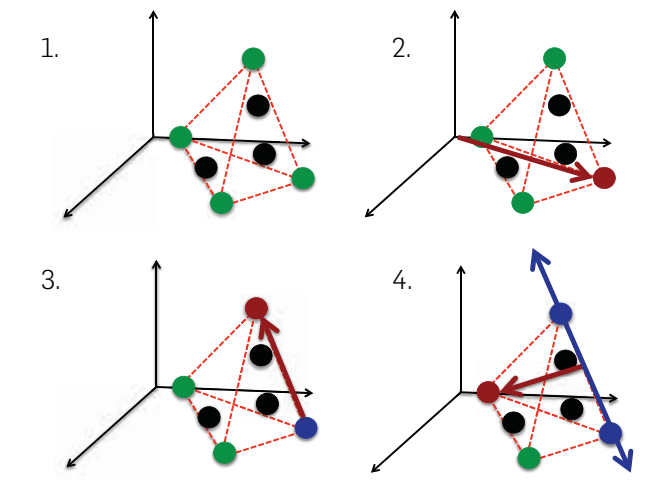
We train models on two synthetic data sets to evaluate performance when model assumptions are correct, and on real documents to evaluate real-world performance. To ensure that synthetic documents resemble the dimensionality and sparsity characteristics of real data, we generate *semi-synthetic* corpora. For each real corpus, we train a model using Gibbs sampling and then generate new documents using the parameters of that model (these parameters are *not* guaranteed to be separable; we found that about 80% of topics fitted by Gibbs sampling had anchor words).

We use two real-world data sets, a large corpus of *New York Times* articles (295k documents, vocabulary size 15k, mean document length 298) and a small corpus of Neural Information Processing Systems (NIPS) abstracts (1100 documents, vocabulary size 2500, mean length 68). Vocabularies were pruned with document frequency cut-offs. We generate semi-synthetic corpora of various sizes from models trained with $K = 100$ from *NY Times* and NIPS, with document lengths set to 300 and 70, respectively, and with document-topic distributions drawn from a Dirichlet with symmetric hyperparameters 0.03.

For the first stage of the algorithm, anchor word recovery, we use the `FindAnchors` algorithm in all cases. The original linear programming-based anchor word finding method presented with `Recover` in Arora et al.,⁴ is too slow to be comparable. For Gibbs sampling we obtain the word-topic distributions by averaging over 10 saved states, each separated by 100 iterations, after 1000 burn-in iterations.

We use a variety of metrics to evaluate the learned models. For the semi-synthetic corpora, we compute the *reconstruction error* between the true word-topic distributions and the

Figure 4. The first three steps of `FindAnchors` consist of finding a starting point furthest from the origin, finding the furthest point from the initial point, and finding the furthest point from the line defined by the first two points.



learned distributions. In particular, given a learned matrix \hat{A} and the true matrix A , we use bipartite matching to align topics, and then evaluate the ℓ_1 distance between each pair of topics. When true parameters are not available, a standard evaluation for topic models is to compute *held-out probability*, the probability of previously unseen documents under the learned model.

Topic models are useful because they provide interpretable latent dimensions. We can evaluate the *semantic quality* of individual topics using a metric called *Coherence*.²¹ This metric has been shown to correlate well with human judgments of topic quality. If we perfectly reconstruct topics, all the high-probability words in a topic should co-occur frequently, otherwise, the model may be mixing unrelated concepts. Given a set of words \mathcal{W} , coherence is

$$Coherence(\mathcal{W}) = \sum_{w_1, w_2 \in \mathcal{W}} \log \frac{D(w_1, w_2) + \epsilon}{D(w_2)}, \quad (2)$$

where $D(w)$ and $D(w_1, w_2)$ are the number of documents with at least one instance of w , and of w_1 and w_2 , respectively. We set $\epsilon = 0.01$ to avoid taking the log of zero for words that never co-occur. Coherence measures the quality of individual topics, but does not measure redundancy, so we measure *inter-topic similarity*. For each topic, we gather the set of the N most probable words. We then count how many of those words do not appear in any other topic's set of N most probable words. For these experiments we use $N = 20$. Some overlap is expected due to semantic ambiguity, but lower numbers of unique words indicate less useful models.

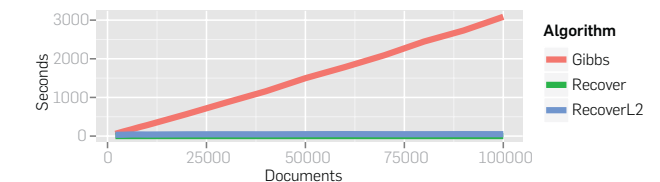
4.2. Efficiency

Both the `Recover` and `RecoverL2` algorithms, in Python, are faster than a heavily optimized Gibbs sampling implementation in Java. Figure 5 shows the time to train models on synthetic corpora on a single machine. Gibbs sampling is linear in the corpus size. `RecoverL2` is also linear ($\rho = 0.79$), but only varies from 33 to 50sec. Estimating Q is linear, but takes only 7sec for the largest corpus. `FindAnchors` takes less than 6sec for all corpora.

