COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

05/2018 VOL.61 NO.05

Speech Emotion Recognition

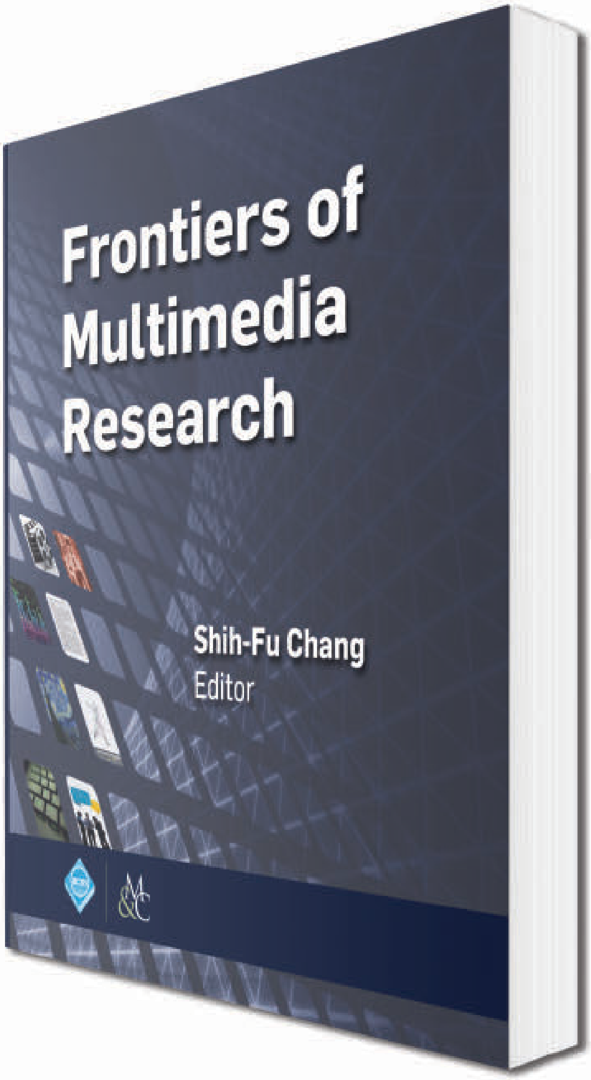
Never-Ending Learning

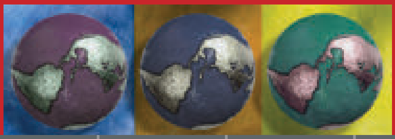
Toward Sustainable Access: Where Are We Now?

Canary Analysis Service

ACM's General Election Ballot

Association for Computing Machinery


Frontiers of Multimedia Research
Shih-Fu Chang
Editor



COMMUNICATIONS OF THE ACM

Departments

5 Cerf's Up
Turing Test 2
By Vinton G. Cerf

6 Editorial
Toward Sustainable Access: Where Are We Now?
By Jack Davidson, Joseph Konstan, Andrew A. Chien, and Scott Delman

9 Vardi's Insights
How We Lost the Women in Computing
By Moshe Y. Vardi

10 Letters to the Editor
Get ACM (and *Communications*) Out of Politics

13 ACM's 2018 General Election

24 BLOG@CACM
Commenting on Code, Considering Data's Bottleneck
Edwin Torres considers the enduring value of code comments, while Walid Saba wonders if we have overreacted to the knowledge acquisition bottleneck.

7 Calendar

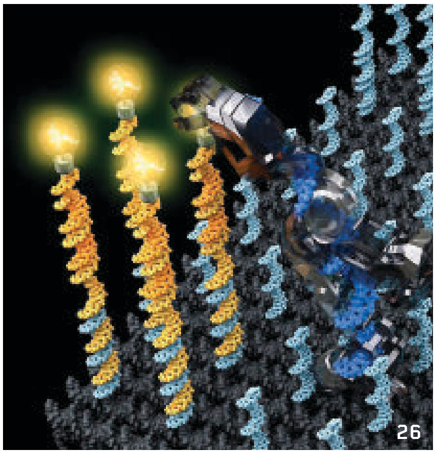
117 Careers

Last Byte

120 Future Tense
Free Press
When all online news and comment can be digitally manipulated, some might recall a more trustworthy way to spread the word.
By Ken MacLeod

Association for Computing Machinery
Advancing Computing as a Science & Profession

News

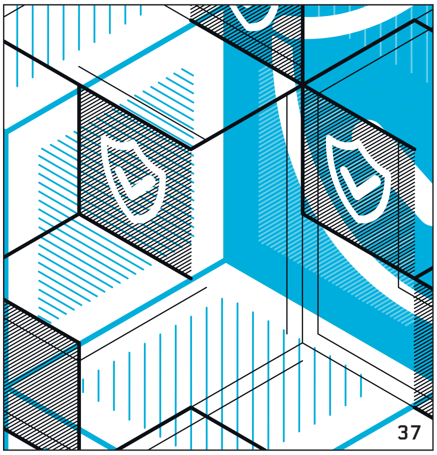


26 **Shrinking Machines, Cellular Computers**
Scientists are using DNA and RNA to build the world's tiniest robots and computing devices.
By Gregory Mone

29 **Using Functions for Easier Programming**
Functional programming languages automate many of the details underlying specific operations.
By Neil Savage

31 **Finding a Healthier Approach to Managing Medical Data**
Researchers are exploring ways to put medical data to greater use while better protecting privacy.
By Samuel Greengard

Viewpoints



34 Law and Technology
Is the Law Ready for Driverless Cars?
Yes, with one big exception.
By Ryan Calo

37 Privacy and Security
Putting Trust in Security Engineering
Proposing a stronger foundation for an engineering discipline to support the design of secure systems.
By Fred B. Schneider

40 Education
Scale or Fail
Moving beyond self-selected computer science education in Switzerland.
By Alexander Repenning

43 Viewpoint
The March into the Black Hole of Complexity
Addressing the root causes of rapidly increasing software complexity.
By Harold "Bud" Lawson

46 Viewpoint
Science, Policy, and Service
Some thoughts on the way forward.
By Margaret Martonosi

(L) IMAGE BY DEMIN LIU, COURTESY OF CALIFORNIA INSTITUTE OF TECHNOLOGY; (R) IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

Practice

50 **Research for Practice: Cluster Scheduling for Datacenters**
Expert-curated guides to the best of CS research.
By Malte Schwarzkopf

54 **Canary Analysis Service**
Automated canarying quickens development, improves production safety, and helps prevent outages.
By Štěpán Davidovič and Betsy Beyer

63 **How Is Your Week Going So Far?**
Praise matters just as much as money.
By Kate Matsudaira

Articles' development led by acmqueue
queue.acm.org

About the Cover:
Dialogue systems like Siri and Alexa may be all the rage, but they don't capture the emotions behind our words. Researchers in the field of speech emotion recognition (SER) have been toiling for over 20 years to make machines *hear* our emotions and in this issue Björn Schuller (p. 90) traces the advances to date and the work ahead. Cover illustration by Vault49.

Contributed Articles

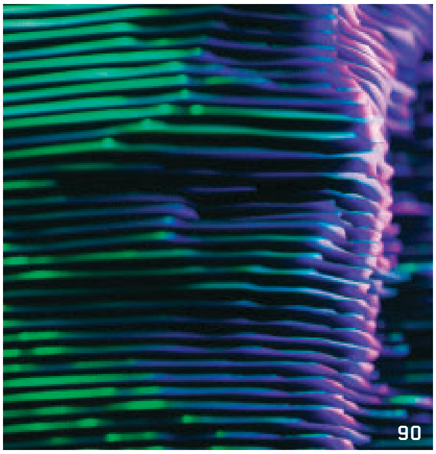
66 **More Than the Code: Learning Rules of Rejection in Writing Programs**
A teacher and students coding together make explicit the unwritten rules of programming.
By Josh Tenenberg, Wolff-Michael Roth, Donald Chinn, Alfredo Jornet, David Socha, and Skip Walter

72 **Internet Freedom in West Africa: Technical Support for Journalists and Democracy Advocates**
The U.S. State Department's Internet Freedom agenda is being adapted to help them communicate without DNS and IP address filtering.
By Richard R. Brooks, Lu Yu, Yu Fu, Oluwakemi Hambolu, John Gaynard, Julie Owono, Archippe Yepmou, and Felix Blanc

Watch the authors discuss their work in this exclusive *Communications* video. https://cacm.acm.org/videos/internet-freedom-in-west-africa

83 **Data Acquisition in Vehicular Ad Hoc Networks**
The data comes from multiple optimal sources in parallel, helping reduce addressing and data-acquisition latency.
By Xiaonan Wang

Review Articles



90 **Speech Emotion Recognition: Two Decades in a Nutshell, Benchmarks, and Ongoing Trends**
Tracing 20 years of progress in making machines hear our emotions based on speech signal properties.
By Björn W. Schuller

Watch the author discuss his work in this exclusive *Communications* video. https://cacm.acm.org/videos/speech-emotion-recognition

Research Highlights

102 Technical Perspective
Breaking the Mold of Machine Learning
By Oren Etzioni

103 **Never-Ending Learning**
By T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling

(L) IMAGE BY NUCLEARTIST; (C) PHOTO BY FEDERICO SCOPPA/AFP/GETTY IMAGES; (R) ILLUSTRATION BY VAULT49

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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

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

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; Xiang-Yang 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

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
Ilia 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
Ilia Rodriguez
ilia.rodriguez@hq.acm.org

Media Kit acmmediasales@acm.org

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

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

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/3200485

Vinton G. Cerf

Turing Test 2

IN 1950, ALAN TURING wrote a paper entitled "Computing Machinery and Intelligence."[a] He proposed a test in which a human attempts to distinguish between a human and a computer by exchanging text messages with each of them. If the human is unable to distinguish between the two, the computer is said to have passed the "Turing Test." In fact, there were variations, including one in which a human interrogator interacting with a man and a woman was to try to tell which was the man and which was the woman. Turing called this the "Imitation Game." The first version is sometimes now called the "Standard Turing Test."

In this modern era, in which the Internet and the World Wide Web play such visible roles, a different problem arises. In this version, which I will call "Turing Test 2," a computer program undertakes textual interactions with a human and another computer. The task of the computer program is to distinguish between the human and the computer. If the computer program successfully identifies which correspondent is a human and which is a computer, it has successfully passed Turing Test 2. If it cannot, then it fails the test. One particular form of this test is called a CAPTCHA[b] (Completely Automated Public Turing test to tell Computers and Humans Apart). These tests take many forms, but a popular variation is to display a distorted image of a word or random string of numbers and characters. In theory, a human interacting with the CAPTCHA will successfully respond with the correct alphanumeric string while a computer program, interacting with the same image will not succeed. There are other variations, for example, in which an image of an equation is displayed and the solution to the equation must be entered in response. Assuming the image is just a set of pixels, the challenge for the computer program trying to appear human is to correctly identify the equation and solve it.

Much has been written about the increasingly sophisticated ability of computer programs to pass the CAPTCHA tests or a variation in which the program sends the image to a human on the Internet who is given some benefit or payment for solving the problem, which is then relayed by the imitating program to the computer program running the CAPTCHA test. This is not merely an amusing game. As computer programs have grown capable of more sophisticated behavior, they are being used to emulate humans to fool less-sophisticated programs into treating computer-generated actions as if they originate from a human. This is an important practical problem because failure to make this distinction may mean malicious programs can register millions of fake identities on an email system for purposes of sending phishing[c] email messages or making comments on social media Web pages. One reason this is now a serious matter is that such programs (called "bots") are being used to distort news and social media to trick humans into accepting false information ("fake news") as true or simply reinforcing incorrect or biased beliefs through confirmation bias and "echo chamber" effects. Of course, bots can also be used to launch denial-of-service attacks or to pollute crowdsourcing systems. The technical challenge is that a computer program may be hard-pressed to distinguish between input from a human or from a computer because the same paths and media are used to carry the input.

For Turing Test 2, a computer program undertakes textual interactions with a human and another computer. The task of the program is to distinguish the human from the computer.

On the other hand, increasingly difficult CAPTCHA practices can drive humans crazy. "Which pictures do NOT contain traffic signs?" "Confirm this statement, 'there are no images or partial images of automobiles in this set of pictures.'"

Humans may justifiably want to throw their computers through the nearest window when poorly executed CAPTCHAs prevent them from legitimately accessing online services. C

[a] Turing, A. Computing machinery and intelligence. *Mind 49,* 236 (Oct. 1950), 433–460; doi: 10.1093/mind/LIX.236.433

[b] https://en.wikipedia.org/wiki/CAPTCHA

[c] Messages intended to fool a human user into clicking on a hyperlink leading to the ingestion of malware into the user's computer or smartphone or taking an action such as sending money to the account of a person committing a fraud.

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

DOI:10.1145/3201219

Jack Davidson, Joseph Konstan, Andrew A. Chien, and Scott Delman

Toward Sustainable Access: Where Are We Now?

ACM'S PUBLICATIONS PROGRAM is a core part of fulfilling its mission to advance computing as a science and a profession. ACM's conference proceedings, journals, books, magazines, and newsletters comprise an essential component of ACM's identity as well as service to members and the profession. We are proud of the preeminence of the ACM Digital Library and its suite of services that provide access to these publications, ensure their preservation, and is also a repository for a growing breadth of related artifacts including video, code, and datasets.

Selling access to ACM's publications, primarily through institutional subscriptions to the DL, pays for the direct costs of running ACM's publishing program, provides funds for SIG-specific initiatives, and supports the many good works of ACM, including its curriculum and education efforts, public policy initiatives, and broad support of diversity efforts worldwide.

For decades, ACM has carefully balanced sustainability of the publishing program with providing authors the opportunity to disseminate their work widely. Examples include longstanding author rights to post their papers through personal or institutional websites, reuse in future publications, and of course to share copies with anyone who might wish to read them.

The publishing landscape is changing, and ACM with it. We recognize the importance of the Open Access (OA) movement—the deeply held belief that research should be available to all—to advance the field and to ensure access for scholars who are unaffiliated or whose institutions are not subscribers. We also recognize and respect other trends, including sponsor mandates for open publishing, the desire for more replicable science, and the accelerating dissemination through distribution of preprints (including prior to peer review).

The publishing landscape is changing, and ACM with it.

For this reason, we take this opportunity to describe what ACM, and in particular its Publications Board, thinks about these issues, recent changes, and the future.

Six Key Principles

As we consider directions in publications, we focus on six key principles:

Sustainability. Both the financial sustainability of ACM and its publishing program underpin the ability to ensure content is available indefinitely. ACM must ensure no ACM publication is ever "out of print." A portion of the publishing budget covers costs of making digital content accessible through changing standards and guarantees there will be a backup provider should ACM be unable to publish the DL.

Access. Broad access by both readers and authors. We strive to keep subscription prices low and provide a variety of mechanisms for authors to make their work visible, easily discoverable, and freely accessible. All ACM publications support Gold OA with an author-paid article processing charge (APC). In addition, authors have a variety of Green OA options including posting an "Author-izer" Web link that gives readers direct access at no charge. And we are careful to ensure author-pays Gold OA publications have provisions for authors who cannot pay. We also help authors comply with funder mandates for open access.

Quality. Highest quality of technical content and publication. The ACM, thanks to its members and volunteers, as well as its history of quality, is a trusted brand. Publications are reviewed regularly for quality; proposed publications

Jack Davidson

Joseph Konstan

undergo rigorous review by the ACM community; and ACM uses state-of-the-art plagiarism-detection software and invests substantial volunteer time in handling cases of plagiarism, research misconduct, and other ethics violations. Our peer-review is first-rate, and we document in the DL key quality metrics associated with each publication.

Author Choice. Provide publication access options that serve authors' scientific objectives. Authors can make decisions about how their work is accessible, selecting among the variety of access models and corresponding fee structures. Similarly, ACM authors can choose how they wish to manage rights, from self-management to having ACM handle everything.

Service. Provide an excellent experience for authors and readers. To better support authors and readers, we are investing in new publishing technology that reduces author effort for manuscript preparation and submission, streamlines the publication process, and provides greatly improved search and discovery of relevant articles. The new platform will also allow ACM to render articles accessible for readers with disabilities and readable on a broad range of devices.

Community Choice. Enable communities to make choices that reflect their different values and priorities. We support conferences and journals that have gone entirely Gold OA (through author-paid APCs or by paying APC costs in conference budgets). All ACM conferences have the option of a month-long "open surround" access period to their proceedings in the DL, and the option of open tables of contents on their Special Interest Group pages that provide open access to specific proceedings.

ACM continues to explore models to enable greater sustainable open access. For example, we are working with institutional subscribers to explore "author-side" subscriptions where institutions pre-commit to paying the Gold OA APCs for some or all of their authors' publications. We are also working, in partnership with other publishers, to better support authors in meeting the open access requirements of government-funded research. In addition, a partnership with arXiv is exploring ideas for how to best connect preprints and final published papers. And of course, a key effort is the next generation of the ACM DL, incorporating features that broaden the notion of publication, supporting dissemination of code, data, and other research artifacts.

We would love to hear from you!