4.3. Semi-synthetic documents

The new algorithms have good ℓ_1 reconstruction error on semi-synthetic documents, especially for larger corpora. Results for semi-synthetic corpora drawn from topics trained on *NY Times* articles are shown in Figure 6 (top) for corpus sizes ranging from 50k to 2M synthetic documents. In addition,

Figure 5. Training time on synthetic NIPS documents.



we report results for the Recover and RecoverL2 algorithms on “infinite data,” that is, the true Q matrix from the model used to generate the documents. Error bars show variation between topics. Recover performs poorly in all but the noiseless, infinite data setting. Gibbs sampling has the lowest ℓ_1 on smaller corpora. However, for the larger corpora the new RecoverL2 algorithm have the lowest ℓ_1 error and smaller variance (running sampling longer may reduce MCMC error further). Results for semi-synthetic corpora drawn from NIPS topics are shown in Figure 6 (bottom), and are similar.

Effect of separability. Notice that as shown in Figure 6, Recover does not achieve zero ℓ_1 error even with noiseless “infinite” data. Here we show that this is due to lack of separability, and that the new recovery algorithms are more robust to violations of the separability assumption. In our semi-synthetic corpora, documents are generated from an LDA model, but the topic-word distributions are learned from data and may not satisfy the anchor words assumption. We now add a synthetic anchor word to each topic that is, by construction, unique to that topic. We assign the synthetic anchor word a probability equal to the most probable word in the original topic. This causes the distribution to sum to greater than 1.0, so we renormalize. Results are shown in Figure 7. The ℓ_1 error goes to zero for Recover, and close to zero for RecoverL2 (not zero because we do not solve to perfect optimality).

Figure 6. ℓ_1 error for learning semi-synthetic LDA models with $K = 100$ topics (top: based on *NY Times*, bottom: based on NIPS abstracts). The horizontal lines indicate the ℓ_1 error of K uniform distributions.

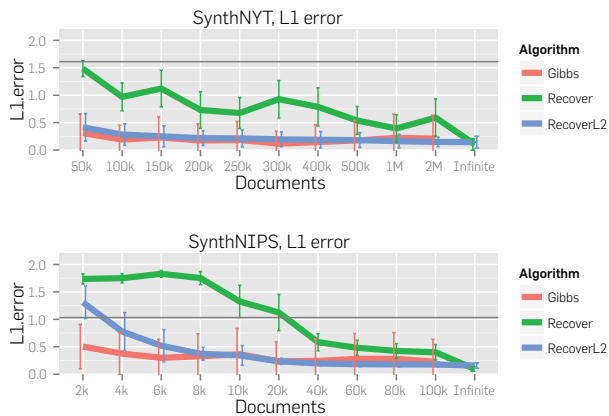
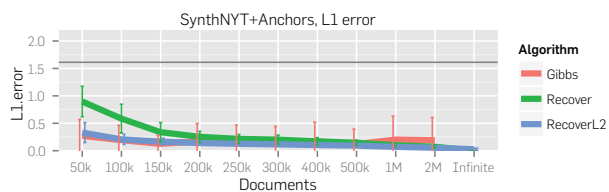


Figure 7. When we add artificial anchor words before generating synthetic documents, ℓ_1 error goes to zero for Recover and close to zero for RecoverL2.

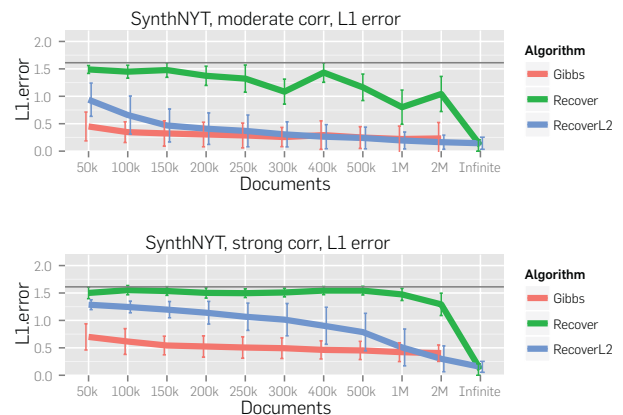


Effect of correlation. The theoretical guarantees of the new algorithms apply even if topics are correlated. To test the empirical performance in the presence of correlation, we generated new synthetic corpora from the same $K = 100$ model trained on *NY Times* articles. Instead of a symmetric Dirichlet distribution, we use a logistic Normal distribution with a block-structured covariance matrix. We partition topics into 10 groups. For each pair of topics in a group, we add a non-zero off-diagonal element (ρ) to the covariance matrix. This block structure is not necessarily realistic, but shows the effect of correlation. Results for $\rho = 0.05$ and 0.1 are shown in Figure 8. Recover performs much worse with correlated topics than with LDA-generated corpora (Figure 6). The other three algorithms, especially Gibbs sampling, are more robust to correlation. Performance consistently degrades as correlation increases. For the recovery algorithms this is due to a decrease in γ , the condition number of the R matrix. With infinite data, ℓ_1 error is equal to the ℓ_1 error in the uncorrelated synthetic corpus (non-zero because of violations of the separability assumption).

4.4. Real documents

The new algorithms produce comparable quantitative and qualitative results on real data. Figure 9 shows three metrics for both corpora. Error bars show the distribution of log probabilities across held-out *documents* (top panel) and coherence and unique words across *topics* (center and bottom panels). Held-out sets are 230 documents for NIPS and 59k for *NY Times*. For the small NIPS corpus we average over five non-overlapping train/test splits. The matrix inversion step in Recover fails for the NIPS corpus so we modify the procedure to use pseudoinverse. This modification is described in the supplementary materials. In both corpora, Recover produces noticeably worse held-out log probability per token than the other algorithms. Gibbs sampling produces the best average held-out probability ($p < 0.0001$ under a paired t -test), but the difference is within the range of variability between documents. We

Figure 8. ℓ_1 error increases as we increase topic correlation (top: $\rho = 0.05$, bottom: $\rho = 0.1$). Based on the *NY Times* semi-synthetic model with 100 topics.



tried several methods for estimating hyperparameters, but the observed differences did not change the relative performance of algorithms. Gibbs sampling has worse coherence than the other algorithms, but produces more unique words per topic. These patterns are consistent with semi-synthetic results for similarly sized corpora (details are in supplementary material).

For each *NY Times* topic learned by RecoverL2 we find the closest Gibbs topic by ℓ_1 distance. The closest, median, and farthest topic pairs are shown in Table 1. We observe that when there is a difference, recover-based topics tend to have more specific words (*Anaheim Angels* vs. *pitch*).

5. CONCLUDING REMARKS

Here we have shown that algorithms based on the separability assumption are highly practical and produce topic models of quality comparable to likelihood-based methods that use Gibbs sampling, while running in a fraction of the time. Moreover, these algorithms are particularly well-suited to parallel implementations, since each of the major steps—with the exception of finding the anchor words—can be trivially parallelized. Our algorithms inherit

Figure 9. Held-out probability (per token) is similar for RecoverL2 and Gibbs sampling. RecoverL2 has better coherence, but fewer unique terms in the top $N = 20$ words than Gibbs. (Up is better for all three metrics.)

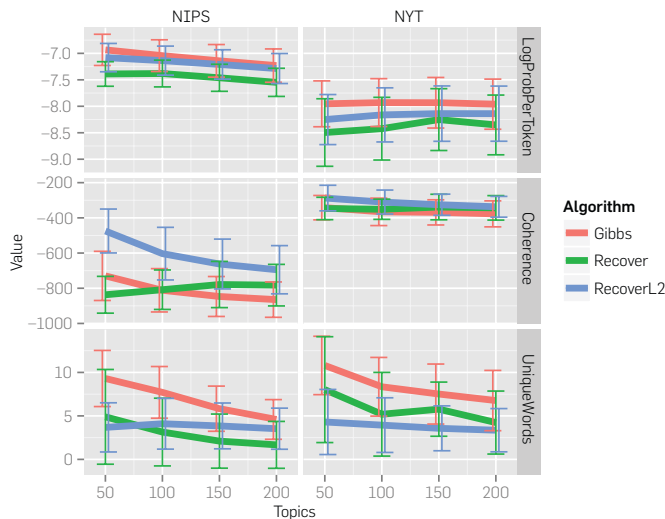


Table 1. Example topic pairs from *NY Times* (closest ℓ_1), anchor words in bold. The UCI *NY Times* corpus includes named-entity annotations, indicated by the zzz prefix. All 100 topics are shown in the supplementary material.

RecoverL2	run inning game hit season zzz_anaheim_angel
Gibbs	run inning hit game ball pitch
RecoverL2	father family zzz_elian boy court zzz_miami
Gibbs	zzz_cuba zzz_miami cuban zzz_elian boy protest
RecoverL2	file sport read internet email zzz_los_angeles
Gibbs	web site com www mail zzz_internet

provable guarantees from earlier approaches⁴ in the sense that given samples from a topic model, the estimates provably converge to the true parameters at an inverse polynomial rate. However an important question going forward is to theoretically explain why these algorithms appear to be (somewhat) robust to model misspecification. In our experiments, we fit a topic model to real data and the resulting topic model is *not* separable, but merely close to being separable. Nevertheless, our algorithms recover high quality topics in this setting too. Likelihood based methods are known to be well-behaved when the model is misspecified, and ideally one should be able to design provable algorithms that not only have good running time and sample complexity, but can also tolerate a realistic amount of noise.