Jack Davidson,
CO-CHAIR
ACM PUBLICATIONS BOARD

Joseph Konstan,
CO-CHAIR
ACM PUBLICATIONS BOARD

Andrew A. Chien,
EDITOR-IN-CHIEF
COMMUNICATIONS OF THE ACM

Scott Delman,
DIRECTOR
ACM PUBLICATIONS

Andrew A. Chien

Scott Delman

Calendar of Events

May 4–6
**I3D '18: Symposium on Interactive 3D Graphics and Games,
Montreal, Canada
Sponsored: ACM/SIG,
Contact: Morgan McGuire,
Email: morgan@cs.williams.edu**

May 8–10
**CF '18: Computing Frontiers Conference,
Ischia, Italy,
Sponsored: ACM/SIG,
Contact: David R. Kaeli,
Email: kaeli@ece.neu.edu**

May 15–18
**I3D '18: Symposium on Interactive 3D Graphics and Games,
Montreal, Canada,
Sponsored: ACM/SIG,
Contact: Morgan McGuire,
Email: morgan@cs.williams.edu**

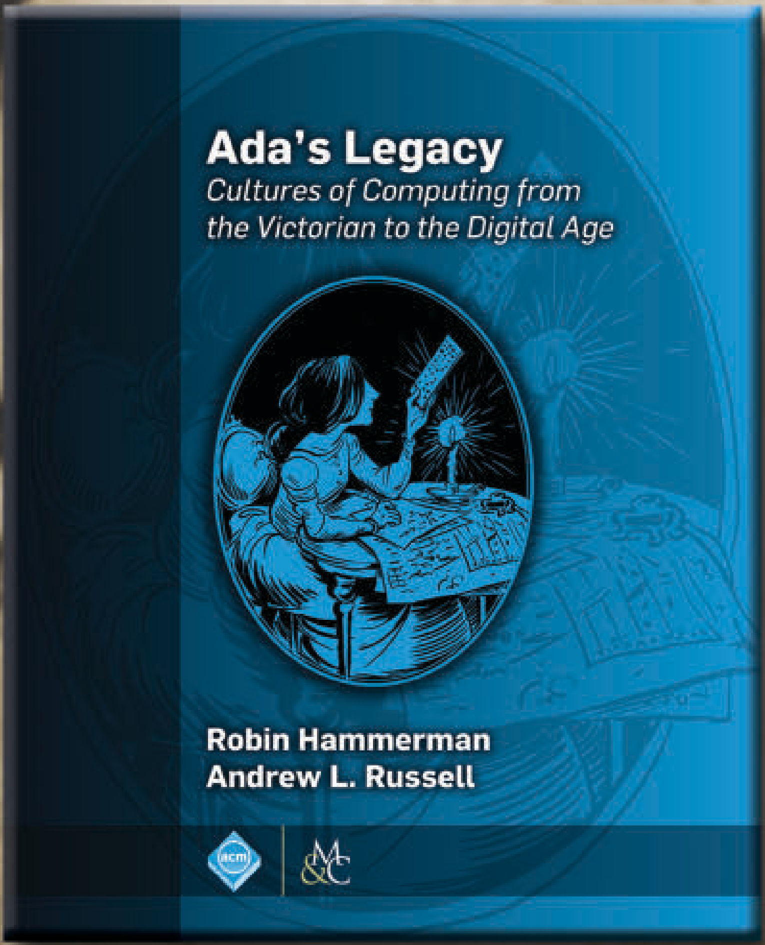
May 23–25
**GLSVLSI '18: Great Lakes Symposium on VLSI 2018,
Chicago, IL,
Sponsored: ACM/SIG,
Contact: Deming Chen,
Email: dchen@illinois.edu**

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**

May 27–28
**ICPC '18: 26th IEEE/ACM International Conference on Program Comprehension,
Gothenburg, Sweden,
Contact: Foutse Khomh,
Email: foutse.khomh@polymtl.ca**

May 27–28
**MOBILESoft '18: 5th IEEE/ACM International Conference on Mobile Software Engineering and Systems,
Gothenburg, Sweden,
Contact: Christine L. Julien ,
Email: c.julien@mail.utexas.edu**

May 27–June 3
**ICSE '18: 40th International Conference on Software Engineering,
Gothenburg, Sweden,
Contact: Ivica Crnkovic,
Email: crnkovic@chalmers.se**


Ada's Legacy
Cultures of Computing from the Victorian to the Digital Age
Robin Hammerman
Andrew L. Russell
acm
M&C

DOI:10.1145/3201113

Moshe Y. Vardi

How We Lost the Women in Computing

In July 2017, Google engineer James Damore distributed a memorandum titled "Google's Ideological Echo Chamber," which was critical of Google's diversity policies. The memo "went viral" and was widely distributed inside and outside of Google, leading to extensive media discussions. In August 2017, Google fired Damore for violation of the company's code of conduct. The U.S. National Labor Relations Board concluded that Google did not violate U.S. federal labor law when it fired Damore, but Damore filed a lawsuit against Google for discrimination.

The memo's central argument was that the gender disparity observed in the tech industry in general, and in Google in particular, could be partially explained by biological differences between women and men. In essence, argued Damore, women are less interested in computing then men. Ironically, over the past few years the historical role of women in computing has become much clearer. We cannot, I believe, understand the current gender disparity in computing without understanding the history of women in computing.

The critical roles played by Ada Lovelace and Grace Hopper are widely known. Lovelace worked closely with Charles Babbage, the British mathematician who was the first to conceive of general-purpose computers, and was first to realize that computers will have applications beyond pure calculation. Hopper was one of the first programmers of the Harvard Mark I computer and played a key role in the development of COBOL. But quite often Lovelace and Hopper are the only women to receive recognition for their significant role in computing history. For example, it is less well known that seven women[a] were the world's first programmers, having programmed the ENIAC, the first general-purpose, electronic, programmable computer.

But the general ignorance of computing history goes deeper. The early programmers were women because until the development of electronic computers, computing used to be a human job; computers were humans who computed. Computing required precision and patience, and most pre-ENIAC human computers were female. Specifically, women played a key role in code breaking, which has had an intimate connection with computing. Three recent books describe this key role played by women in cryptology. *Women Codebreakers at Bletchley Park*, by Kerry Howard,[b] deciphers the legacy of British women codebreakers in World War II. *Code Girls*, by Liza Mundy, tells the story of over 11,000 women, who comprised more than 70% of all U.S. code breakers during that war. *The Woman Who Smashed Codes: A True Story of Love, Spies, and the Unlikely Heroine Who Outwitted America's Enemies*, by Jason Fagone, chronicles the life of Elizabeth Smith Friedman, who played a leading role in U.S. cryptanalysis for 40 years.

Another recent book, *Brotopia*, by Emily Chang, describes how "Silicon Valley disrupts everything but the Boys' Club." "From its earliest days," Chang writes, "the industry has self-selected for men: first, antisocial nerds, then, decades later, self-confident and risk-taking bros." As a prelude, I suggest reading the *Vanity Fair* disputed excerpt,[c] featuring Chang's reporting about "exclusive, drug-fueled, sex-laced parties" where women are preyed upon. But the controversial sex parties are a small part of Silicon Valley's problems. The main story of the book is of a culture is that highly hostile to women.

A.T. Wynn and S.J. Correll, two Stanford sociologists, reach the same conclusion in their recent paper in *Social Studies of Science*, titled "Puncturing the pipeline: Do technology companies alienate women in recruiting sessions?"[d] Using original observational data from recruiting sessions hosted by technology companies, they found that company representatives often engage in behaviors known to create a chilly environment for women. They concluded that representatives "may puncture the recruiting pipeline, lessening the interest of women at the point of recruitment into technology careers."

One may think these problems are specific to Silicon Valley, but the recent #MeToo movement made it clear that academic environments can also be hostile to women. I urge you to read 'What Happens to Us Does Not Happen to Most of You,'[e] where Kathryn McKinley provides "a personal account of sexism, harassment, and racism that I and some anonymous members of the computer-architecture community have experienced."

So how did we lose the women in computing? They did not just leave; they were pushed out. There is hard work ahead of us to start to undo the damage. Check out ACM SIGARCH's Committee to Aid REporting on discrimination and haraSsment (CARES)[f] and its diversity conversations[g] to see what the computer-architecture community is doing.

[a] https://goo.gl/8AvX9G
[b] https://goo.gl/WX2shx
[c] https://goo.gl/gmLMVU
[d] https://goo.gl/sdxqwD
[e] https://goo.gl/zfDAco
[f] https://goo.gl/UgU2m9
[g] https://goo.gl/yuy3oU

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*.

DOI:10.1145/3203077

Get ACM (and *Communications*) Out of Politics

RECENT EDITORIAL POLICY seems to have let ACM morph into what I would call the *left-leaning* ACM. Examples include Moshe Y. Vardi's editorial "ACM's Open-Conference Principle and Political Reality" (Mar. 2017) where he addressed bathroom laws in several U.S. states with respect to men who might want to use the "ladies room" and vice versa. Vardi said, "In January 2017, the ACM SIGMOD Executive Committee decided to move the SIGMOD/PODS 2017 conference out of North Carolina" due to its HB2 Public Facilities Privacy & Security Act, as passed in March 2016, prohibiting flexibility for the transgendered, even though Vardi, for the record, disagreed with the move. None of this is relevant to computers or programming.

Another example is Thomas Haigh's "Historical Reflections" column "Defining American Greatness: IBM from Watson to Trump" (Jan. 2018). In its "Watson and Trump" section, Haigh said, "Trump promised to make America great again by building walls, stepping back from its commitment to the defense of NATO allies, and tearing up trade deals." I see zero relevance of such a statement to computers nor was it completely accurate. What then-candidate, now-President Donald J. Trump has said about NATO is the U.S. will require its "allies" to "pay their fair share of the cost" of the common defense, defined as a percentage of GDP, rather than continue to mooch off the American taxpayer. The U.S. president is *obliged* to keep illegal aliens out of the U.S. and "tear up trade deals" that are bad for America, as spelled out in the Constitution. I see no relevance to computer technology in mentioning Trump. I should think a headline saying, for example, "Defining American Greatness: IBM" without mentioning Trump would have been sufficient.

Haigh also neglected the darker side of IBM's history (such as selling tabulating machines to Nazi Germany to help identify citizens with even a fraction of Jewish lineage so they could be rounded up for genocide[1] or efforts to crush its competitors, resulting in a 1956 U.S. Department of Justice consent decree and in 1973 Telex Corp. being awarded in Federal Court $352.5 million from IBM for antitrust violations. As published, the column could have come from IBM's PR department.

Having had the honor of being published in *Communications* ("The NSA and Snowden: Securing the All-Seeing Eye," May 2014), I can attest to the rigor of the editorial review when I strayed even a little short of the highest standards. I am not complaining, as it made for a better, more credible article. I urge ACM's leadership and *Communications*' editors to reconsider their editorial policy and ask the two authors I mention here to explain their motivations or revise the organization's name to reflect its left-leaning inclinations.

Reference

1. Black, E. *IBM and the Holocaust: The Strategic Alliance Between Nazi Germany and America's Most Powerful Corporation.* Crown Publishers, New York, 2001.

Bob Toxen, Peachtree Corners, GA, USA

Author Responds:

I fail to see how my editorial can be called "left leaning." I also fail to see how policies on locations for ACM conferences are outside the scope for Communications. *And, as Toxen's own* Communications *article shows,* Communications *is definitely not only about computers and programming.*

Moshe Y. Vardi, Houston, TX, USA

Author Responds:

To understand computing's history we need to understand IBM, and to understand IBM we need to understand IBM's evolving political context. IBM's old slogan was "World Peace Through World Trade." As a charter member of Eisenhower's "military industrial complex" IBM helped America build unrivaled military, scientific, and economic might while safeguarding democracy in Europe. I argued that IBM's later shift to gaming short-term financial metrics, which shrank the company and shifted jobs from the U.S. to India, illustrated the appeal of Trump's diatribes against "globalists." By accusing me of leftist bias for glorifying free trade and old-school corporate capitalism (just think about that for a second), Toxen unwittingly captured the rise of open vs. closed political alignments over traditional left vs. right ones.

Thomas Haigh, Shorewood, WI, USA

For Old(er) Users, Talking Still Beats Texting

Bran Knowles's and Vicki L. Hanson's contributed article "The Wisdom of Old(er) Technology (Non)Users" (Mar. 2018) took a condescending attitude, saying old(er) users must learn to be more "fully participating, independent citizens in our increasingly digital society." As a 70-something software engineer who still teaches computer science and cybersecurity in a U.S. university, I was put off by such arrogance.

Consider the following touchstones of today's mass digital culture:

Perceptions of risk and responsibility. We old(er) users do not fear technology but rather the carelessness of the people administering it. After getting letters describing how our data had been exposed and stolen from the U.S. Office of Personnel Management, Target, Equifax, and others, why should we trust it to yet another organization's data sieve? Moreover, it is not that we fear making decisions we previously left to others, but finally realize it is pointless to even try to keep up with the everyday tweaks to the system;

Values. When I originally was in college (1965–1969), before a professor would arrive in the classroom, the classroom would naturally be abuzz with conversation over very human concerns, say, test results, dating, or an upcoming basketball game. When the professor entered the room, a hush would replace the conversation. Today, when I (now as the prof) enter a classroom, I see students hunched over cellphones, with the only sound thumbs striking glass;

Cultural expectations. Those were Knowles's and Hanson's biases, not mine, apparently seeing us old(er) people as obsolete. We in turn choose to see the younger generation as impulsive narcissists who could use some advice. But that would require today's generation to put down their phones and talk to us;

Listening. The fact that younger people choose to ignore us has not changed in 6,000 years. Moreover, I admit we did not do it either when it was our turn; and

Changing interface. Since 1987, I have used Mail, Lotus Notes, Roundcube, multiple versions of Outlook, and other systems I can no longer name. What I do need is a reliable way to send and receive information. Do I need HTML? Not really. And even when I use it, it gets stripped out anyway when I email something to colleagues in government agencies, something I do often. Same with fancy colors, fonts, backgrounds, and cute pictures. Please also do not insist on constantly changing the features just to sell a new version.

We old(er) humans are simply not all that enamored of the latest and greatest tech (recall that, in many cases, we created it), nor are we impressed by the ability to add emojis to our digital correspondence. We have learned that talking is more satisfying than texting, and visits from grandchildren are better than Facebook. Do not pity us—though, if you like, you may envy us.

Joseph M. Saur, Virginia Beach, VA, USA

Authors Respond:

Saur reflects many of the frustrations we reported in our article, but we must not forget that young people use many digital tools not by choice but out of social and economic necessity. Our concern is that society is becoming less accommodating to people lacking resources or desire to develop digital skills, and that chipping away at the freedom to reject technologies will silence important debates about their effects on our lives. Instead of forcing older adults to digitize, their objections need greater attention.

Bran Knowles, Lancaster, U.K., and
Vicki L. Hanson, Rochester, NY, USA

Don't Trust the Deadly Dilemma

Imagine the year is 2028, and self-driving cars have the run of U.S. roads, with more than 25 million at any given moment. Imagine further a well-designed cyberattack or self-motivated artificial intelligence bot causing simultaneous malfunctions in the braking systems in, say, 70% of them, while also directing others into crowds to maximize some evil intent. Tens of millions are injured or killed in possibly the greatest one-day tragedy ever.

Now imagine a peaceful alternative, with self-driving technology revolutionizing road transportation. Not only does the technology allow drivers to use their time more efficiently, it also significantly reduces the number of car crashes, potentially to zero. Major causes of crashes are practically eliminated, most notably due to driver error. Compare against the current reality of human-operated vehicles, whereby motor-vehicle collisions in the U.S. alone, for example, are associated with approximately 37,000 deaths per year.[1]

In the context of designing trustworthy self-driving cars, Benjamin Kuipers, in his review article "How Can We Trust a Robot?" (Mar. 2018) addressed the "deadly dilemma," or an AI designed to choose between two bad options, both very likely harmful to humans, illustrating a rare, but plausible, situation in which the computational intelligence controlling a self-driving car must choose between two alternatives—one that could result in the driver's injury or death and the other that will save the driver but is certain to cause harm to others.

The fundamental assumption of the deadly dilemma is that self-driving cars will indeed be in broad use someday. It does not assume other technologies that might help eliminate crashes will significantly advance by the time they fill the road. Ignoring it could disqualify the dilemma's inverse correlation with increased trust. Technologies expected to obviate both possibilities, as outlined in the deadly dilemma, include GPS navigation and car- and ground-located sensors designed to identify other cars, along with pedestrians and bicyclists; braking systems that can stop a car just in time; and new types of construction materials in cars and roads more likely to protect humans than their current counterparts.

History has recorded many technologies that were not viewed, even by their users, as trustworthy at first. But transportation-related technologies have ultimately won our trust, though all have resulted in some number of human deaths and injuries. Despite the occasional destructive results, most are still in use because they overall improve human quality of life while saving money and time. Anticipating the level of trust in a new, potentially harmful technology, particularly autonomous AI-directed machines, should thus account for all other related technologies that, combined, could result in some generally acceptable risk that will not prevent the broad adoption of the new technology. Determining the trustworthy option in the deadly dilemma must account for all associated technologies. Otherwise, it is not just potentially misleading but really no dilemma at all.

Reference

1. Association for Safe International Road Travel. *Annual United States Road Crash Statistics, 2018*; http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics

Uri Kartoun, Cambridge, MA, USA

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

Coming Next Month in **COMMUNICATIONS**

ACM A.M. Turing Award Recipients: John L. Hennessy and David A. Patterson

Identifying Patterns in Medical Records with Latent Analysis

Bias and the Web

Ability-Based Design

Privacy in Decentralized Cryptocurrencies

Coz: Finding Code that Counts with Causal Profiling

Also, the latest news about how deep learning works, 3D sensor chips and apps, and getting hooked on technology.

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).

2017
Volume 6, Number 1

ACM Transactions on
Human-Robot Interaction

Article 1	O. C. Jenkins, S. Sabanovic	Editorial Introduction: Impact, Sustainability, and Inclusion for JHRI
Article 2	F. Levillain, E. Zibetti	Behavioral Objects: The Rise of the Evocative Machines
Article 3	H. Admoni, B. Scassellati	Social Eye Gaze in Human-Robot Interaction: A Review
Article 4	G. Briggs, T. Williams, M. Scheutz	Enabling Robots to Understand Indirect Speech Acts in Task-Based Interactions

Association for Computing Machinery

Advancing Computing as a Science & Profession

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 May 2018.

For further information and to submit your paper, please visit **https://thri.acm.org**.

Association for
Computing Machinery

DOI:10.1145/3196943

Meet the candidates who introduce their plans—and stands—for the Association.

ACM's 2018 General Election

Please take this opportunity to vote.

THE ACM CONSTITUTION provides that our Association hold a general election in the even-numbered years for the positions of President, Vice President, Secretary/Treasurer, and Members-at-Large. Biographical information and statements of the candidates appear on the following pages (candidates' names appear in random order).

In addition to the election of ACM's officers—President, Vice President, Secretary/Treasurer—two Members-at-Large will be elected to serve on ACM Council.

Electronic Balloting Procedures. Please refer to the instructions posted at https://www.esc-vote.com/acm2018.

To access the secure voting site, you will need to enter your email address (the email address associated with your ACM member record) and your unique PIN provided by Election Services Co.

Paper Ballots. Should you wish to vote by paper ballot please contact Election Services Co. to request a paper copy of the ballot and follow the postal mail ballot procedures: acmhelp@electionservicescorp.com or +1-866-720-4357.