Since its publication, our algorithms have been extended in a number of directions. Roberts et al.²⁴ consider applications in the social sciences and find that using an anchor-based model as an initialization for a likelihood-based algorithm reduces variability and improves model fit. Nguyen et al.²² improve the topic recovery step by adding regularization to smooth the estimated topic-word distributions, resulting in improved interpretability. A number of authors have suggested new approaches to find anchor words. Ding et al.¹⁰ present a distributed algorithm that can be parallelized across multiple servers. Zhou et al.²⁷ find anchor words by projecting rows of \bar{Q} into the plane, and selecting words that often appear as extreme points. Lee and Mimno¹⁸ replace random projections with a single heavy-tailed *t*-SNE projection that does not preserve pairwise ℓ_2 distances, but preserves local distances, allowing points to be more spread out in the projected space. Ge and Zou¹² relaxed the anchor word assumption to a subset separable assumption that can hold even when anchor words are not in a single topic, but a combination of a few topics. Other recent work¹⁷ established criteria necessary for the anchor factorization. Enforcing these criteria on the input matrix through an initial rectification step substantially improved model robustness, especially for small numbers of topics.

More broadly, the anchor words themselves have also proven to be a useful tool in summarizing the meaning of a topic and distinguishing a topic from related topics. When coupled with the right visualization and analytic tools, it may be possible to design semi-supervised learning algorithms where a domain expert helps choose the final set of anchors. It is also possible that anchor words will find applications beyond text analysis, and will enable efficient algorithms in other domains much the same way this assumption has in topic modeling. □

References

- Ahmed, A., Aly, M., Gonzalez, J., Narayanamurthy, S., Smola, A.J. Scalable inference in latent variable models. In *WSDM '12: Proceedings of the fifth ACM international conference on Web search and data mining* (New York, NY, USA, 2012), ACM, 123–132.
- Anandkumar, A., Foster, D., Hsu, D., Kakade, S., Liu, Y. Two SVDs suffice: Spectral decompositions for probabilistic topic modeling and latent dirichlet allocation. In *NIPS* (2012).
- Arora, S., Ge, R., Kannan, R., Moitra, A. Computing a nonnegative matrix factorization—Provably. In *STOC* (2012), 145–162.

4. Arora, S., Ge, R., Moitra, A. Learning topic models—Going beyond SVD. In *FOCS* (2012).
5. Blei, D. Introduction to probabilistic topic models. *Commun. ACM* (2012), 77–84.
6. Blei, D., Lafferty, J. A correlated topic model of science. *Ann. Appl. Stat.* (2007), 17–35.
7. Blei, D., Ng, A., Jordan, M. Latent dirichlet allocation. *J. Mach. Learn. Res.* (2003), 993–1022. Preliminary version in *NIPS* 2001.
8. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R. Indexing by latent semantic analysis. *JASIS* (1990), 391–407.
9. Dempster, A.P., Laird, N.M., Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B* (1977), 1–38.
10. Ding, W., Rohban, M.H., Ishwar, P., Saligrama, V. Efficient distributed topic modeling with provable guarantees. *JMLR* (2014), 167–175.
11. Donoho, D., Stodden, V. When does non-negative matrix factorization give the correct decomposition into parts? In *NIPS* (2003).
12. Ge, R., Zou, J. Intersecting faces: Non-negative matrix factorization with new guarantees. In *Proceedings of The 32nd International Conference on Machine Learning* (2015), 2295–2303.
13. Griffiths, T.L., Steyvers, M. Finding scientific topics. *Proc. Natl. Acad. Sci. USA* (2004), 5228–5235.
14. Hoffman, M.D., Blei, D.M. Structured stochastic variational inference. In *18th International Conference on Artificial Intelligence and Statistics* (2015).
15. Kalai, A.T., Moitra, A., Valiant, G. Disentangling gaussians. *Commun. ACM* 55, 2 (Feb. 2012), 113–120.
16. Kivinen, J., Warmuth, M.K. Exponentiated gradient versus gradient descent for linear predictors. *Inform. and Comput.* 132 (1995).
17. Lee, M., Bindel, D., Mimno, D.M. Robust spectral inference for joint stochastic matrix factorization. In *NIPS* (2015).
18. Lee, M., Mimno, D. Low-dimensional embeddings for interpretable anchor-based topic inference. In *EMNLP* (2014).
19. Li, W., McCallum, A. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML* (2007), 633–640.
20. McCallum, A. Mallet: A machine learning for language toolkit (2002). <http://mallet.cs.umass.edu>.
21. Mimno, D., Wallach, H., Talley, E., Leenders, M., McCallum, A. Optimizing semantic coherence in topic models. In *EMNLP* (2011).
22. Nguyen, T., Hu, Y., Boyd-Graber, J. Anchors regularized: Adding robustness and extensibility to scalable topic-modeling algorithms. In *ACL* (2014).
23. Pearson, K. Contributions to the mathematical theory of evolution. *Philos. Trans. R. Soc. Lond. A* 185 (1894), 71–110.
24. Roberts, M.E., Stewart, B.M., Tingley, D. Navigating the local modes of big data: The case of topic models. In *Data Science for Politics, Policy and Government* (Cambridge University Press, New York, 2014).
25. Sontag, D., Roy, D. Complexity of inference in latent dirichlet allocation. In *NIPS* (2011), 1008–1016.
26. Vavasis, S. On the complexity of nonnegative matrix factorization. *SIAM J. Optim.* (2009), 1364–1377.
27. Zhou, T., Biltmes, J.A., Guestrin, C. Divide-and-conquer learning by anchoring a conical hull. In *NIPS* (2014), 1242–1250.