Postal Mail Ballot Procedures. Please return your ballot in the enclosed envelope, which must be signed by you on the outside in the space provided. The signed ballot envelope may be inserted into a separate envelope for mailing if you prefer this method.

All ballots must be received by **no later than 16:00 UTC on 24 May 2018.**

The ACM Tellers Committee will validate the computerized tabulation of the ballots. Validation by the Tellers Committee will take place at 14:00 UTC on **29 May 2018.**

Sincerely,

Gerald Segal
CHAIR, ACM ELECTIONS COMMITTEE

candidates for

PRESIDENT

(7/1/18 – 6/30/20)

JACK DAVIDSON
Professor of Computer Science
University of Virginia
Charlottesville, VA
U.S.A.

Biography

Education and Employment

- B.A.S. (Computer Science) Southern Methodist University, 1975; M.S. (Computer Science) Southern Methodist University, 1977; PhD (Computer Science) University of Arizona, 1981.
- University of Virginia, Professor, 1982–present.
- President, Zephyr Software, 2001–present.
- Princeton University, Visiting Professor, 1992–1993.
- Microsoft Research, Visiting Researcher, 2000–2001.
- Programmer Analyst III, University of Texas Health Science Center at Dallas, 1993–1997.

ACM and SIG Activities

- ACM member since 1975.
- ACM Publications Board Co-Chair, 2010–present; ACM Publications Board member, 2007–2010.
- ACM Student Chapter Excellence Award Judge, 2010–2017.
- ACM Student Research Competition Grand Finals Judge, 2011–2017.
- Associate Editor, ACM TOPLAS, 1994–2000. Associate Editor, ACM TACO, 2005–2016.

Member of SIGARCH, SIGBED, SIGCAS, SIGCSE, and SIGPLAN.

- SIGPLAN Chair, 2005–2007.
- SIGPLAN Executive Committee, 1999–2001, 2003–2005
- SGB Representative to ACM Council, 2008–2010.
- SGB Executive Committee, 2006–2008.

Awards and Honors

- DARPA Cyber Grand Challenge Competition, 2nd Place, $1M prize (2016)
- ACM Fellow (2008).
- IEEE Computer Society Taylor L. Booth Education Award (2008).
- UVA ACM Student Chapter Undergraduate Teaching Award (2000).
- NCR Faculty Innovation Award (1994).

Jack Davidson's research interests include compilers, computer architecture, system software, embedded systems, computer security, and computer science education. He is co-author of two introductory textbooks: *C++ Program Design: An Introduction to Object-Oriented Programming* and *Java 5.0 Program Design: An Introduction to Programming and Object-oriented Design*. Professionally, he has helped organize many conferences across several fields. He participated in the organization of several international summer schools including the International Summer School on Advanced Computer Architecture and Compilation for Embedded Systems, the inaugural Indo-U.S. Engineering Faculty Leadership Institute held in Mysore, India, and the First International Summer School on Information Security and Protection held in Beijing, China. Most recently he served as Program Chair for the HiPEAC 2018 held in Manchester, U.K.

Davidson's current research focuses on cyber security and societal computing. He is PI on two active research contracts to secure critical infrastructure and autonomous vehicles.

Statement

I joined ACM in 1975 as a student member. Our student group organized programming contests, planned chapter activities, wrote code just for fun, and discussed the awesome power of programs (that generate programs)*. Since then, I have had the privilege to contribute to ACM's mission in many capacities, most recently serving as co-Chair of ACM's Publication Board. I am honored to have been asked to stand for election as President.

A nomination statement typically discusses the challenges facing ACM and the candidate's plans to address those challenges. As I see it, ACM is a vibrant volunteer-led organization in excellent financial health. Each year thousands of volunteers plan and carry out a variety of activities and initiatives—organizing conferences, performing and publishing the very best research in the field, working to improve diversity in the computing profession, and developing new education programs. These activities are carried out on a solid financial footing. Non-profits normally seek financial reserves equal to one year of annual expenses. Our reserves are approaching twice that.

The most important challenges I believe ACM must confront are outward rather than inward. I advocate we marshal our ample resources and dedicated volunteers to help address the important challenges facing society posed by the powerful and pervasive digital technologies we are creating. In doing so, ongoing internal challenges—membership value, diversity, inclusiveness—can continue to be addressed—but addressed from a more relevant platform.

No doubt each of us could name a half-dozen computing technologies that are or will have a profound impact on society: AI, machine learning, cyber social networks, ubiquitous and invisible networks, cyber currencies, autonomous systems, wearable sensors, and cyber-enhancement of the body. The benefits of these technologies are undeniable.

Unfortunately, such technologies can have unanticipated negative consequences on basic human values of privacy, freedom, democracy, individual autonomy, and quality of life. These technologies and the resulting systems are complex and their integration into society cuts across geographic, cultural, gender, age, and socioeconomic boundaries. It is only through international, multidisciplinary efforts involving academia, industry, and government that these problems can be addressed.

ACM leadership is essential for initiating and facilitating these international collaborative efforts. Our involvement would, as a side effect, strengthen ACM by creating a sense of global unity on important problems that affect us all. By presenting a value proposition to the worldwide computing community that is compelling, meaningful, and relevant, we can become a more inclusive and diverse professional society.

Now is the time for ACM to use its resources and the energy of our volunteers to help address pressing societal problems posed by emerging computing technologies. I ask for your vote, support, and involvement.

candidates for PRESIDENT

(7/1/18 – 6/30/20)

CHERRI M. PANCAKE
Professor Emeritus and Intel Faculty Fellow
School of Electrical Engineering & Computer Science
Oregon State University
Corvallis, OR
U.S.A.

Biography

Cherri Pancake is Professor Emeritus and Director of the Northwest Alliance for Computational Science and Engineering (NACSE), an interdisciplinary research center known for software systems that analyze large-scale scientific data to yield results that "make sense" to decision-makers. She is a Fellow of ACM and IEEE.

Pancake started her career as an ethnographer conducting fieldwork in Guatemalan Indian communities, where she applied cross-cultural techniques to study social change. After earning a PhD in Computer Engineering, she leveraged her ethnographic expertise to address problems in computing. She was among the first worldwide to use ethnographic techniques to improve software usability, an approach which is now standard in the field. She also conducted much of the seminal work identifying how the needs of scientists differ from computer scientists.

She then turned to studying how "virtual collaborations"—interactions that span large interdisciplinary and physically distributed communities—differ from those where collaborators are physically co-located. Under her guidance, NACSE developed processes and software tools to make remote collaboration and data sharing fit naturally into typical patterns of scientific research and practice.

A member of ACM since 1982, Pancake has served in a wide variety of roles, including Vice President. Previously, she was Awards Co-Chair, an elected member of ACM Council, and area editor for *Communications of the ACM*. She also led two ACM/IBM industry advisory boards, chaired the Gordon Bell Prize and Fellows committees, and has held leadership roles in one of ACM's largest conferences since 1990.

Pancake's efforts were instrumental in creating SIGHPC (Special Interest Group on High Performance Computing) and she served as its first Chair. Under her leadership, it grew to over 1,000 members and achieved financial viability in the first year, setting two records for ACM. In 2015, she obtained a $1.5M endowment from Intel to establish the SIGHPC/Intel Computational & Data Science Fellowships, which to date have provided $15,000/year to 26 outstanding women and minority graduate students from seven countries. She is currently coordinating efforts to establish a new competition designed to attract students to computing by engaging them in data analysis and computation to address socially relevant problems.

Statement

It's exciting to be nominated for ACM president during a time of real change, within our organization and in our profession. I believe my experience in both has prepared me well for the role. I have served on ACM-wide committees and held the positions of Awards co-chair, Council member, and Vice President. Most of ACM's activities occur at the level of SIGs and boards; I will leverage my past work with conferences sponsored by four different SIGs, in editorial positions, and as founder of one of the newest SIGs to help identify areas for growth and renewal. Across the international community, I've held leadership and advisory roles in a number of research- and data-sharing collaborations, including the Protein Databank, the Long-Term Ecological Research Program, the National Biological Information Infrastructure, and the Network for Earthquake Engineering Simulation. Those experiences plus my unique background—coupling anthropology with computer engineering—have given me the broad perspective needed for the position of President.

I see great opportunities for ACM now that advanced capabilities, such as machine learning, location-aware computing, and wireless sensor networks, have become accessible to a broad spectrum of users in many fields. As the leading society for computing, ACM is uniquely positioned to be the "glue" that joins emerging practitioner communities with classical computer scientists. I believe ACM must proactively engage the new groups, providing conference and publication opportunities that will help drive advances in their fields.

The expansion of computing into other fields has also introduced new challenges for our profession. For example, in many settings it has become critical that computational results be reproducible. Bridging the gap between ease of computing and reproducibility requires patient experimentation. When we formed SIGHPC, one of our first actions was to join forces with the Publications Board and other SIGs to promote reproducibility through competitions, associating data with Digital Library publications, and acknowledgment of outside verification. As President, I hope to expand those efforts and extend them to other audiences.

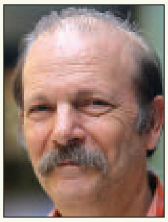
I also believe ACM should do more in response to the growing demand for computing professionals. As a field, we must attract new people from diverse backgrounds, not just to fill the pressing need but also to enrich the ideas and processes used in computing research, education, and practice. I've seen several ACM boards and SIGs make great strides to increase the diversity of their conferences, publications, and competitions, discovering that relatively simple changes can make a surprising difference. One of my goals as President is to ensure that these best practices are shared across ACM.

In this time of change, ACM needs a President with the experience and expertise to identify strategies that can broaden our base. With your help and support, I believe I can do that.

candidates for

VICE PRESIDENT

(7/1/18 – 6/30/20)



MOSHE Y. VARDI
Professor in Computational Engineering
Director, Ken Kennedy Institute
for Information Technology
Rice University
Houston, TX
U.S.A.

Biography

Moshe Y. Vardi is the Karen Ostrum George Distinguished Service Professor in Computational Engineering and Director of the Ken Kennedy Institute for Information Technology Institute at Rice University. He is the co-recipient of three IBM Outstanding Innovation Awards, the ACM SIGACT Goedel Prize, the ACM Kanellakis Award, the ACM SIGMOD Codd Award, the Blaise Pascal Medal, the IEEE Computer Society Goode Award, the ACM Outstanding Contribution Award, and two ACM Presidential Awards. He is the author and co-author of over 500 papers, as well as two books: *Reasoning about Knowledge* and *Finite Model Theory and Its Applications*.

He is a Fellow of the Association for the Advancement of Artificial Intelligence, the Association for Computing Machinery, the American Association for the Advancement of Science, the Institute for Electrical and Electronic Engineers, and the Society for Industrial and Applied Mathematics. He is a member of the U.S. National Academies of Science and of Engineering, the American Academy of Arts and Science, the European Academy of Science, and Academia Europaea.

He holds honorary doctorates from the Saarland University in Germany, Orleans University in France, UFRGS in Brazil, and University of Liege in Belgium. He served for a decade as the Editor-in-Chief of *Communications of the ACM*, and chaired the ACM Job Migration Taskforce. Vardi's research interests focus on automated reasoning, a branch of artificial intelligence with broad applications in computer science, including database theory, computational-complexity theory, multiagent systems, computer-aided verification, and teaching logic across the curriculum.

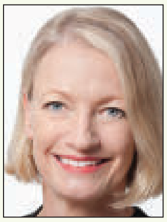
Statement

Serving as Editor-in-Chief of *Communications of the ACM* for about a decade has offered me a unique opportunity to get a very broad view of computing and of ACM. ACM is facing today a significant challenge, I believe; at the same time ACM has a unique opportunity to play a major societal role.

Challenge: A quick examination of ACM's annual report indicates that scholarly publishing is the financial mainstay of ACM. Publishing profits help ACM carry a wide area of activities. Yet, 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.

Opportunity: A profound shift in the public view of computing has taken place recently. Computing was traditionally viewed as a source of innovation, economic growth, good jobs, and cool gadgets. In the past few months, one reads in the mainstream media descriptions of cyberspace as "a dark and lawless realm where malevolent actors ranging from Russian trolls to pro-ISIS Twitter users could work with impunity to subvert the institutional foundations of democracy." Computing today is one of the greatest forces driving societal change, and computing professionals must accept their share of social responsibility. ACM is involved in several activities related to social responsibility. Yet, these efforts are dispersed and lack coordination. ACM must be more proactive in addressing social responsibility issues raised by computing technology. An effort that serves as a central organizing and leadership force within ACM would bring coherence to ACM's various activities in this sphere, and would establish ACM as a leading voice on this important topic.

candidates for
VICE PRESIDENT
(7/1/18 – 6/30/20)



ELIZABETH CHURCHILL
Director of User Experience
Google
Mountain View, CA
U.S.A.

Biography

Elizabeth Churchill is a Director of User Experience at Google. Her field of study is Human Computer Interaction and User Experience, with a current focus on the design of effective designer and developer tools.

Churchill has built research groups and led research in a number of well-known companies, including as Director of Human Computer Interaction at eBay Research Labs in San Jose, CA, as a Principal Research Scientist and Research Manager at Yahoo! in Santa Clara, CA, and as a Senior Scientist at PARC and before that at FXPAL, Fuji Xerox's Research lab in Silicon Valley.

Working across a number of research areas, she has published research, patented prototypes, and taught courses at a number of universities. She has more than 50 patents granted or pending, seven academic books, and over 100 publications in theoretical and applied psychology, cognitive science, human-computer interaction, mobile and ubiquitous computing, computer mediated communication and social media. In 2016, she received the Citris-Banatao Institute Athena Award for Executive Leadership.

The current Secretary/Treasurer of the ACM, Churchill served on the Executive Committee of the ACM's Special Interest Group on Computer-Human Interaction (SIGCHI), for eight years, six years of those as Executive Vice President and two as Vice President for Chapters. She has also held leadership committee positions on a number of ACM SIGCHI associated conferences. Churchill is a Distinguished Scientist and Distinguished Speaker of the ACM, and a member of the SIGCHI Academy.

Churchill earned her BSc. in Experimental Psychology and her MSc. in Knowledge Based Systems from the University of Sussex, U.K., and her PhD in Cognitive Science from the University of Cambridge, U.K. Her dissertation research focused on the design and development of Programmable User Models. After her PhD, she was a Postdoctoral Research Fellow at the University of Nottingham before leaving the U.K. and moving to industry in 1997.

Statement

I am honored to be nominated for ACM Vice President.

As the current Secretary/Treasurer of ACM, I am very aware of the need to serve our membership effectively through judicious investment in the right initiatives. As an industry applied research leader committed to education, I believe ACM has a central role in academic and practitioner development. Finally, I also believe ACM has an important leadership role to play when it comes to inclusiveness, equality, equity, and ethics in all computer sciences.

If elected, I will be a strong voice for deepening our efforts in the following areas:

- *Early-stage development and career support of our field's future leadership.* ACM membership currently skews toward mid- to late stage professionals. Initiatives focused on early career support, such as the Future of Computing Academy in 2017, will provide a solid foundation for growth and relevance for many years to come.
- *Community development through broader global outreach efforts.* Our membership is globally based, yet ACM is often mistaken as an 'American association' for computing sciences. Greater focus on relevant initiatives, programs, and promotions in the global arena will ensure ACM is known worldwide as '*the* association for computing sciences,' not just the American one.
- *Leveraging ACM's existing programs and platforms to underscore its place as the key, lifelong professional network for those involved in all aspects of computer science.* Drawing on the deep expertise of our membership whose work centers on social networking offers ACM an unrivaled opportunity to further develop the social connectivity of all its members.
- *Meeting increasing challenges faced by the CS and technological world.* Further developing our commitment to ethical, equitable computing education and application, and to the promotion of diversity and inclusiveness in CS education and practice, will benefit not only our members, but also society at large.

ACM is already a leader. From our Digital Library to our many conferences, symposia, and other events, we provide an essential knowledge platform for the theoretical and applied computing sciences, and an unparalleled opportunity for academics and practitioners to engage in lifelong learning and community.

It would be my privilege, as ACM Vice President, to work with ACM staff and volunteers to ensure this leadership continues.

candidates for

SECRETARY/TREASURER

(7/1/18 – 6/30/20)

YANNIS IOANNIDIS
President and General Director
"Athena" Research & Innovation Center
Professor of Informatics & Telecom
University of Athens
Greece

Biography

Yannis Ioannidis holds a PhD in Computer Sciences (Univ. of California–Berkeley, 1986), an MSc in Applied Mathematics (Harvard Univ., 1983), and a Diploma in Electrical Engineering (National Technical Univ. of Athens, 1982).

He is currently President and General Director of the "Athena" Research & Innovation Center in Athens, Greece (since 2011) and a professor of Informatics & Telecom at the Univ. of Athens (since 1997). Previously, he was a professor of Computer Sciences at the Univ. of Wisconsin–Madison (1986–1997).

His research interests include database and information systems, data science, recommender systems and personalization, and electronic infrastructures. His work is often motivated by data management problems that arise in the context of other scientific fields (Life Sciences, Cultural Heritage and the Arts, Physical Sciences). He has published over 150 articles in leading journals and conferences and also holds three patents.

Ioannidis is an ACM and IEEE Fellow (essentially both "for contributions to database systems, particularly query optimization"), a member of Academia Europaea, and a recipient of several research and teaching awards, including Presidential Young Investigator (1993), UW Chancellor's Teaching Award (1996), VLDB 10-Year Best Paper (2003).

An ACM member since 1983, he currently serves on the ACM Europe Council (since 2017), the SIG Governing Board Executive Committee (since 2012), and the ACM Publications Board as SGB liaison (since 2014). He has also served four-year terms as vice-chair and then chair of the Special Interest Group on Management of Data (SIGMOD). In 2017, he received the ACM SIGMOD Outstanding Contributions Award.

Ioannidis is the Greek delegate to the European Strategy Forum on Research Infrastructures (ESFRI) and a member of its Executive Board. He is also a member of the steering committee of the IEEE Int'l Conf. on Data Engineering, and has served on several other professional boards and committees, including the IEEE Technical Committee on Data Engineering and the VLDB Endowment Board of Trustees.