Sanjeev Arora (arora@cs.princeton.edu), Princeton University, Princeton, NJ, USA.
Rong Ge (rongge@cs.duke.edu), Duke University, Durham, NC, USA.
Yoni Halpern (yhalpern@gmail.com), Google, Cambridge, MA, USA.
David Mimno (mimno@cornell.edu), Cornell University, Ithaca, NY, USA.

Ankur Moitra and David Sontag (moitra, dsontag@mit.edu), MIT, Cambridge, MA, USA.
Yichen Wu (ychwu5@gmail.com), Stanford University, Stanford, CA, USA.
Michael Zhu (mhzhu@cs.stanford.edu), Stanford University, Stanford, CA, USA.

Copyright held by owners/authors.



The FIRST authoritative resource.

EDITED BY

Sharon Oviatt, *Incaa Designs*

Björn Schuller, *University of Passau, Imperial College London*

Philip Cohen, *VoiceBox Technologies*

Daniel Sonntag, *German Research Center for Artificial Intelligence*

Gerasimos Potamianos, *University of Thessaly*

Antonio Krüger, *German Research Center for Artificial Intelligence*



ISBN: 978-1-970001-64-8 DOI: 10.1145/3015783

<http://books.acm.org>

<http://www.morganclaypoolpublishers.com/acm>

CAREERS

University of Central Florida Tenure-Track Assistant Professor

The University of Central Florida (UCF) Center for Research in Computer Vision (CRCV) solicits applications for a full-time 9-month, tenure-track assistant professor in the area of Deep Learning.

CRCV is the world class leader in computer vision research and related disciplines including, but not limited to, the algorithmic aspects of deep learning and their applications in computer vision.

The selected candidate will be expected to begin to work on August 8, 2018. Salary and a start-up package will be commensurate with qualifications.

Apply to: <https://www.jobswithucf.com/postings/51488>.

University of Central Missouri Instructor of Computer Science

The School of Computer Science and Mathematics at the University of Central Missouri is accepting applications for two non tenure-track positions in Computer Science at the rank of Instructor. The appointment will begin August 2018. We are looking for faculty excited by the

prospect of shaping our school's future and contributing to its sustained excellence.

The Position: Duties will include teaching undergraduate courses in computer science, cybersecurity and/or software engineering, and developing new courses depending upon the expertise of the applicant and school needs, program accreditation and assessment. Faculty are expected to assist with school and university committee work and service activities, and advising majors. The typical teaching load is 12 credit hours per semester.

Required Qualifications:

- ▶ M.S. in Computer Science, Cybersecurity or Software Engineering
- ▶ Demonstrated ability to teach existing courses at the undergraduate level
- ▶ Excellent verbal and written communication skills

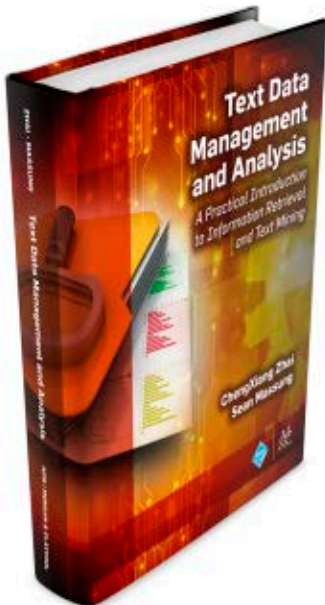
The Application Process: To apply online, go to <https://jobs.ucmo.edu>. Apply to position #997344 or #998560. The following items should be attached: a letter of interest, a curriculum vitae, copies of transcripts, and a list of at least three professional references including their names, addresses, telephone numbers and email

addresses. Official transcripts and three letters of recommendation will be requested for candidates invited for on-campus interview. For more information, contact

Dr. Songlin Tian, Search Committee Chair
School of Computer Science and Mathematics
University of Central Missouri
Warrensburg, MO 64093
(660) 543-4930
tian@ucmo.edu

Initial screening of applications begins March 15, 2018, and continues until position is filled. AA/EEO/ADA. Women and minorities are encouraged to apply.

UCM is located in Warrensburg, MO, which is 35 miles southeast of the Kansas City metropolitan area. It is a public comprehensive university with about 13,000 students. The School of Computer Science and Mathematics offers undergraduate and graduate programs in both Computer Science and Cybersecurity. The undergraduate Computer Science program is accredited by the Computing Accreditation Commission of ABET. The School also offers the first and only undergraduate Software Engineering program in the state of Missouri.



The most useful and practical knowledge for building a variety of text data applications.

COMPUTER SCIENCE STUDENTS
(Undergrad & Graduate)
LIBRARY & INFORMATION SCIENTISTS
TEXT DATA PRACTITIONERS

ChengXiang Zhai & Sean Massung (Authors)
University of Illinois at Urbana-Champaign

Text Data Management and Analysis covers the major concepts, techniques, and ideas in **information retrieval** and **text data mining**. It focuses on the practical viewpoint and **includes many hands-on exercises designed with a companion software toolkit** (i.e., MeTA) to help readers learn how to apply techniques of information retrieval and text mining to real-world text data.



ISBN: 978-1-970001-16-7 DOI: 10.1145/2915031

<http://books.acm.org>

<http://www.morganclaypoolpublishers.com/text>

[CONTINUED FROM P. 96] can move one kilometer in one minute. Can the admiral do this with 65 probes or fewer? If so, show how. For convenience, assume the submarine is always at an integer location.

Solution. Start by deploying probes at 20, 30, 40, 50, 60, and 70 measured from kilometer 0 of the line segment. This would be sufficient to know the submarine's location within an interval of size 10. So, for example, the Blue admiral could know the submarine is within $[0..9]$ if the probe at 20 kilometers detects the submarine but no other. The submarine is within $[10..19]$ if the probes at 20 and 30 kilometers detect the submarine but no others. The submarine is in $[20..29]$ if the probes at 20, 30, and 40 detect the submarine but no others. The submarine is in $[30..39]$ if the probes at 20, 30, 40, and 50 detect the submarine but no others. The submarine is in $[40..49]$ if the probes at 30, 40, 50, and 60 detect the submarine (and possibly the probe at 20, too) but no higher ones. The submarine is in $[50..60]$ if the probes at 40, 50, 60, and 70 (and possibly the probe at 30 too) detect the submarine. The submarine is in $[61..70]$ if the probes at 50, 60, and 70 detect the submarine. The submarine is in $[71..80]$ if the probes at 60 and 70 detect the submarine. The submarine is in $[81..90]$ if the probe at 70 detects the submarine. If no probes detect the submarine, then October is in $[91..99]$.

At the end of the probing the Blue admiral knows the submarine is at some location $[L..L+10]$. The admiral now waits 15 minutes, at which point the submarine is in the interval $[L-15..L+24]$, then puts probes at locations $L-26$, $L+25$, and $L+35$. The submarine is at $[L-15..L-6]$ if only probe $L-26$ detects it. The submarine is at $[L-5..L+4]$ if no probes detect it. The submarine is at $[L+5..L+14]$ if $L+25$ detects it, but $L+35$ does not. The submarine is at $[L+15..L+24]$ if $L+25$ and $L+35$ both detect it. The admiral can do this probing every 15 minutes so would need nine probes initially, then three probes every 15 minutes. The admiral would thus need $6 + 19 \times 3 = 63$ probes.

Upstart 1. On a line segment of length M , a detection radius d , and

The Blue admiral could know that the submarine is within $[0..9]$ if the probe at 20 kilometers detects the submarine but no other.

time T , find an algorithm that uses the minimum number of probes so at every moment of time up to time T , the Blue admiral achieves a precision of x ; that is, at every moment in time, the admiral knows some position p such that the submarine is in the interval $[p-x..p+x]$.

Upstart 2. Generalizing upstart 1 to two dimensions, now consider an area of size M by M , detection radius d , and time T , and find an algorithm that uses the minimum number of probes the Blue admiral would need so at every moment of time up to time T , the admiral achieves a precision of x ; that is, at every moment in time, the admiral knows some position p such that the submarine is in the circle of radius x around p .

Upstart 3. Generalize the two earlier upstarts to k dimensions to achieve a precision on a hypersphere of dimension k and radius x .

Upstart 4. How do these first three upstarts change if the Blue admiral can specify for each probe its detection distance just before deploying it? For example, the admiral can drop a first probe with detection radius 20 kilometers and then another probe with detection radius eight kilometers.

All are invited to submit their solutions to upstartpuzzles@cacm.acm.org; solutions to upstarts and discussion will be posted at <http://cs.nyu.edu/cs/faculty/shasha/papers/cacmpuzzles.html>

Dennis Shasha (dennisshasha@yahoo.com) is a professor of computer science in the Computer Science Department of the Courant Institute at New York University, New York, USA, as well as the chronicler of his good friend the omniheurist Dr. Ecco.

Copyright held by the author.