Statement

The very concept of a scientific society is being challenged these days. If honored to be elected as ACM Secretary/Treasurer, I will use my past experience as a volunteer in several roles to serve the community and help ACM maintain and further strengthen its current position of scientific leadership and financial stability. In this direction, I believe ACM should enrich and diversify its profile in at least three key dimensions: scientific areas of concern, membership, and conceptions of publication.

While remaining current on the purely technological advances in computing, ACM should significantly expand its scientific domain and become the home of all interdisciplinary areas that involve computing, possibly through strategic alliances with peer scientific and scholarly societies and appropriate joint activities. This will have the added benefit of attracting new members with non-traditional backgrounds, for example, computational or data scientists and practitioners.

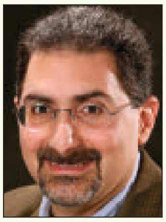
It is also important that ACM strengthens its role in computing education at all levels globally, starting with very young kids, so that they grow up well versed in algorithmic thinking (a fundamental skill) and possibly inspired to follow a relevant career in computing. In the long term, this should largely eliminate most ACM member underrepresentation based on gender, geography, or age.

Finally, in the new era of Open Science, as a top-quality publisher, ACM should be a pioneer again and help redefine scholarly communication and the entire research life cycle, under the principles of reproducibility and accountability. It should treat software and data as first-class publishable results, embed all provenance artifacts in a publication, and explore new review processes and access policies.

candidates for

SECRETARY/TREASURER

(7/1/18 – 6/30/20)



MEHRAN SAHAMI
Professor (Teaching) and Associate Chair for Education
Computer Science Department
Stanford University
Stanford, CA
U.S.A.

Biography

Mehran Sahami is a Professor (Teaching) and Associate Chair for Education in the Computer Science department at Stanford University. He is also the Robert and Ruth Halperin University Fellow in Undergraduate Education at Stanford. Prior to joining the Stanford faculty in 2007, he was a Senior Research Scientist at Google (2002–2007) and a Senior Engineering Manager at Epiphany (1998–2002).

Mehran is currently completing his second two-year term as Co-chair of the ACM Education Board and Education Council, helping to initiate and oversee educational activities for the ACM. He co-chaired the ACM/IEEE-CS joint task force on Computer Science Curricula 2013 (CS2013), which was responsible for creating curricular guidelines for college programs in Computer Science at an international level. In 2014, he received the ACM Presidential Award for his leadership of this effort. He also co-founded and served as the first General Chair of the ACM Conference on Learning at Scale, which has become an annual meeting (now in its 5th year) focused on interdisciplinary research at the intersection of the learning sciences and computer science. Additionally, he was co-founder and first Chair for the annual Symposium on Educational Advances in Artificial Intelligence (EAAI), now in its 8th year.

Mehran's research interests include computer science education, artificial intelligence, and Web search. He has published numerous technical papers, including the book *Text Mining: Classification, Clustering and Applications*. He has over 20 patent filings on a variety of topics including machine learning, Web search, recommendation engines in social networks, and email spam filtering that have been deployed in several commercial applications. He received the 2017 CIKM Test of Time Award, recognizing outstanding papers published 10 or more years ago that had a sustained impact on the research community.

He received his BS, MS, and PhD in CS from Stanford. And despite all that, he still hasn't figured out how to get his kids to brush their teeth at bedtime without a fuss.

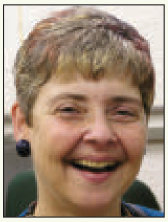
Statement

As a Life Member of ACM, I am honored to be nominated for the position of Secretary/Treasurer. As Co-chair of the ACM Education Board, I've served on the Extended Executive Committee of the ACM for nearly four years. That experience gives me a deep appreciation for the issues facing the organization and provides the opportunity to hit the ground running in the new capacity of Secretary/Treasurer.

My main goals are working to better serve the needs of the membership, specifically pursuing opportunities to push for more open access models for publications, increasing development of content relevant to practitioners, and more fully realizing ACM's mission to be a global association. Additionally, I am concerned with the enormous enrollment growth in college CS programs and am committed to exploring how ACM might be able to help educational institutions better address this issue. Of course, the challenge is to pursue these goals while ensuring the financial viability of the organization. By pushing ACM to remain relevant for younger academics and practitioners, we can help to extend the membership base and expand support for the organization more globally.

Another responsibility of the Secretary/Treasurer is chairing ACM's investment committee. I have over a decade of experience with investment stewardship, including as an investor and limited partner in several venture capital funds. As Secretary/Treasurer, I would help ACM continue to develop agreements with other organizations to pursue our mutual goals. Previously, I helped create agreements for joint projects between ACM and IEEE, AIS, and other computing societies. I look forward to continuing to serve as an ACM volunteer (in any capacity) and I appreciate your consideration. Thank you for reading this statement.

candidates for
MEMBERS AT LARGE
(7/1/18 – 6/30/22)



CLAUDIA BAUZER MEDEIROS
Professor of Computer Science
University of Campinas
Brazil

Biography

Claudia Bauzer Medeiros is full professor, Computer Science, at U. of Campinas (Unicamp), Brazil, with Brazilian and international awards for excellence in research, teaching, and work fostering the participation of women in computing. She is a Commander of the Brazilian Order of Scientific Merit, and holds two honorary doctorates from U. Antenor Orrego, Peru, 2007, and U. Paris IX Dauphine, France, 2015. For engaging women in IT, she earned the Google Brazil Award and the Grace Hopper Agent of Change Award.

Her research is centered on managing scientific data. In 1994, she created Unicamp's Laboratory of Information Systems, one of the first labs in Brazil dedicated to interdisciplinary data-intensive research. Since 1997, she has headed large multi-institutional projects in biodiversity, health, agriculture and environmental planning, involving universities in Brazil, Germany, and France.

One of the few Brazilians to become a Distinguished Speaker of ACM, she is a member of IEEE, SIGMOD, SIGSPATIAL and ACM-W and has served as ACM-W Latin America ambassador and SIGMOD liaison.

While President of the Brazilian Computer Society (2003–2007), she launched the first countrywide initiatives to draw women to computing, and liaised with funding agencies to create opportunities in CS research and education. She has served as member and/or chair of scientific evaluation panels in Brazil, for the Ministries of Education and of Science and Technology, and the São Paulo State Foundation (FAPESP).

Since 2013, she's represented FAPESP in a network of agencies from 12 countries, for research in the Social Sciences and Humanities. Since 2014, she's coordinated the FAPESP eScience research-funding program to foster data- and/or computing-intensive interdisciplinary research.

Statement

It is an honor to be nominated for Member at Large. I became an ACM member in 1983 as a PhD student, and have not ceased to discover the many opportunities it offers—for learning, research, professional growth, and networking. If elected, I would like to help expand some key activities along these lines. I look at ACM from at least three perspectives: as a (Latin American) academic; as an active player in creating and enforcing policies for universities and scientific societies; and as someone who has closely worked with and for several funding agencies in Brazil and abroad.

Information technology is all pervasive. ACM should find new ways to raise awareness of ethical issues associated with computing research and practices, not only among computing professionals (handled by its Code of Ethics), but also among those whose work requires some kind of computing skills. Data and algorithm ethics need to be better exploited, as do ethical concerns raised by other domains.

My eScience research and work for funding agencies taught me the advantages (and pitfalls) of interdisciplinarity. I would like to help promote mechanisms to foster this, both within computing fields and in our work with other domains. We should encourage interactions across SIGs, and open science initiatives. Cross-disciplinary collaboration should be fostered early, and nurtured throughout one's career—through advocacy, mentoring, and educational material.

Last but not least, there is still much to be done toward attracting women and minorities to computing, particularly in Latin America. ACM should further promote inclusiveness in the workplace and give more visibility to initiatives within ACM-W. To this end, cultural and social differences must be carefully analyzed and taken into account.

NENAD MEDVIDOVIĆ
Professor of Computer Science & Informatics
University of Southern California
Los Angeles, CA
U.S.A.

Biography

Nenad Medvidović is a professor in the Computer Science Department and in the Informatics Program at the University of Southern California (USC). Medvidović is the Founding Director of the SoftArch Laboratory at USC. He has previously served as Director of the USC Center for Systems and Software Engineering (2009–2013), Associate Chair for PhD Affairs in USC's CS Department (2011–2015), and Chair of the Steering Committees for the two premier conferences in his field: ICSE — International Conference on Software Engineering (2013–2015) and FSE — Symposium on the Foundations of Software Engineering (2015–2017). He has been the Program Chair for several conferences, including ICSE 2011. Medvidović has served as an Associate Editor of 10 different journals. He is currently the Editor-in-Chief of *IEEE Transactions on Software Engineering*, a flagship software engineering journal, as well as the Chair of ACM SIGSOFT.

Medvidović received his PhD in 1999 from the Department of Information and Computer Science at UC Irvine. He is a recipient of the National Science Foundation CAREER (2000) award, the Okawa Foundation Research Grant (2005), the IBM Real-Time Innovation Award (2007), the USC Mellon Mentoring Award (2010), and Orange County Engineering Council's Distinguished Engineering Merit Award (2018). Medvidović has over 200 publications in the area of software engineering research. Several of his publications have received Most Influential Paper (a.k.a. "Test of Time"), Best Paper, and Most Cited Paper awards. He is a co-author of a textbook on software system architectures. Medvidović is an ACM Distinguished Scientist and an IEEE Fellow.

Statement

ACM serves a broad constituency with divergent perspectives, interests, and needs. I have been involved with ACM and my home SIG—SIGSOFT—for almost 25 years in a number of capacities: as a volunteer, attendee, and presenter at ACM-sponsored conferences; in various conference organizing and overseeing roles; and as Chair of ACM SIGSOFT. During the past quarter-century, the computing community has grown and changed tremendously. Our field has become global and interdisciplinary. Even within SIGSOFT, our flagship conference, International Conference on Software Engineering (ICSE), was held for the first time outside the West only 10 years ago. Since then, we have gone to China, South Africa, India, and Argentina.

ACM's future will be shaped by the growth and globalization of computing. As an ACM Council Member at Large, I will dedicate my energy to help ACM in its efforts to embrace and facilitate these trends. I will rely on my experience in expanding the reach of SIGSOFT into different geographic regions and different segments of the software engineering community. As illustrations of this experience, I chaired or participated in "Warm-up Workshops" for ICSEs held in South Africa and Argentina, whose shared objective was to expose the local computing communities to SIGSOFT; I served as Program Co-Chair of last year's Indian Software Engineering Conference; and in my role as SIGSOFT Chair, I oversaw the creation of CSoft, SIGSOFT's Chinese Chapter. I have also been active in efforts to engage a larger cross-section of the professional software community in the activities of SIGSOFT, for example, through the appointment of a SIGSOFT Industry Liaison. I believe these experiences make me well positioned to participate in and impact ACM Council's important work.

candidates for

MEMBERS AT LARGE

(7/1/18 – 6/30/22)

PJ NARAYANAN
Professor and Director
IIIT Hyderabad
India

Biography

Education: BTech in CSE (1984), IIT Kharagpur; PhD in CS (1992) University of Maryland, College Park.

Employment: Lipi Indian language word-processor group of CMC Ltd.; Research Faculty Member, Robotics Institute of CMU (1992–1996); Head of Vision and VR group, Centre for Artificial Intelligence and Robotics, DRDO; (1996–2000), Faculty member, IIIT Hyderabad (from 2000).

Contributions: Narayanan's PhD thesis was on parallel processing for Computer Vision. At CMU he built Virtualized Reality, the first system to capture 3D representations of dynamic events using cameras. He built a VR resource center at CAIR, resulting in applications of VR in DRDO. The IIIT Vision group is among the world's largest, with over 100 researchers. Early work on using the GPU for Vision and other tasks has influenced the GPU-led deep learning revolution.

At IIIT, Narayanan was the first postgraduate coordinator and the first Dean of Research. He was appointed Director of IIIT in 2013. The institute has since established stronger connections with industry and vital engagements with startups. TCS Foundation endowed a Kohli Centre on Intelligent Systems, the largest AI group in India, at IIIT in 2015.

Narayanan was General Chair of the 2nd Indian Vision conference (ICVGIP) in 2000 and the Program Chair of ICVGIP2010 and ACCV2006. He was an SPC member or Area Chair of IJCAI2007, ICCVs (2007, 2011, 2015), CVPR 2017; ACCVs (2007, 2009, 2010, 2018), among others. He was on the JPDC Editorial Board till 2017.

Narayanan helped establish ACM in India as the founding Co-Chair and the first elected President, 2012–2014. He now leads the ACMI Research Board and is active in creating a computing community within India. He has also been on several committees related to research and education in India.

Statement

I started volunteering for ACM from 2009 as a founding Co-Chair of ACM India Council, its first elected President, and the chair of its Research Board. We set out to make ACM India the voice of the Indian computing community. We set up an ACM India Dissertation award, a survey on Indian PhD production, an annual event attended by Turing Award Laureates as well as hundreds of students, a student travel-support scheme for conferences abroad, an annual Research Summit with Microsoft Research, and several education initiatives. ACM membership tripled in India from 2009, to become the second-largest country by membership.

The impact of computing on life continues to grow, with AI potentially enhancing and disrupting unimaginable aspects of life everywhere. The non-Western geographies will have greater roles in coming years, with the growing access to computing, communication, and social media. The positive and detrimental impacts of computing need to be understood also from a point of view of impoverished populace, inadequate resources, and deficient governments. Technology can be a strong force to promote equity and to lessen the gap. Teachers, researchers, and professionals should keep in mind the human and social impact of each advancement. ACM will need to enhance and diversify its activities to enhance its influence in such a future. I will try to make this happen from the Council.

Several experiences influence my professional outlook: research roles in USA/India; teaching, especially as the head of a top institution, in India; and voluntary activities for ACM and different academic/government bodies. I believe my experiences can greatly help ACM traverse the future with greater balance. I pledge my efforts toward a more relevant ACM.

THEO SCHLOSSNAGLE
Founder and CEO
Circonus
Fulton, MD
U.S.A.

Biography

Theo Schlossnagle has spent the last 20 years applying computer science to pressing problems in industry. He founded four companies all grounded in large-scale distributed systems technology.

Theo studied at The Johns Hopkins University where he received a BS in Computer Science in 1997 and a MSE the following year related to his graduate work. In 2003, he left academia for industrial pursuits prior to completing his doctorate.

Beginning in 1996, Theo began participating in various open source communities including the Apache Software Foundation and in 1999 began a career in public speaking on topics both technical and professional. He has contributed significantly to over 100 open source projects and shared his experience with industry peers through over 200 speaking engagements.

Theo authored *Scalable Internet Architectures*, (Sams) and wrote chapters for *Web Operations* (O'Reilly) and *Seeking SRE* (O'Reilly).

Having founded four engineering-led organizations, his perspective on the computing profession is both varied and well informed; a perspective formed by operating some of the largest systems architectures on Earth, on-call rotations as an Site Reliability Engineer, developing both open and closed software systems, hiring engineering staff, mentoring, and guiding professional development of staff.

Statement

The ACM is dear to me; it represents the industry I love and those that are positioned to build the technology underlying our future. With eight years of exposure to academia and 20 years of intense, entrepreneurial participation in industry I feel I have a grounded perspective on how best the ACM can serve its members.

My experience as co-chair of ACM's *Queue* and participation on the ACM Practitioners Board provides immersion in the parts of ACM that directly touch the largest portion of its membership: the practitioner. ACM must represent the practicing computer scientist in all of their various forms. As ACM Member at Large, I will aim to influence decisions to consistently align them with the needs of practitioners of today and tomorrow.

ACM

ON A MISSION TO SOLVE TOMORROW.

Dear Colleague,

Without computing professionals like you, the world might not know the modern operating system, digital cryptography, or smartphone technology to name an obvious few.

For over 60 years, ACM has helped computing professionals be their most creative, connect to peers, and see what's next, and inspired them to advance the profession and make a positive impact.

We believe in constantly redefining what computing can and should do.

ACM offers the resources, access and tools to invent the future. No one has a larger global network of professional peers. No one has more exclusive content. No one presents more forward-looking events. Or confers more prestigious awards. Or provides a more comprehensive learning center.

Here are just some of the ways ACM Membership will support your professional growth and keep you informed of emerging trends and technologies:

- Subscription to ACM's flagship publication ***Communications of the ACM***
- Online books, courses, and videos through the **ACM Learning Center**
- Discounts on registration fees to ACM Special Interest Group conferences
- Subscription savings on specialty magazines and research journals
- The opportunity to subscribe to the **ACM Digital Library**, the world's largest and most respected computing resource

Joining ACM means you dare to be the best computing professional you can be. It means you believe in advancing the computing profession as a force for good. And it means joining your peers in your commitment to solving tomorrow's challenges.

Sincerely,

Vicki L. Hanson
President
Association for Computing Machinery

Association for
Computing Machinery

Advancing Computing as a Science & Profession

BLOG@CACM

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

twitter

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

DOI:10.1145/3193752 http://cacm.acm.org/blogs/blog-cacm

Commenting on Code, Considering Data's Bottleneck

Edwin Torres considers the enduring value of code comments, while Walid Saba wonders if we have overreacted to the knowledge acquisition bottleneck.

Edwin Torres

Why Code Comments Still Matter

http://bit.ly/2FglIP9
February 26, 2018

In computer science, you are taught to comment your code. When you learn a new language, you learn the syntax for a comment in that language. Although the compiler or interpreter ignores all comments in a program, comments are valuable. However, there is a recent viewpoint that commenting code is bad, and that you should avoid all comments in your programs. In the 2013 article *No Comment: Why Commenting Code Is Still a Bad Idea*, Peter Vogel continued this discussion.

Those who believe commenting code is a bad idea argue that comments add unnecessary maintenance; when code changes, you must also modify comments to keep them in sync. They argue it is the responsibility of the programmer to write really obvious code, eliminating the need for comments. Although these are valid reasons to avoid commenting code, the arguments are simplistic and general; comments are *necessary* for a variety of reasons:

1. Not all programmers can write really obvious code. Beginning programmers are just happy to write a correct program; they are still mastering the craft. Even experienced programmers write sloppy code. Programs are unique like fingerprints, so judging whether code is obvious is a subjective call.

2. It can be tedious to comment too much, but some comments are like titles and subtitles in articles; they guide, provide context, and convey overall meaning.

3. Comments are not just for code; they can document important program information such as author, date, license, and copyright details.