Association for
Computing Machinery

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

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

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

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

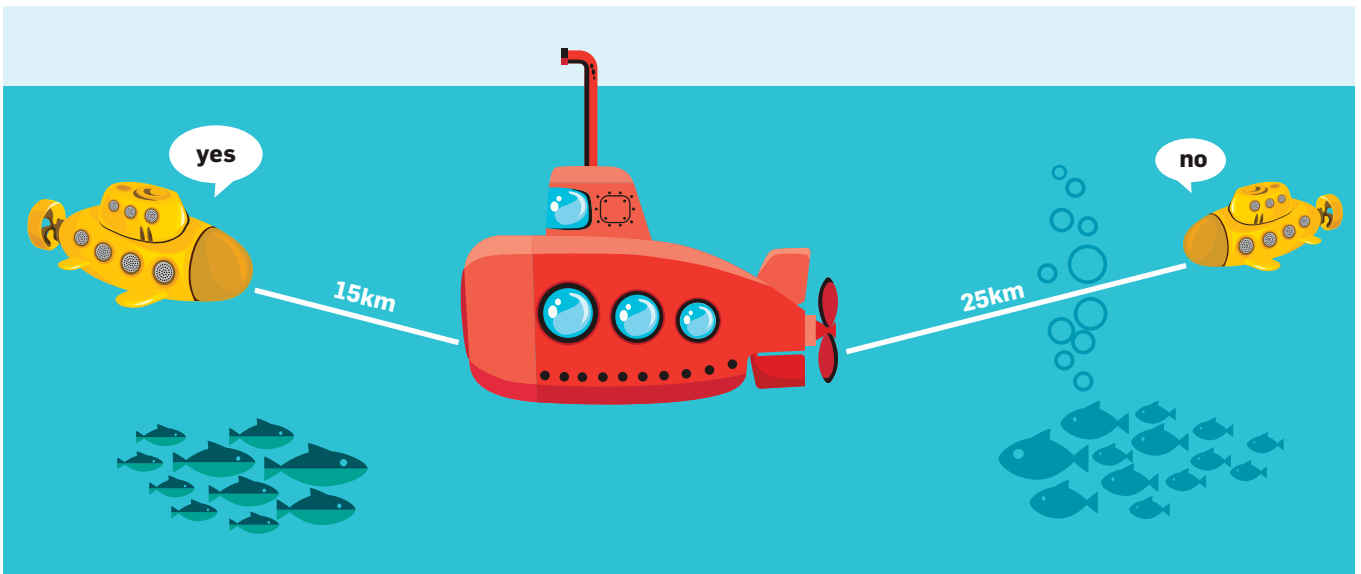


DOI:10.1145/3186264

Dennis Shasha

Upstart Puzzles

Finding October



A set of range probes dropped into the water over time track the position of the red submarine to a tight interval as it moves.

TWO TEAMS, BLUE AND RED, play a game in which Red has a submarine we call October, as in Tom Clancy's novel *The Hunt for Red October*. The Blue admiral is trying to locate the submarine using helicopter-deployed probes that are dropped in the water (see the Figure here). A probe will detect the submarine if the probe's position is within some distance d of the submarine, returning a message either "detected within d " or "not detected within d ."

The Blue admiral's goal is to know the position of the submarine within a distance x over a time period T , using as few probes as possible.

Position knowledge goes stale over time because the submarine can move, but the degree of staleness is bounded by the speed of the submarine. For example, suppose the submarine must move on a line and at most one kilometer in one minute. If

the Blue admiral knows October's position at, say, minute t to be no more than distance e with respect to some position p , and $e < x$, then October's position is within distance x of point p for at least $x - e$ more minutes.

Position knowledge goes stale over time because the submarine can move, but the degree of staleness is bounded by the speed of the submarine.

Warm-Up. If a probe is able to detect October's position within a distance of 20 kilometers from the probe or not, can the Blue admiral use some number of probes to determine whether October's position is now between, say, kilometer 40 and kilometer 45 along a particular line segment?

Solution. Yes. For example, suppose Blue drops one probe A at 60 and another B at $65 + \epsilon$ for some tiny ϵ . If probe A detects the submarine, but B does not, then the submarine is in the interval $[40..45]$.

Challenge. Suppose the submarine can move only on a line segment of length 100 kilometers, and the probe range d is 20 kilometers. At every moment in time up to time 600 minutes from the starting time 0, the Blue admiral would like to know the location of the submarine within an interval of 10 kilometers. Assume the submarine [CONTINUED ON P. 95]

ICMI 2018

The 20th International Conference on Multimodal Interaction

Boulder, Colorado
October 16-20, 2018

icmi.acm.org

ICMI is the conference for research on multimodal human-human and human-computer interaction, interfaces, and system development. The conference focuses on theoretical and empirical foundations, component technologies, and combined multimodal processing techniques that define the field of multimodal interaction.

Long and Short Paper
Submission May 1, 2018

The Emotion Recognition in
the Wild Challenge Submission July 1, 2018

The ICMI Eating Analysis
Challenge Submission May 30, 2018

Topics include:

- Affective computing
- Cognitive modeling
- Gesture, touch and haptics
- Healthcare, assistive technologies
- Human communication dynamics
- Human-robot/agent interaction
- Interaction with smart environments
- Multimodal machine learning
- Multimodal mobile systems
- Multimodal behavior generation
- Multimodal datasets and validation
- Multimodal dialogue modeling
- Multimodal fusion and representation
- Multimodal interactive applications
- Multimodal social interactions
- Multimodal system components
- Visual behaviors in social interaction
- Virtual/augmented reality

2018 Artificial Intelligence · Blockchain · Cloud

— · CREATING THE FUTURE · —

June 25 - June 30, Seattle, USA

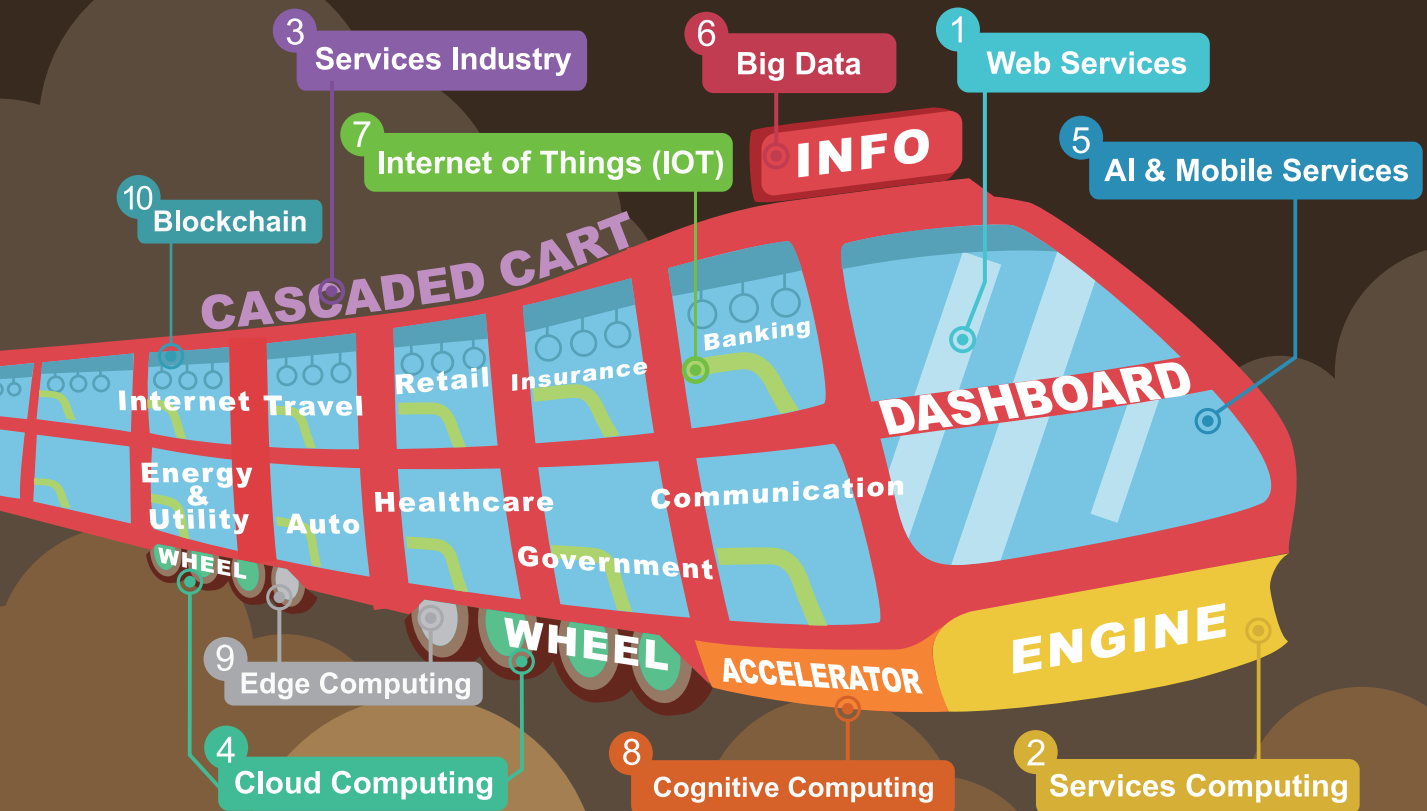
CELEBRATING THE 16th BIRTHDAY OF ICWS



SCF

SERVICES
CONFERENCE
FEDERATION

- 1 2018 International Conference on Web Services (**ICWS 2018**)
- 2 2018 International Conference on Services Computing (**SCC 2018**)
- 3 2018 World Congress on Services (**SERVICES 2018**)
- 4 2018 International Conference on Cloud Computing (**CLOUD 2018**)
- 5 2018 International Conference on AI & Mobile Services (**AIMS 2018**)
- 6 2018 International Congress on Big Data (**BigData Congress 2018**)
- 7 2018 International Conference on Internet of Things (**ICIOT 2018**)
- 8 2018 International Conference on Cognitive Computing (**ICCC 2018**)
- 9 2018 International Conference on Edge Computing (**EDGE 2018**)
- 10 2018 International Conference on Blockchain (**ICBC 2018**)



Submission Deadlines

3/16/2018: ICWS 2018 (<http://icws.org>)
 3/21/2018: SCC 2018 (<http://theSCC.org>)
 3/31/2018: SERVICES 2018 (<http://ServicesCongress.org>)
 3/16/2018: CLOUD 2018 (<http://theCloudComputing.org>)
 3/21/2018: AIMS 2018 (<http://ai1000.org>)

3/21/2018: BigData Congress 2018 (<http://BigDataCongress.org>)
 3/31/2018: ICIOT 2018 (<http://iciot.org>)
 3/31/2018: ICBC 2018 (<http://theCognitiveComputing.org>)
 3/31/2018: EDGE 2018 (<http://theEdgeComputing.org>)
 3/31/2018: ICBC 2018 (<http://Blockchain1000.org>)

Email:
cnfs@ServicesSociety.org



Largest not-for-profits organization (501(c)(3))
 dedicated for serving 30,000+ worldwide
 services computing professionals



ICWS.ORG