4. Some programming languages are cryptic, like the Glass programming language. This sample program (http://esolangs.org/wiki/Glass#Fibonacci_sequence) is hard to decipher, but prints a Fibonacci sequence. Is the meaning of this program clear to you? It may be possible to write it in a more obvious way, but a comment could convey its meaning.

5. Some companies require employees to comment their code. Google's programming style guides specify how to write comments in programming languages like Java, JavaScript, and C++.

6. Specialized comments allow tools like javadoc, JSDoc, and apiDoc to generate professional, thorough, and consistent documentation for programs.

7. Comments can be placeholders for future work, a useful way to create an outline for a large program. The Eclipse Integrated Development Environment (IDE) creates a TODO comment when it generates a main method, a reminder to add the starting code of a program.

Commenting may be tedious or overwhelming, but it is valuable in many situations. Even if you think you write obvious code, try reading your code months or years later; will it still obvious to you, or would you wish for comments?

Comments

*A key characteristic of comments is with respect to narration, as Ward Cunningham has pointed out. It can be important to distinguish what the code is *for*, not just what it is, and what the key assumptions and constraints might be. It is valuable to develop a grasp for what the requirements are, and code is rarely a substitute for that.*

—Dennis Hamilton

Dennis — That is a good point. There are times when you just need a quick overview of the code, without spending time to trace through it. Comments help here, assuming they are correct.

—Edwin Torres

There are many things we agree on. I should, for example, point out that my objection is to comments "in" code, not to comments at, for example, the start of a method, that include the name of the author, date created, and so on (though often, source control can automate that work).

I also definitely agree with you that code at the start of a method should describe what the method is "for"—why the method exists or was written. This is something even really obvious code often cannot communicate. At best, really obvious code can communicate the "how" of what the code does (though the name of the method can sometimes help address what this code is "for").

We even agree about the need to comment cryptic code. I will, however, suggest that the problem should not be first addressed by writing a comment: it should be first addressed by writing obvious code. If there is a bug in that Fibonacci generator or there's a need to enhance it, clear code will help you find the bug or enhance the code in a way that a comment cannot. I suggest we consider a comment, in this scenario, as an apology from the original programmer to the next one: "I did the best I could but, for various reasons, I still ended up with this unfortunate code. Here's what I can do to help."

I like the idea of comments as headings to guide a programmer through the code. As a part-time technical writer, that especially appeals to me. Of course, I'm going to suggest those comments (like headings) be only two or three words in length and, perhaps, evidence that this method should be refactored into several methods with their names reflecting those titles. But, in many cases, I can see that being overkill.

In fact, I will disagree with you in only one place: the idea that a programmer who can't write obvious code is, for some reason, capable of writing a comment that is (a) obvious, (b) accurate, and (c) complete. We hire programmers, after all, for their ability to write code, not for their ability as technical writers. And if the comment isn't accurate, well, to paraphrase "The Elements of Programming Style," the code and its comments provide two descriptions of the processing logic. If they disagree, only the code is true. At this point, code that goes beyond describing what a method is for creates a maintenance burden of fixing the comments to keep them in line with the code. Programmers are busy enough.

—Peter Vogel

Peter — My goal was to highlight some additional needs for comments. I agree that the "code doesn't lie." Also, too many comments can be overwhelming and distracting. I find it interesting that a discussion on comments even exists today. Who would've thought?

—Edwin Torres

*That's a great list of reasons to comment, to which I would add one more: what is obvious to *you*, the author of the code, probably isn't obvious to *me*, the reader. If you've been working on the feature you are building for the last week, you have spent that week building a mental model of that area of the problem domain and its mapping onto the software system. I do not have that model. Developers should try to empathize with the developer who understands software, but is new to *this problem*, and help improve their understanding.*

—Graham Lee

Graham — Great point. One of my earliest lessons in programming was that it is much harder to change someone else's program than create your own. When used effectively, comments can help here.

—Edwin Torres

Walid Saba

Did We Just Replace the 'Knowledge Bottleneck' With a 'Data Bottleneck'?

http://bit.ly/2tdSHfS

February 26, 2018

One of the main reasons behind the quantitative and data-driven revolution that took artificial intelligence (AI) by a storm in the early 1990s was the brittleness of symbolic (logical) systems and their never-ending need for carefully crafted rules. The rationale was that there is a knowledge acquisition *bottleneck* in the quest to build intelligent systems. The new cliché? Let the system 'discover' the logic/rules by crunching as much data as you can possibly get your hands on. With powerful machine learning techniques, the system will 'discover' an approximation of the probability distribution function and will 'learn' what the data is, and what it means, and will be ready for any new input hereafter. It all sounded good; too good to be true, in fact.

Notwithstanding the philosophical problems with this paradigm (for one thing, that induction is not a sound inference methodology—outside of mathematical induction, that is), in practice, it seems that avoiding the knowledge acquisition bottleneck has not resulted in any net gain. In the world of data science, it seems data scientists spend more than half of their time not on the science (models, algorithms, inferences, etc.), but on preparing, cleaning, massaging, and making sure the data is ready to be pushed to the data analysis machinery—whether the machinery was SVM, deep neural networks, or what have you.

Some studies indicate data scientists spend almost 80% of their time on preparing data, and even after that tedious and time-consuming process is done, unexpected results are usually blamed by the data 'scientist' on the inadequacy of the data, and another long iteration of data collection, data cleaning, transformation, massaging, and more, goes on. Given that data scientists are some of the most highly paid professionals in the IT industry today, isn't 80% of their time on cleaning and preparing the data to enter the inferno something that should raise some flags—or, at least, some eyebrows?

Such techniques, even after the long, tedious process of data cleaning and data preparation, still will be vulnerable. These models can be fooled by data that is similar, yet it will cause these models to erroneously classify them. The problem of adversarial data is getting too much attention, without a solution in sight. It has been shown that any machine learning model can be attacked with adversarial data (whether an image, an audio signal, or text) and can make the classifier decide *anything the attacker wants the classification to be*, often by changing one pixel, one character, or one audio signal—changes otherwise unnoticeable for a human.

Maybe not everything we want is in some data distribution? Maybe we are in a (data) frenzy? Maybe we went a bit too far in our reaction to the knowledge acquisition bottleneck?

Edwin Torres is a full-time software engineer at The MITRE Corporation and an adjunct professor of computer science at Monmouth University. **Walid Saba** is Principal AI Scientist at Astound.ai, where he works on Conversational Agents technology.

Science | DOI:10.1145/3193743

Gregory Mone

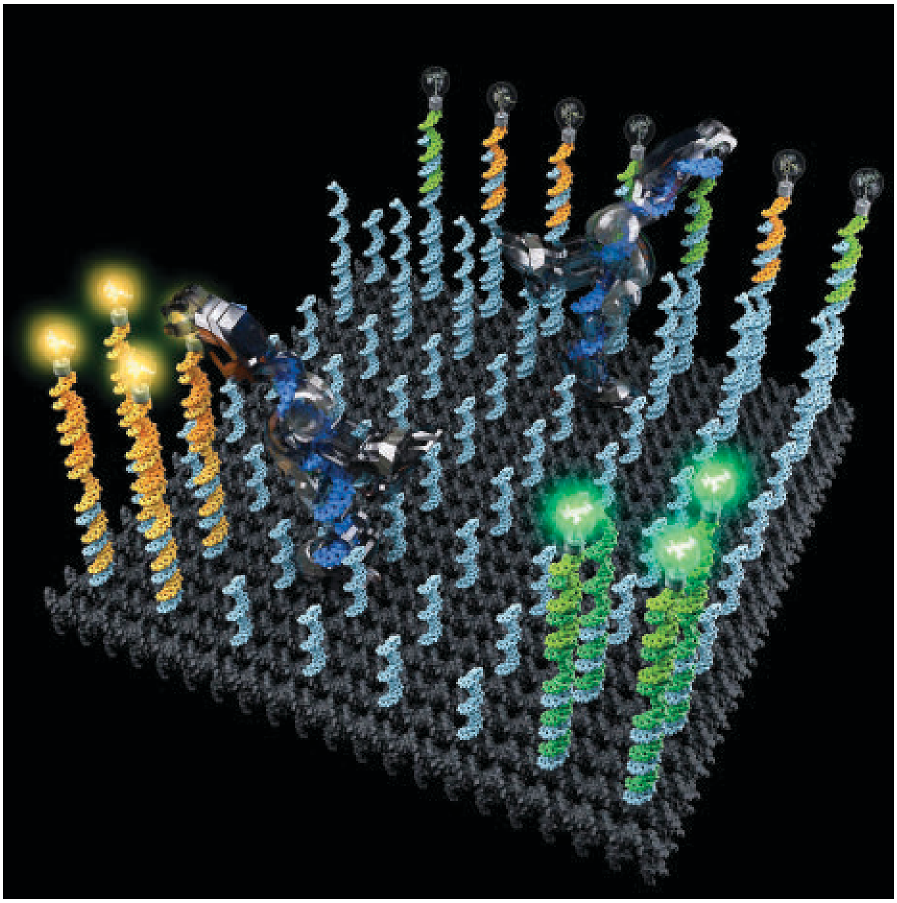
Shrinking Machines, Cellular Computers

Scientists are using DNA and RNA to build the world's tiniest robots and computing devices.

Since research in synthetic biology began nearly two decades ago, the field has expanded beyond its original mandate of using engineering principles to study and manipulate cells. Today, scientists are building biological computers and DNA-based robots that can carry out logical operations and complete tasks.

These miniscule machines look nothing like laptops or Roombas. Yet, algorithms still guide the robots through tasks, and the biological computers funnel inputs through logic gates. While a standard circuit works with electrical currents, though, the inputs in the biological version are biochemical signals triggered by presence of a protein or pathogen. The outputs, in turn, are another set of biochemical signals that trigger cellular responses, such as the activation of a gene.

The potential applications vary widely, ranging from reprogramming immune cells to fight infections without inducing harmful side-effects, to triggering molecular robots in trash to accelerate decomposition. However, scientists caution that biological computers and robots are still in their early stages of development, in part because



Conceptual illustration of two DNA robots sorting cargo on a DNA origami surface by transporting fluorescent molecules from initially unordered locations to separated destinations.

IMAGE BY DEMIN LIU, COURTESY OF CALIFORNIA INSTITUTE OF TECHNOLOGY

the cellular environment is such a challenging space. "You really need to be an engineer to design them correctly," says Massachusetts Institute of Technology (MIT) synthetic biologist Christopher Voigt. "You have to understand the molecular biology, but you have to think like an engineer."

Computing Inside the Cell

While Voigt and many other scientists prefer to use DNA as the building blocks of their biological computers, Arizona State University bioengineer Alexander Green and his colleagues at Harvard University's Wyss Institute for Biologically Inspired Engineering have been building RNA-based circuits. In their work, the design of the system is coded into DNA, which is then inserted into bacteria as a ring of DNA called a plasmid. From here, the cell takes over, essentially constructing the computer by transforming the DNA into what Green calls gate RNA.

These gate RNAs are folded in such a way that they only interact with the cell's ribosome to produce proteins when activated by additional strands. When one of these strands (the input) latches onto a gate RNA, the hybrid pair then instructs the ribosome to produce a glowing protein (the output).

Green refers to the system as a ribocomputer, and envisions multiple applications, including a future version that could be used to detect viruses even if they mutate rapidly. For example, if the gate RNA were programmed to respond to two different types of viral RNA (the inputs) associated with the Zika virus, and to trigger the ribosome to generate a glowing signal (the output) in the presence of one or the other, then the ribocomputer would be using the biological equivalent of an OR gate, since either one strand of viral RNA or the other would stimulate the output.

The most complex ribocomputer Green and his colleagues have devised so far can carry out 12 logic operations: five AND, five OR, and two NOT. In this case, a variety of RNA inputs can interact with the gate RNA. An AND operation is triggered when two complementary inputs are present, for example. Another set of RNA strands effectively prevents other inputs from interacting with the gate RNA, functioning as a NOT gate because it shuts down activity. "We have developed a very programmable way to use RNA to do simple computations in living cells," says Green. "We're basically enabling cells to make very logical decisions."

"You really need to be an engineer to design [biological computers and robots] correctly. You have to understand molecular biology, but you have to think like an engineer."

DNA Robots

Scientists building biological robots are making difficult decisions of their own in determining how to construct molecular machines that move and manipulate objects. For example, chemist Peter Allen of the University of Idaho and colleagues Andrew Ellington of the University of Texas, Austin, and Cheulhee Jung of Korea University, first built a two-legged, DNA-based robot that could walk across a surface littered with strands of DNA.

Each of the robot's legs was designed to be complementary to these anchor pieces of DNA—the particular sequences of Adenine, Thymine, Cytosine, and Guanine molecules on each were selected to bind to each other. For the robot to move, its legs first had to latch onto neighboring strands. One would unbind and attach to another strand, then the next would follow. As each of the legs completed this action, releasing one strand and binding to a neighbor, the robot walked. The problem with this approach, Allen explains, is that both strands could abandon their grip at the same time. "There was a chance that both legs could detach and the walker could float away," Allen says.

This glitch prompted Allen's colleague, Jung, to devise a single-legged DNA walker. The scientists took microscopic plastic spheres and covered the surface with 10,000 strands of synthesized DNA. As with the previous scenario, the walker moved along the surface by latching onto these tiny strands of DNA, then transferring to the next one like a snake moving between tree branches. As the robot attached itself to each DNA strand, the bond triggered another reaction that caused a molecule to fluoresce. When the walker covered enough of the surface and interacted with enough of the DNA strands, its fluorescence was bright enough to be detected with a microscope.

Allen envisions building robots that lie dormant until they come into contact with certain molecules or pathogens, which then spur them to move and trigger the fluorescent signals. "You could put your DNA robots into a vial of blood and it would either light up or not," he says, "and you'd know if the pathogen was there or not."

Random Walks

Calling these machines 'robots' in their current form is slightly misleading, Allen concedes, but recent advances suggest they eventually will be able to complete the kind of tasks normally associated with large-scale robots; they just might not do so in the same way. For example, bioengineer Lulu Qian of the California Institute of Technology and her colleagues recently demonstrated a DNA robot capable of sorting cargo. Given such a task, a standard electromechanical robot would probably explore its space, pick up and attempt to recognize each item, deposit an item in the right place, and then search for the next one. "Obviously, it would be very hard to program all that 'intelligence' into a single molecule," Qian explains.

So her group designed a robot that completed the sorting task by following a random walk algorithm, with which the robot does not remember where it has been or recognize the cargo. In the experiment, the DNA robot moved on a substrate covered with about 100 strands of DNA, dubbed pegs. The cargo items, fluorescent molecules, were linked to different pegs, and the robot's job was to find them, pick them up, and deliver them to the appropriate drop-off point.

The robot consisted of a leg with two feet (each a segment of a strand of DNA) and an arm with a hand. The segment of DNA on each foot was synthesized to latch itself to the DNA on the pegs. When one took hold, the other moved freely until it grabbed onto a neighboring strand of DNA; then the previous leg released and searched for its next mooring.

The robot's hand was designed to bind to strands of DNA attached to the fluorescent molecules, so when the robot bumped into a piece of cargo, it would grab the molecule and continue its random walk. Eventually, the robot would come across the drop-off point. Another segment of the cargo strand served as an identification tag, almost like a Universal Product Code. The drop-off point recognized that segment, latched on, and took the cargo strand and the fluorescent molecule attached to it. The robot then continued its random walk.

The entire experiment, which involved sorting six cargo items, required 300 steps and 24 hours, so the work is not exactly designed for holiday fulfillment operations in an Amazon warehouse. Yet Qian says there are several important, transferrable lessons coming from the experiment. The first is that complex tasks, like cargo sorting, can be accomplished with simple algorithms. "The simpler the algorithm," she notes, "the more likely it can be carried out by simple molecules."

The other key demonstration is the importance of modularity in building these robots. Before the current experiment, Qian's group demonstrated the viability of the walking robot using only one leg and two feet. "We then showed that adding an arm and a hand segment, without any changes to the leg and foot segments, allowed the robot to pick up and drop off cargos while moving around in random directions," she explains. "This proof-of-concept demonstration opens up future possibilities for developing additional building blocks that can be added to the toolbox of DNA robots."

Eventually, that expanding toolbox could be used to develop new applications. While the scientists are hesitant to go into too much detail about these applications, given the early stages of the research, Allen observes that the potential is tremendous: "The idea of being able to assemble matter at the atomic or molecular level and have control over it at a rational level, that's extremely appealing."

Further Reading

Cameron, N.E., Bashor, C.J., and Collins, J.J.
A brief history of synthetic biology. *Nature Reviews Microbiology*, May 2014.

Jung, C., Allen, P.B., and Ellington, A.D.
A stochastic DNA walker that traverses a microparticle surface. *Nature Nanotechnology*, February 2016.

Thubagere, A.J., Li, W., Johnson, R.F., et. al.
A cargo-sorting DNA robot, *Science*, Vol. 357, Issue 6356.

Nielson, A.A.K., Der, B.S., Shin, J., et. al.
Genetic circuit design automation, *Science*, Vol. 352, Issue 6281.

Gregory Mone is a Boston-based science writer and the co-author, with Bill Nye, of *Jack and the Geniuses: At the Bottom of the World.*

Security

Quantum Computing: The End of Encryption?

Security researchers warn that hackers are gobbling up encrypted data and waiting for the day when quantum computers will easily break their encryptions.

"Intelligence agencies are already recording massive amounts of encrypted data sent over the networks in the hope to successfully decrypt them with powerful quantum computers in a few years," says Tim Guneysu, chair for security engineering at Germany's Ruhr University Bochum.

High on the radar of the data thieves: trade secrets, health records, criminal records, and any other sensitive data hackers believe they will be able to sell, trade, or leverage in the quantum era.

"Think of the secret recipe for Coca-Cola, or blueprints for a supersonic plane," says Tanja Lange, chair of Cryptology at Eindhoven University of Technology in the Netherlands. "Trade secrets are often held close by companies and never published or patented."

Also, "Identifying dissidents and decrypting their communication will be worthwhile to some regimes, even years later," says Lange.

Such nightmare scenarios—and many more—are Lange's stock in trade. She leads PQCRYPTO, a European research consortium of 11 universities and companies assembled by the European Commission and charged with developing a preventative solution to the looming threat of widespread data theft.

PQCRYPTO operates under the assumption that many of us are likely to become victims of technology's ongoing success if a solution to security in the age of quantum computers is not found. The group anticipates encryption methods currently considered impenetrable by individuals, companies, and many governments—including the RSA public key cryptosystem, and elliptic-curve cryptography (ECC)—could become child's play to decrypt once quantum computers become hacker tools.

"RSA and ECC belong to the class of asymmetric cryptography that is known to be broken by tomorrow's quantum computers," Guneysu says. "Hence, for both schemes, we need replacements as soon as possible."

Lange agrees. "Sadly, none of the currently used public-key crypto—e.g., in https—is safe."

Researchers have made some progress on alternative encryption technologies they believe could defeat quantum computers. Granted, they may not have quantum computers to work with, but they are able to extrapolate how quantum computers can neutralize today's encryption, and they have come up with alternative encryption methods for the quantum era.

"While we do not have big, scalable quantum computers yet, it is clear what operations they can execute," Lange says. "When analyzing the security of proposed cryptosystems, we take these extra operations into account."

IT security researchers the world over began engaging in a friendly competition sponsored by the U.S. National Institute for Standards and Technology (NIST) late last year to develop encryption alternatives. Lange says 69 new encryption methods have been submitted so far.

These new encryption methods have withstood the scrutiny of other researchers, while another seven have been shown to have flaws, and still other methods submitted are considered questionable, according to Lange.

Yet even methods that have withstood extreme vetting have a problem: preliminary tests indicate it takes longer to transmit data over the Internet using these newer encryption methods.

The remaining challenge, Lange says, is for researchers to develop a new encryption method that is both bulletproof and practical.

"The NIST competition is something that keeps the community busy right now, and on the attack and implementation side, this is very much an ongoing project," Lange says.

Adds Daniel Gauthier, a professor of physics at The Ohio State University exploring secure Internet communications via quantum key distribution, "We really need to be thinking hard now of different techniques that we could use for trying to secure the Internet."

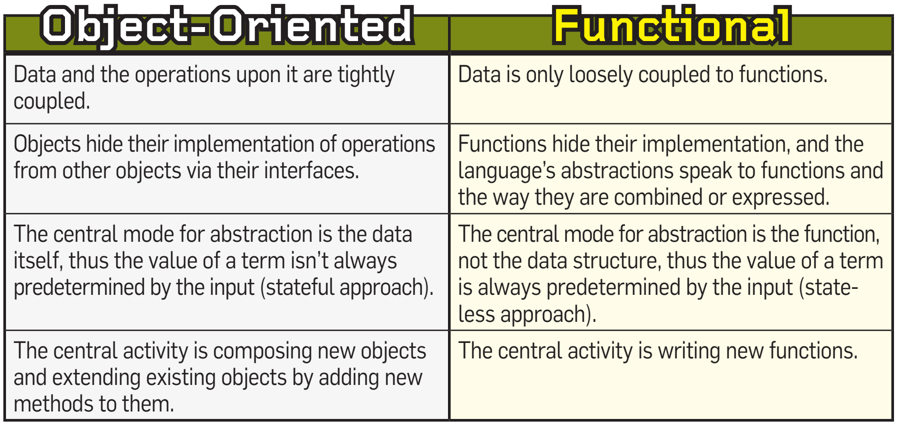
—Joe Dysart is an Internet speaker and business consultant based in Manhattan, NY, USA.

Technology | DOI:10.1145/3193776

Neil Savage

Using Functions for Easier Programming

Functional programming languages automate many of the details underlying specific operations.



Object-Oriented	Functional
Data and the operations upon it are tightly coupled.	Data is only loosely coupled to functions.
Objects hide their implementation of operations from other objects via their interfaces.	Functions hide their implementation, and the language's abstractions speak to functions and the way they are combined or expressed.
The central mode for abstraction is the data itself, thus the value of a term isn't always predetermined by the input (stateful approach).	The central mode for abstraction is the function, not the data structure, thus the value of a term is always predetermined by the input (stateless approach).
The central activity is composing new objects and extending existing objects by adding new methods to them.	The central activity is writing new functions.

TABLE CONTENT COURTESY OF BELATRIXSF.COM/BLOG

As computers become more powerful and the programs that run them grow more complex, programmers are increasingly trying to make their lives easier by turning to an idea that dates to the early days of computer languages, an approach called functional programming.

"Functional programming's on a long steady burn, starting 30 or 40 years ago," says Simon Peyton Jones, a researcher at Microsoft Research in Cambridge, U.K., where he focuses on the functional language Haskell.

Programming languages break down into broad categories. There are imperative languages, which say, "do this, then do that," specifying a series of steps to accomplish a task. Functional languages, on the other hand, rely on functions, which are mathematical operations.

"A function is just basically a piece of an algorithm," says Iavor Diatchki, a senior research and development engineer at Galois, a software company in Portland, OR. "It's something that you can give some inputs and it computes some outputs." For instance, "plus" is a basic function; it says to take two integers and combine them to produce one integer as a result.

Functional languages operate at a higher level of abstraction, automating a lot of the details that underlie a particular operation. That makes it easier to write programs quickly. Years ago, when computers were slower, that ease came with a cost, Diatchki says; the program's default steps were not always the most efficient, and a programmer could make it run better by taking the time to tweak the details. That has changed. "Computers are a lot faster, so things like that don't matter all that much," says Diatchki, who argues that making better use of a programmer's time has become more important. "Also, the kind of software we write tends to be a lot more complicated, so some of these small performance issues are not nearly as important. What's important is that you manage to put all the pieces together and get the program to work."

Higher-level languages are more productive, says Sergio Antoy, Textronics Professor of computer science at Oregon's Portland State University, in the sense that they require fewer lines of code. A program written in machine language, for instance, might require 100 pages of code covering every little detail, whereas the same program might take only 50 pages in C and 25 in Java, as the level of abstraction increases. In a functional language, Antoy says, the same task might be accomplished in only 15 pages.

Additionally, the less a developer writes, the less opportunity he has to include something that causes a problem. "As you remove details, you remove the potential for error," Antoy says. Higher-level languages also lead to programs that are easier to modify if the underlying machine architecture changes.

Another important aspect of functional programming is that functions do not change the data with which they work. Having immutable data makes it easier to do parallel processing. If two processors are manipulating the same data at the same time, one processor may alter the input to the other; if that happens in the wrong order or at the wrong time, the results could be incorrect. In functional programming, each processor would only need to read the original data, which would remain unaltered.

The fact that the computational state of the system does not change makes it easier to understand what exactly a program is doing, says Antoy. "It's more difficult to reason about computation in which there is a state," he says. Object-oriented imperative languages such as C, Java, or Python change their state as they run. "If you have a computation in which you have variables that can change over time, when you reason about this you don't know what the value of the variable is at that time, so it's like shooting a moving target. You want to reason about something and things keep changing."

Because the state does not change, functional languages are attractive for programs that require a high level of security, such as those used by financial institutions, where functional programs are becoming increasingly popular, says Philip Wadler, a professor of theoretical computer science at the University of Edinburgh in the U.K. and one of the creators of Haskell. "You need to be able to write a program pretty quickly and still have high assurance it's going to do what you expect," Wadler says. For instance, the New York-based international trading firm Jane Street Capital uses Ocaml, a dialect of ML, one of the earliest functional languages.

Wadler is also a senior research fellow for IOHK, a Hong Kong-based company designing cryptocurrency, a digital form of money of which bitcoin is the most well-known example. The ease of creating and validating programs written in functional languages makes them than appealing for such applications, Wadler says. "There's a huge amount of money that's at stake there."

Matter of Opinion

Just what counts as a functional language can be open to interpretation. "Haskell and ML are widely considered to be functional. Languages like C++ or C and even Java are considered to be imperative," Diatchki says. "Then there are languages like Scala, which sort of depends on who you talk to whether they're functional or imperative, because they support both styles." Often, programmers choose the style that suits them best, or even employ both in different parts of their code.

To Diatchki's thinking, imperative languages are more like recipes, laying out a series of steps to accomplish a task, while functional programming is more like a calculator, except that it manipulates more than just integers. "You type in a big expression and say, 'what is the result now?'" He would not consider a language functional, he says, unless it treated functions as 'first-class citizens'; in other words, the functions can be used as arguments for other functions, and be returned as values by still other functions.

While much of the work in functional programming originated with academia—Haskell was created in 1990 by academics who were doing much the same work, but needed a common language to share their findings—it is increasingly finding a place in industry. For instance, Google's MapReduce, used for searching the Web, draws heavily on ideas from functional programming, Wadler says. *Map* applies a function to a large collection of data—finding instances of a term on scattered Web pages, say—and *reduce* then accumulates what it returns, presenting the search results as a new page.

Facebook uses Haskell to filter spam from postings. Microsoft supports F#, a dialect of Ocaml, in its Visual Studio development environment. A functional language developed by industry, Erlang, was created by the Stockholm, Sweden-based telecommunications company Ericsson for telephone systems and now is used to build scalable real-time systems for banking, e-commerce, and instant messaging; the messaging and Voice-over-IP program Whatsapp is written in Erlang.

Laboratory Space

Peyton Jones sees functional programming as a laboratory for trying out new ideas in programming that then find their way into other languages. "Functional programming has been an effective seedbed of ideas," says Peyton Jones. "It started off extremely impractical, but nevertheless intellectually appealing, and that very practicality forced a series of ideas to come to the surface that have then turned out to be useful much more broadly." For instance, garbage collection, an automated approach to managing memory, started out in functional programming but has since been more widely adopted. It has also been a testbed for static type checking, which looks through the program for errors before runtime and can vastly reduce the number of bugs in a program.

In fact, Peyton Jones says, there is a lot of work among functional programming researchers on developing richer type systems, which verify the accuracy of a program. The theorem prover Coq is a dependently typed functional language able, because of its type structure, to verify mathematical statements, though it is difficult to write programs in, he says.

He sees functional programming moving toward formal verification, giving developers greater confidence a program will do what it is designed to do. "The more people care about reliability, about knowing that something is true, that money depends on it or people's lives depend on it, the more they're going to care about this," Peyton Jones says.

Of course, Diatchki says, it is not only functional programmers who care that what they write works correctly. "It's just that functional programming seems to be blurring the line between writing the program and verifying it."

The style also lends itself to new variations. Antoy and Michael Hanus, a professor of computer science at the University of Kiel, Germany, are studying functional logic programming, which adds a concept called determinism. That allows them to attack problems where no precise algorithm is known, and some trial and error is required. "This is quite convenient when you have to compute with partial or incomplete information," Antoy says, or when getting precise and complete information is inconvenient.

Peyton Jones believes functional programming is becoming more popular as more developers learn about it. Eventually, he says, its distinction from imperative languages will all but evaporate, as the useful features of one approach seep into the other. "They are kinds of ends of the spectrum that are converging," he says. "When the limestone of imperative programming has worn away, the granite of functional programming will be revealed underneath."

Further Reading

Diatchki, I.S., Hallgren, T., Jones, M.P., Leslie, R., and Tolmach, A.
Writing systems software in a functional language: An experience report
PLOS 2007
http://yav.github.io/publications/plos07.pdf

Antoy, S. and Hanus, M.
Functional Logic Programming, *CACM 43, April 2010*, doi:10.1145/1721654.1721675

Bernardy, J-P., Boespflug, M., Newton, R., Peyton Jones, S., and Spiwack, A.
Linear Haskell: Practical linearity in a higher-order polymorphic language,
***Proceedings of the ACM on Programming Languages 2*, 2017**
https://www.microsoft.com/en-us/research/publication/retrofitting-linear-types/

Vazou, N., Choudhury, V., Scott, R.G., Newton, P.R., Wadler, P., and Jhala, R.
Refinement reflection: Complete verification with SMT
Principles of Programming Languages (POPL), Los Angeles, 8—13 January 2018.
https://dl.acm.org/citation.cfm?doid=3177123.3158141

Functional Programming and Haskell
https://www.youtube.com/watch?v=LnX3B9oaKzw

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

Society | DOI:10.1145/3193759

Samuel Greengard

Finding a Healthier Approach to Managing Medical Data

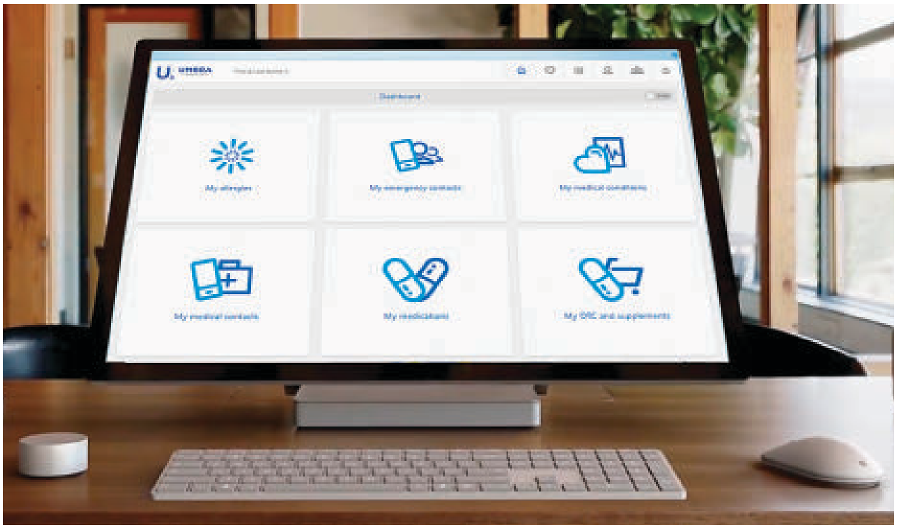
Researchers are exploring ways to put medical data to greater use while better protecting privacy.

One of the formidable challenges healthcare providers face is putting medical data to maximum use. Somewhere between the quest to unlock the mysteries of medicine and design better treatments, therapies, and procedures, lies the real world of applying data and protecting patient privacy.

"Today, there are many barriers to putting data to work in the most effective way possible," observes Drew Harris, director of health policy and population health at Thomas Jefferson University's College of Population Health in Philadelphia, PA. "The goals of protecting patients and finding answers are frequently at odds."

It is a critical issue and one that will define the future of medicine. Medical advances are increasingly dependent on the analysis of enormous datasets—as well as data that extends beyond any one agency or enterprise. What's more, as connected healthcare devices flourish, at-home and remote monitoring blossoms and big data analytics advances at a staggering rate, the stakes—and the ability to use, misuse, and abuse confidential data grows significantly.

"Healthcare is at a very important crossroads. To move to a more value-based framework and one that rewards patient and doctor behavior, we need to have systems in place that manage data and protect individuals," says Ophir Frieder, professor of computer science and information processing at Georgetown University in Washington, D.C., and professor of biostatistics, bioinformatics, and biomathematics at the Georgetown University Medical Center.



LIFEOGRAPHY DASHBOARD. IMAGE COURTESY OF UMBRA HEALTH.

Make no mistake, researchers are exploring ways to better manage and protect patient data. These methods revolve largely around machine learning, integrating blockchain into electronic healthcare records (EHRs) and other systems, and finding other ways to anonymize data, validate records, and prevent data leaks. "We must strike a balance between data ownership, interoperability, security, and dynamic consent for patients, so that data can be used and shared at the right times and under the right circumstances," says Jim Nasr, chief software architect for the Centers for Disease Control and Prevention (CDC).

Beyond Data

The level of disruption rippling through the healthcare industry is staggering. According to research firm IDC, the overall volume of data in the industry will increase from 153 exabytes in 2013 to 2,314 exabytes in 2020.

There also is a greater variety of data to manage. Electronic healthcare records, personal fitness devices, connected home monitoring systems, and a variety of other sensors, machines, and systems are pushing the boundaries of medicine in new directions. As a result, researchers, physicians, and other practitioners—using big data analytics and machine learning—can spot patterns, trends, and causalities that would otherwise escape human detection. This makes it possible to improve therapies, procedures, and drugs, while improving diagnostics and care for individual patients.

Yet the risks are also enormous—and they are magnified by the fact that there are no clear boundaries for what constitutes appropriate or inappropriate use. In some cases, it's possible to trace aliases, codes, and metadata used for anonymous tracking back to individuals. At the same time, a substantial amount of health data—particularly information from activity trackers, website searches, and credit card records—remains unregulated in the U.S. and many other countries. Because all this data falls outside the scope of the U.S. Health Insurance Portability and Accountability Act (HIPAA), data scientists can circumvent privacy protections by combining publicly available data with anonymized data to gain deep insights into personal behavior and health.

Marketers and companies looking to target consumers can tap into this data. "The result is a blizzard of transactions hidden to the public in which companies (called data miners) buy, sell, and barter anonymized but intimate profiles of hundreds of millions of Americans," says Adam Tanner, who authored a 2017 report for The Century Foundation, *Strengthening Protection of Patient Medical Data*. "While the anonymization of patient data may seem like a good firewall for protecting privacy, it increasingly is not." In fact, Tanner says, "Data scientists can now circumvent HIPAA's privacy protections by making very sophisticated guesses, marrying anonymized patient dossiers with named consumer profiles available elsewhere—with a surprising degree of accuracy."

Says Andreas Holzinger, professor of machine learning at the Medical University of Graz in Austria and founder and lead of the university's Human-Computer Interaction and Knowledge Discovery and Data Mining (HCI-KDD) group. "Healthcare data represents enormous value to both legitimate businesses and the hacker community." For example, employers could potentially use private medical and healthcare data to guide decisions about hiring and firing. Insurance companies could use personal data to make coverage and pricing decisions, and individuals could find the public release of personal health data embarrassing or costly in other ways. "If we are unable to protect people but at the same time enable the use of data in an appropriate manner, we risk the public losing confidence in the system and medical researchers losing opportunities to solve problems," Holzinger says.

Protection Schemes

The need for more sophisticated research methods and controls is redefining healthcare. Researchers are exploring new and more sophisticated ways to collect, manage, and exchange data. In addition, regulatory requirements in many countries are adding to the urgency. For instance, the General Data Protection Regulation (GDPR) in the European Union requires any organization handling data for even a single European citizen to abide by strict privacy guidelines, or risk a substantial fine. In Japan, a law introduced last year requires stricter controls over how healthcare providers manage data. While all electronic healthcare records must be searchable for academic researchers, drug companies, and others, facilities must make the data completely anonymous.

> **"There are almost no standards for data. As a result, doctors rarely have access to a complete and accurate medical record."**

Holzinger is developing a human-in-the-loop machine learning approach that offers a high level of data traceability, and the ability to explain how the system arrives at a conclusion. "We must understand how and why an algorithm makes decisions and that data is verifiable. Consequently, we have to move beyond a black box approach to have full confidence that data is accurate and systems work as advertised," he explains.

The central problem Holzinger is attempting to address is that medical data is intrinsically complex, high-dimensional, and noisy, and contains much unstructured information. A prime example is the use of Gaussian processes, where automated machine learning (aML) systems with standard kernel machines attempt to find answers through stochastic modeling. These systems struggle with basic extrapolation functions that remain very simple for humans.

An interactive machine learning (iML) model, on the other hand, allows a researcher, doctor, or other expert in the loop to select specific parameters or reduce an exponential search space through heuristic selection of samples. Such a model can also help guide causality, though it can also reflect biases and introduce or amplify human errors.

By combining cross-functional expertise, it also is possible to explore data models in entirely different ways—all while maintaining tight security and privacy controls through a tool such as blockchain. The goal is a glass-box approach to data processing. "It introduces the concept of explainable medicine. A medical doctor can retrace what a certain algorithm has done, and this may provide insight that ultimately delivers a medical explanation," Holzinger says.

At the CDC, Nasr and an accelerator development team are building a software framework that incorporates blockchain to integrate disparate systems used to address public health issues such as opioid abuse and infectious disease. Blockchain would guarantee the anonymous data is accurate, and that it comes from a legitimate source. This is critical because different groups and agencies—across a spectrum of public and private entities—must share data feeds and databases while ensuring errors, intentional or inadvertent, are not introduced into the data stream.

"We need to have greater flexibility than current data service architectures provide," Nasr explains.

At the center of this emerging model is a simple but profound issue, he says. "We must be able to effectively communicate software decisions and direction to a large customer base of physicians, epidemiologists, and public health experts. We have to ensure large numbers of disparate groups working on unconnected, separately funded, contract-based projects can access, share, and process data efficiently."

Nasr has his sights set on designing an interoperable software framework that can tie together databases, IoT devices, and more. The approach uses open source software, deployed through Docker containers, to create a mesh of functions and applications—Nasr calls this the "Software Theme Park"—that can be connected and rearranged to address broad data analytics requirements. These functions could carry out much of what is needed for public health data surveillance, including automatically validating anonymous data from different sources and across different application programming interfaces (APIs). The key, Nasr says, is standard-definition APIs regulated through an API gateway. This makes it possible to verify data across local, state, federal, and international agencies, as well as private organizations and other third-party data sources. The end result is a "robust information supply chain," he says.

For the Records

Finding ways to improve healthcare at the patient level is also at the center of this revolution in informatics. Georgetown's Frieder, who also serves as chief scientific officer for Umbra Health, has developed a framework for connecting and merging disparate medical data automatically while protecting the identity of patients. The system allows authorized healthcare practitioners—doctors, dentists, therapists, nutritionists, and others—to access EHRs and view only the specific information they require to do their jobs.

Frieder's motivation? Medical errors account for about 250,000 deaths a year in the U.S. alone, according to a 2016 study conducted by Johns Hopkins Medicine. In addition, errors account for many other injuries and therapy failures globally. "Part of the problem is inconsistent systems and processes. There are almost no standards for data. As a result, doctors rarely have access to a complete and accurate medical record."

The approach, which the company calls Lifeography, ties a person's data together in a secure cloud-based HIPAA-compliant environment. It harnesses blockchain to create a traceable ledger that spans healthcare touchpoints. Frieder says the system serves as a *lingua franca* for healthcare data, and delivers a longitudinal view from birth to death. The software framework delivers only the data the patient and attendant providers deem necessary for a specific interaction or transaction and strips out any unnecessary personal information, while also providing a forensic trail of users and devices. The end goal is to deliver "more precise orders, more accurate diagnoses, and more coordinated healthcare, including predictive and preventative capabilities," he says.

Of course, the ultimate challenge is to ensure these next-generation systems and processes make data widely accessible, while locking it down. The goal is to mine data to the extent possible, but protect personal privacy.

For now, many questions remain about who should hold and maintain blockchain ledgers, who should be granted privileges to modify or view data, and how identities should be managed and displayed on a blockchain. No clear consensus or direction has emerged.

Nevertheless, the future of healthcare is clear: "We need greater portability of data, greater interoperability between systems, and a more coordinated approach to patient care," Harris says. "As we incorporate fitness devices, medical monitoring devices, and more advanced analytics, systems must address the often-competing interests of putting data to maximum use but also protecting it." C

Further Reading

Holzinger, A.
Beyond Data Mining: Integrative Machine Learning for Health Informatics. April, 2016; https://online.tugraz.at/tug_online/voe_main2.getVollText?pDocumentNr=1347744&pCurrPk=89271.

Holzinger, A.
Machine Learning for Health Informatics. ML for Health Informatics, LNAI 9605, pp. 1–24, 2016; https://pure.tugraz.at/portal/files/7615225/HOLZINGER_2016_Machine_Learning_for_Health_Informatics.pdf.

Aitken, M., de St. Jorre, J., Pagliari, C., Jepson, R., and Cunningham-Burley, S.
Public responses to the sharing and linkage Of health data for research purposes: a systematic review and thematic synthesis of qualitative studies. *BMC Medical Ethics*, Vol. 17, No. 1. Jan. 12, 2016.

Prater, V.S.
Confidentiality, Privacy and Security of Health Information: Balancing Interests. Dec. 8, 2014. https://healthinformatics.uic.edu/resources/articles/confidentiality-privacy-and-security-of-health-information-balancing-interests/

Johns Hopkins Medicine, News Release, Study Suggests Medical Errors Now Third Leading Cause of Death in the U.S., May 3, 2016; http://bit.ly/2BeXY5T

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

Milestones

ACM Bestows Turing Award, Prize in Computing

At press time, ACM announced the names of this year's recipients of its most prominent awards.

The A.M. Turing Award, ACM's most prestigious technical award, is given for major contributions of lasting importance to computing.

This year's ACM A.M. Turing Award is being presented to John Hennessy, executive chairman of Alphabet Inc., Google's parent company, and David Patterson, Pardee Professor of Computer Science, Emeritus, at the University of California at Berkeley (and a past president of ACM) "for pioneering a systematic, quantitative approach to the design and evaluation of computer architectures with enduring impact on the microprocessor industry."

The ACM Prize in Computing (formerly the ACM-Infosys Foundation Award in the Computing Sciences) recognizes an early to mid-career fundamental innovative contribution in computing that, through its depth, impact, and broad implications, exemplifies the greatest achievements in the discipline.

This year's ACM Prize in Computing is being presented to Dina Katabi of the Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory for her creative contributions to wireless systems. Katabi applies methods from communication theory, signal processing, and machine learning to solve problems in wireless networking.

These awards will be presented at the ACM Awards Banquet in San Francisco in June.

***Communications* will provide in-depth coverage of these award recipients in upcoming issues, beginning with interviews with Hennessy and Patterson in the June issue.**

—Lawrence M. Fisher

DOI:10.1145/3199599

Ryan Calo

Law and Technology

Is the Law Ready for Driverless Cars?

Yes, with one big exception.

I AM A law professor who teaches torts and has been studying driverless cars for almost a decade. Despite the headlines, I am reasonably convinced U.S. common law is going to adapt to driverless cars just fine. The courts have seen hundreds of years of new technology, including robots. American judges have had to decide, for example, whether a salvage operation exercises exclusive possession over a shipwreck by visiting it with a robot submarine (it does) and whether a robot copy of a person can violate their rights of publicity (it can). Assigning liability in the event of a driverless car crash is not, in the run of things, all that tall an order.

There is, however, one truly baffling question courts will have to confront when it comes to driverless cars—and autonomous systems in general: What to do about genuinely unforeseeable categories of harm?

Imagine a time when driverless cars are wildly popular. They are safer than vehicles with human drivers and their occupants can watch movies or catch up on email while traveling in the vehicle. Notwithstanding some handwringing by pundits and the legal academy, courts have little trouble sorting out who is liable for the occasional driverless car crash. When someone creates a product that is supposed to move people around safely and instead crashes, judges assign liability to whoever built the vehicle or vehicles involved in the accident.

There are some difficult cases on the horizon. Policymakers will have to determine just how much safer driverless cars will need to be compared to human-operated cars before they are allowed—or even mandated—on public roads.

There are some difficult cases on the horizon.

Courts will have to determine who is responsible in situations where a human or a vehicle could have intervened but did not. On the one hand, courts tend to avoid questions of machine liability if they can find a human operator to blame. A court recently placed the blame of an airplane accident exclusively on the airline for incorrectly balancing the cargo hold despite evidence the autopilot was engaged at the time of the accident.[7] On the other hand, there is presumably a limit on how much responsibility a company can transfer to vehicle owners merely because they clicked on a terms of service agreement. In the fatal Tesla crash last year, the deceased driver seemingly assumed the risk of engaging the autopilot. The pedestrian killed this year by an Uber driverless car took on no such obligation. In its centuries of grappling with new technologies, however, the common law has seen tougher problems than these and managed to fashion roughly sensible remedies. Uber will likely settle

PHOTO BY PHASE4STUDIOS

its case with the pedestrian's family. If not, a court will sort it out.

Some point to the New Trolley Problem, which posits that cars will have to make fine-grained moral decisions about whom to kill in the event of an accident. I have never found this hypothetically particularly troubling. The thought experiment invites us to imagine a robot so poor at driving that, unlike you or anyone you know, the car finds itself in a situation that it *must kill someone*. At the same time, the robot is so sophisticated that it can somehow instantaneously weigh the relatively moral considerations of killing a child versus three elderly people in real time. The New Trolley Problem strikes me as a quirky puzzle in search of a dinner party.

Technology challenges law not when it shifts responsibility in space and time, as driverless cars may, but when the technology presents a genuinely novel affordance that existing legal categories failed to anticipate.

Imagine one manufacturer stands out in this driverless future. Not only does its vehicle free occupants from the need to drive while maintaining a sterling safety record, it adaptively reduces its environmental impact. The designers of this hybrid vehicle provide it with an objective function of greater fuel efficiency and the leeway to experiment with system operations, consistent with the rules of the road and passenger expectations. A month or so after deployment, one vehicle determines it performs more efficiently overall if it begins the day with a fully charged battery. Accordingly, the car decides to run the gas engine overnight in the garage—killing everyone in the household.

Imagine the designers wind up in court and deny they had any idea this would happen. They understood a driverless car could get into an accident. They understood it might run out of gas and strand the passenger. But they did not in their wildest nightmares imagine it would kill people through carbon monoxide poisoning.

This may appear, at first blush, to be just as easy a case as the driverless car collision. It likely is not. Even under a strict liability regime—which dispenses with the need to find intent or negligence on the part of the defendant—courts still require the plaintiff to show the defendant could foresee at least the category of harm that transpired. The legal term is "proximate causation." Thus, a company that demolishes a building with explosives will be liable for the collapse of a nearby parking garage due to underground vibrations, even if the company employed best practices in demolition. But, as a Washington court held in 1954, a demolition company will not be liable if mink at a nearby mink farm react to the vibrations by instinctively eating their young.[2] The first type of harm is foreseeable and therefore a fair basis for liability; the second is not.

We are already seeing examples of emergent behavior in the wild, much less in the university and corporate research labs that work on adaptive systems. A Twitter bot once unexpectedly threatened a fashion show in Amsterdam with violence, leading the organizers to call the police.[3] Tay—Microsoft's ill-fated chatbot—famously began to deny the

Holocaust within hours of operation.[5] And who can forget the flash crash of 2010, in which high-speed trading algorithms destabilized the market, precipitating a 10% drop in the Dow Jones within minutes.[4]

As increasing numbers of adaptive systems enter the physical world, courts will have to reexamine the role foreseeability will play as a fundamental arbiter of proximate causation and fairness.[1] That is a big change, but the alternative is to entertain the prospect of victims without perpetrators. It is one thing to laugh uneasily at two Facebook chatbots that unexpectedly invent a new language.[a] It is another to mourn the loss of a family to carbon monoxide poisoning while refusing to hold anyone accountable in civil court.

We lawyers and judges have our work cut out for us. We may wind up having to jettison a longstanding and ubiquitous means of limiting liability. But what role might there be for system designers? I certainly would not recommend stamping out adaptation or emergence as a research goal or system feature. Indeed, machines are increasingly useful *precisely because* they solve problems, spot patterns, or achieve goals in novel ways no human imagined.

Nevertheless, I would offer a few thoughts for your consideration. First, it seems to me worthwhile to invest in tools that attempt to anticipate robot behavior and mitigate harm.[b] The University of Michigan has constructed a faux city to test driverless cars. Short of this, virtual environments can be used to study robot interactions with complex inputs. I am, of course, mindful of the literature suggesting that the behavior of software cannot be fully anticipated as a matter of mathematics. But the more we can do to understand autonomous systems before deploying them in the wild, the better.

Second, it is critical that researchers be permitted and even encouraged to test deployed systems—without fear of reprisal. Corporations and regulators can and should support research that throws curveballs to autonomous technology to see how it reacts. Perhaps the closest analogy is bug bounties in the security context; at a minimum, terms of service agreements should clarify that safety-critical research is welcome and will not be met with litigation.

Finally, the present wave of intelligence was preceded by an equally consequential wave of connectivity. The ongoing connection firms now maintain to intelligence products, while in ways problematic, also offers an opportunity for better monitoring.[6] One day, perhaps, mechanical angels will sense an unexpected opportunity but check with a human before rushing in.

None of these interventions represent a panacea. The good news is that we have time. The first generation of mainstream robotics, including fully autonomous vehicles, does not present a genuinely difficult puzzle for law in this law professor's view. The next well may. In the interim, I hope the law and technology community will be hard at work grappling with the legal uncertainty that technical uncertainty understandably begets. **C**

[a] Tim Collins and Mark Prigg, "Facebook shuts down controversial chatbot experiment after AIs develop their own language to talk to each other," *Daily Mail* (Jul. 31, 2017); http://dailym.ai/2vnk47J Also see "Did Facebook Shut Down an AI Experiment Because Chatbots Developed Their Own Language?" Snopes.com (Aug. 1, 2017); http://dailym.ai/2vnk47J (concluding that Facebook did not necessarily expect the behavior but nor did it shut down the experiment as a result of it).

[b] For a prescient discussion, see Jeffrey Mogel, "Emergent (Mis) Behavior vs. Complex Software Systems." *ACM SIGOPS 40*, 4 (Oct. 2006), 293–304.

References

1. Calo, R. Robotics and the lessons of cyberlaw. *California Law Review 513*, 103 (2015).
2. *Foster v. Preston Mill Co.* 268 P.2d 645 (Wash. 1954).
3. Hill, K. Who do we blame when a robot threatens to kill people? Spinter.com (Feb. 15, 2015); http://bit.ly/2FFKszl
4. Hope, B. and Ackerman, A. 'Flash crash' overhaul is snarled in red tape. *Wall Street Journal* (May 5, 2015); http://on.wsj.com/2ph9w4D
5. Price, R. Microsoft is deleting its AI chatbot's incredibly racist tweets. *Business Insider* (Mar. 24, 2016); http://read.bi/1ZwcFYZ
6. Walker Smith, B. Proximity-driven liability. *Georgetown Law Journal 1777*, 102 (2014).
7. Vladeck, D.C. Machines without principles. *Washington Law Review 117*, 89 (2014).

Ryan Calo (rcalo@uw.edu) is a Lane Powell and D. Wayne Gittinger Endowed Professorship Associate Professor of Law at the University of Washington in Seattle, WA, USA.

DOI:10.1145/3199601

Fred B. Schneider

Privacy and Security
Putting Trust in Security Engineering

Proposing a stronger foundation for an engineering discipline to support the design of secure systems.

WHEN WE MUST depend on a system, not only should we want it to resist attacks but we should have reason to believe that it will resist attacks. So security is a blend of two ingredients: mechanism and assurance. In developing a secure system, it is tempting to focus first on mechanism—the familiar "build then test" paradigm from software development. This column discusses some benefits of resisting that temptation. Instead, I advocate that designers focus first on aspects of assurance, because they can then explore—in a principled way—connections between a system design and its resistance to attack. Formalizing such connections as mathematical laws could enable an engineering discipline for secure systems.

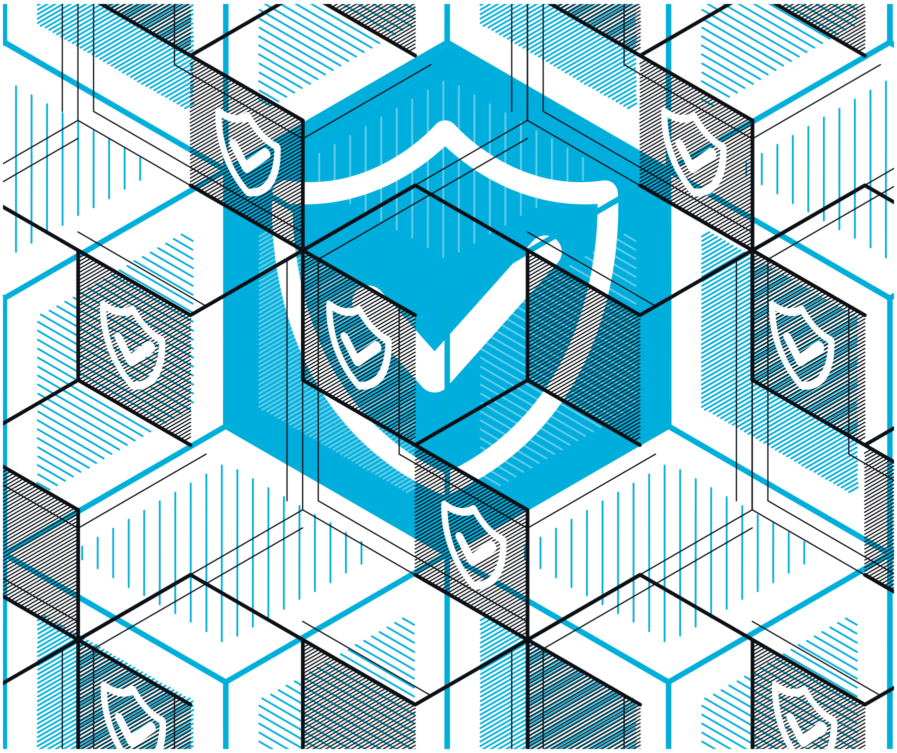


IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

Trust and Assumptions

To computer security practitioners, the term "trust" has a specific technical meaning, different from its use in everyday language. To *trust* a component *C* is to assert a belief that *C* will behave as expected, despite attacks or failures. When I say that "we should trust *C*" then either I am asking you to ignore the possibility that *C* is compromised or I am asserting the availability of evidence that convinced me certain (often left implicit) aspects of *C*'s behavior cannot be subverted. Availability of evidence is required in the second case, because what convinces me might not convince you, so psychological questions that underpin trust claims now can be explored separately in discussions about the evidence.

Trust is often contingent on assumptions. These assumptions must be sound, or our trust will be misplaced. We often make explicit our assumptions about the environment, talking about anticipated failures or the capabilities of attackers. But we also make implicit assumptions. For example, when expectations about behavior are couched in terms of operations and interfaces, we are making an implicit assumption: that attackers have access to only certain avenues for controlling the system or for learning information about its state. We also are making an implicit assumption when we ignore delays associated with memory access, since attackers might be able to make inferences by measuring those delays.

Assumptions are potential vulnerabilities. If a component will behave as expected only if some assumption holds, then an attacker can succeed simply by falsifying that assumption.

Most attacks can be deconstructed using this lens. For example, buffer-overflow attacks exploit an assumption about the lengths of values that will be stored in a buffer. An attack stores a value that is too long into the buffer, which overwrites values in adjacent memory locations, too. The recent Spectre[1] attack illustrates just how subtle things can get. With access to an interface for measuring execution times, an attacker can determine what memory locations are stored in a cache. Speculative execution causes a processor to access memory locations, transferring and leaving information in the cache. So, an attacker can learn the value of a secret by causing speculative execution of an instruction that accesses different memory depending on that secret's value. The implicit assumption: programs could not learn anything about speculative executions that are attempted but reversed.

One aspect of a security engineering discipline, then, would be to identify assumptions on which our trust depends and to assess whether these assumptions can be falsified by attackers. Whether such assumptions can be falsified will depend, in part, on an attacker's capabilities. The Defense Science Board[3] groups attackers into three broad classes, according to attacker capabilities:

- Those who only can execute existing attacks against known vulnerabilities;
- Those who can analyze a system to find new vulnerabilities and then develop exploits; and
- Those who can create new vulnerabilities (e.g., by compromising the supply chain).

Or we might characterize attackers in terms of what kind of access they have to a system:

- Physical access to the hardware;
- Access to the software or data; or
- Access to the people who use or run the system.

Cryptographers characterize attackers by bounding available computation; they see execution of PPT (probabilistic polynomial-time) algorithms as defining the limit of feasible attacks.

To build a system we are prepared to trust, we eliminate assumptions that constitute vulnerabilities we believe could be exploited by attackers.

- Analysis of a system or its components could allow weaker assumptions to replace stronger assumptions, because we then know more about possible and/or impossible behaviors. To embrace the results of such analysis, however, we must be prepared to trust the analyzer.
- Incorporating security mechanisms in a system or its components provides a means by which an assumption about possible and/or impossible behaviors can be made, because the security mechanism prevents certain behaviors. So we can weaken assumptions about system behaviors if we are prepared to trust a security mechanism.

In both cases, we replace trust in some assumption by asserting our trust in something else. So assumptions and trust have become the driving force in the design of a system.

An example will illustrate this role for assumptions and trust. To justify an assumption that service *S* executes in a benign environment, we might execute *S* in its own process. Process isolation ensures the required benign environment, but we now must trust an operating system *OS* to enforce isolation. To help discharge that assumption, we might run only the one process for *S* in *OS* but also execute *OS* in its own (isolated) virtual machine. A hypervisor *VMM* that implements virtual machines would then allow us to assume *OS* is isolated, thereby requiring a reduced level of trust in *OS*, because *OS* now executes in a more benign environment. Since a hypervisor can be smaller than an operating system, *VMM* should be easier to understand than *OS* and, therefore, easier to trust. Any isolation, however, is relative to a set of interfaces. The designers of *VMM* and *OS* both will have made assumptions about what interfaces to include. And, for example, if the interface to a memory cache was not included in the set of isolated interfaces then attacks like Spectre become feasible.

To build a system we are prepared to trust, we eliminate assumptions that constitute vulnerabilities that could be exploited by attackers.

Bases for Trust

The approach advocated in this column depends critically on having methods to justify trust in components and in systems built from those components. There seem to be three classes of methods: axiomatic, analytic, and synthesized. They are often used in combination.

Axiomatic Basis for Trust. This form of trust comes from beliefs that we accept on faith. We might trust some hardware or software, for example, because it is built or sold by a given company. We are putting our faith in the company's reputation. Notice, this basis for our trust has nothing to do with the artifact we are trusting. The tenuous connection to the actual component makes axiomatic bases a weak form of evidence for justifying trust. Moreover, an axiomatic basis for trust can be difficult for one person to convey to another, since the underlying evidence is, by definition, subjective.

Analytic Basis for Trust. Here we use testing and/or reasoning to justify conclusions about what a component or system will and/or will not do. Trust in an artifact is justified by trust in some method of analysis. The suitability of an analysis method likely will depend on what is being analyzed and on the property to be established.

The feasibility of creating an analytic basis for trust depends on the amount of work involved in performing the analysis and on the soundness of any assumptions underlying that analysis.

- *Testing.* In theory, we might check every input to every interface and conclude that some properties about behaviors are always satisfied. But enumeration and checking of all possible inputs is likely to be infeasible, even for simple components. So only a subset of the inputs to certain interfaces would be checked. An assumption is thus being introduced—that the right set of inputs is being checked.

▸ *Formal Verification.* Software is amenable to logical analysis, either manual or automated. Today's state of the art for automated analysis allows certain simple properties to be checked automatically for large components and allows rich classes of properties to be verified by hand for small components. Research in formal verification has made steady progress on widening the class of properties that can be checked automatically, as well as on increasing the size and complexity that can be handled.

An analytic basis for trust can be conveyed to some consumer by sharing the method and the results the method produced. When testing is employed, the artifact, set of test cases, and expected outputs are shared. For undertaking other forms of automated analysis, we would employ an analyzer that not only outputs a conclusion ("program type checked") but also generates and provides a transcript of the inferences that led to this conclusion—in effect, the analyzer produces a proof of the conclusion for the given artifact. Proof checking is, by definition, a linear-time process in the size of the proof, and proof checkers are far simpler programs than proof generators (that is, analyzers). So, without duplicating work, a consumer can check the soundness of a manually or automatically produced proof.

Synthesized Basis for Trust. Trust in the whole here derives from the way its components are combined—a form of divide and conquer, perhaps involving trust in certain of the components or in the glue used to combine them. Most of the mechanisms studied in a computer security class are intended for supporting a synthesized basis of trust. OS kernels and hypervisors enforce isolation, reference monitors and firewalls restrict the set of requests a component will receive, ignorance of a secret can impose unreasonable costs on an outsider attempting to perform certain actions.

With synthesized bases for trust, we place trust in some security mechanisms. These mechanisms ensure some component executes in a more-benign setting, so the component can be designed to operate in an environment characterized by stronger assumptions than we are prepared to make about the environment in which the synthesis (mechanism plus component) is deployed.

An engineering discipline should provide means to analyze and construct artifacts that will satisfy properties of interest.

Assumptions about independence are sometimes involved in establishing a synthesized basis for trust. With *defense in depth*, we aspire for a combination of defenses to be more secure than any of its elements. Two-factor authentication for withdrawing cash at an ATM is an example; a bankcard (something you have) and a PIN (something you know) both must be presented, so stealing a wallet containing the card alone does not benefit the attacker.[a] Defense in depth improves security to the extent that its elements do not share vulnerabilities. So an *independence* assumption is involved—we make an assumption that success in attacking one element does not increase the chances of success in attacking another.

Independence does not hold in replicated systems if each replica runs on the same hardware and executes the same software; the replicas all will have the same vulnerabilities and thus be subject to the same attacks. However, we can create some measure of independence across replicas by using address space layout randomization, which causes different replicas of the software to employ different memory layouts, so an attack that succeeds at one replica is not guaranteed to succeed at another. Other randomly selected per-replica semantics-preserving transformation would work, too.

Program rewriters are another means for creating synthesized bases. The rewriter takes a software component C as its input, adds checks or performs analysis, and outputs a version C' that is robust against some class of attacks, because C' is incapable of certain behaviors. If we trust the rewriter, then we have a basis for enhanced trust in C'. But even if we do not trust the rewriter, we could still have a basis for enhanced trust in that rewriter's output by employing a variant of *proof-carrying code*.[2] With proof-carrying code, the rewriter also outputs a proof that C' is a correctly modified version of C; certifiers for such proofs can be simple and small programs, independent of how large and complicated the rewriter is.

Conclusion

An engineering discipline should provide means to analyze and construct artifacts that will satisfy properties of interest. This column proposed a foundation for an engineering discipline to support the design of secure systems. It suggests that system design be driven by desires to change the assumptions that underlie trust. Security mechanisms change where we must place our trust; analysis allows us to weaken assumptions. ■

[a] Kidnapping the person usually gets the wallet, too. So the two mechanisms here have a vulnerability in common.

References

1. Kocher, et al. Spectre attacks: Exploiting speculative execution; https://spectreattack.com/spectre.pdf
2. Necula, G.C. Proof-carrying code. In *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming* (Paris, France), 1997, 106–119.
3. *Resilient Military Systems and the Advanced Cyber Threat.* Defense Science Board Task Force Report (Oct. 2013).

Fred B. Schneider (fbs@cs.cornell.edu) is Samuel B. Eckert Professor of Computer Science and chair of the at Cornell University computer science department, Cornell University, USA.

Supported in part by AFOSR grant F9550-16-0250 and NSF grant 1642120. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of these organizations or the U.S. government.

Schneider is grateful to Susanne Hambrusch, John King, Carl Landwehr, Steve Lipner, and Greg Morrisett for their very helpful feedback on early drafts of this column.

DOI:10.1145/3199603

Alexander Repenning

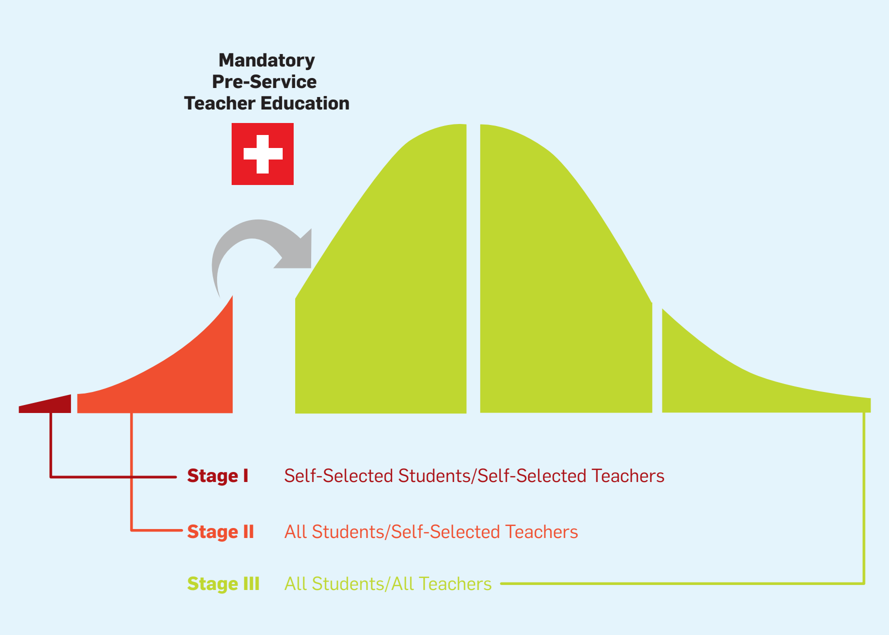
Education
Scale or Fail

Moving beyond self-selected computer science education in Switzerland.

K–12 COMPUTER SCIENCE Education (CSed) is an international challenge with different countries engaging in diverse strategies to reach systemic impact by broadening participation among students, teachers and the general population. For instance, the CS4All[9] initiative in the U.S. and the Computing at School[4] movement in the U.K. have scaled up CSed remarkably. While large successes with these kinds of initiatives have resulted in significant impact, it remains unclear how early impact[3] becomes truly systemic. The main challenge preventing K–12 CSed to advance from teachers who are technology enthusiasts to pragmatists is perhaps best characterized by Crossing the Chasm,[5] a notion anchored in the diffusion of innovation literature. This chasm appears to exist for CSed. It suggests it is difficult to move beyond early adopters (Figure 1, red and orange stages) of a new idea, such as K–12 CSed, to the early majority (Figure 1, green stage). Switzerland, a highly affluent, but in terms of K–12 CSed somewhat conservative country, is radically shifting its strategy to cross this chasm by introducing mandatory pre-service teacher computer science education starting at the elementary school level.

Three fundamental CSed stages, captured in Figure 1, are characterized by permutations of *self-selected/all* and *students/teachers* combinations. It took approximately 20 years to transition through these stages. Each stage is described here from a more general CSed perspective as well as my personal perspective.

Figure 1. Switzerland is crossing the computer science education chasm with mandatory pre-service teacher education.


Mandatory
Pre-Service
Teacher Education
Stage I
Self-Selected Students/Self-Selected Teachers
Stage II
All Students/Self-Selected Teachers
Stage III
All Students/All Teachers

Stage I: The "Friday Afternoon Computer Club" Stage (Self-Selected Students/Self-Selected Teachers): Stage I focused mostly on the "right" tools. In the 1990s the overall negative perception of programming by children, best described as "hard and boring,"[7] suggested *cognitive* as well as *affective* challenges. Being constantly just one semicolon away from total disaster with traditional programming languages such as C or Pascal, is an example of syntax being a cognitive challenge. But even with the presence of early educational programming languages, such as Logo and BASIC, programming in schools was typically marginalized into after-school contexts such as Friday afternoon clubs. These clubs, in turn, attracted only the usual suspects, that is, self-selected kids (mostly male) instructed by self-selected teachers (also mostly male). To make computing more relevant to schools it was crucial to focus less on tools supporting programming per se but to create tools to *forge explicative ideas through computing*, or as Papert started to call it, computational thinking.[6]

I developed *computational thinking tools* to allow students to get past the cognitive and affective challenges[7] at the University of Colorado. Computational thinking tools have the ultimate goal to transform people's perception of programming from "hard and boring" to "accessible and exciting." K–12 CSed does not have the intent of producing a fleet of programmers but on producing computational thinkers. Students should be enabled by computational thinking tools to participate in activities including programming, such as creating STEM simulations or games, in the regular K–12 course context and not just at the Friday afternoon computer clubs. With AgentSheets we made programming more accessible by moving beyond the challenges of syntax through drag-and-drop programming.[7] With AgentCubes we made programming more exciting[a] to kids by providing them with browser-based tools to create 3D shapes, which they can 3D print, compose into interactive 3D worlds, and share with their friends.

Stage II: The "Professional Development Movement" Stage (All Students/Self-Selected Teachers). Once computational thinking tools sufficiently addressed the cognitive and affective challenges dimensions of CSed, it was time to shift the research focus from tools to curricula and teacher professional development. Researchers started not only to design, and conduct teacher professional development but also to systematically evaluate its efficacy. The teachers attending were mostly self-selected, but in many cases the classes they would teach were aimed at all students and no longer limited to after-school contexts. Stage II started to reach a much broader audience than Stage I.

I initiated Scalable Game Design as a strategy to teach computational thinking through game design and STEM simulation creation activities.[8] Scalable Game Design is rooted in a didactic model called the Zones of Proximal Flow, which helps students to design their own games and STEM simulations based on the understanding of common computational thinking patterns.[1] Teacher professional development gradually scaled from local face-to-face summer institutes at the University of Colorado, to online and blended professional development, reaching schools in all 50 U.S. states.

K–12 computer science education does not have the intent of producing a fleet of programmers but on producing computational thinkers.

Private funding supported international Scalable Game Design sites, for example, Scalable Game Design Mexico funded by Google.

Stage III: The "Mandatory Pre-Service Teacher Education" Stage (All Students/All Teachers). Compelling curricula can spread through networks of excited CS teachers surprisingly quickly,[2] but how will they cross the CSed chasm (Figure 1) and persuade more conservative audiences? If society truly believes computational thinking is an essential 21st-century skill, just like mathematical thinking, then it will have to shift to transformative mandatory practices reaching all pre-service teachers.

In 2013, I began leading a pilot project in Switzerland to explore systemic impact on K–12 CSed through mandatory pre-service teacher education. This project engages all pre-service elementary school teachers at one of Switzerland's largest Schools of Education, PH FHNW, in CSed, through two mandatory CS courses. Other Swiss Schools of Education, e.g., the PHSZ, are offering similar mandatory CS courses. The new Swiss Lehrplan 21, a national curriculum reaching the 21 German-speaking states out of all 26 states in Switzerland, enabled this kind of transformational practice because it mandates K–12 CSed. The transformative nature hinges on the possibility for students to fail becoming an elementary teacher based on failing a mandatory CS course. This unparalleled kind of causality resulted in government intervention. Four years of negotiations, where the societal benefits of CS had to be explained to the Swiss Conference of Cantonal Ministers of Education, finally resulted in the accreditation of mandatory CS courses for elementary school teachers.

Scalable Game Design Switzerland Courses

Conceptually, the Scalable Game Design Switzerland course is rooted in the Scalable Game Design framework developed at the University of Colorado[8] but it had to go through radical transformation to deal with the different needs of a non-self-selected audience. Less than 1% of these pre-service teachers had programming experience. The expectations of teachers (75% women, 25% men) varied widely with many anticipating learning how to use Word or PowerPoint. Instead, just like the original course, this new course focused on computational thinking conveyed through game design and STEM simulation building. To make the course relevant to pre-service teachers, Scalable Game Design Switzerland had to be carefully aligned with the learning goals defined as competences in the Swiss Lehrplan 21.

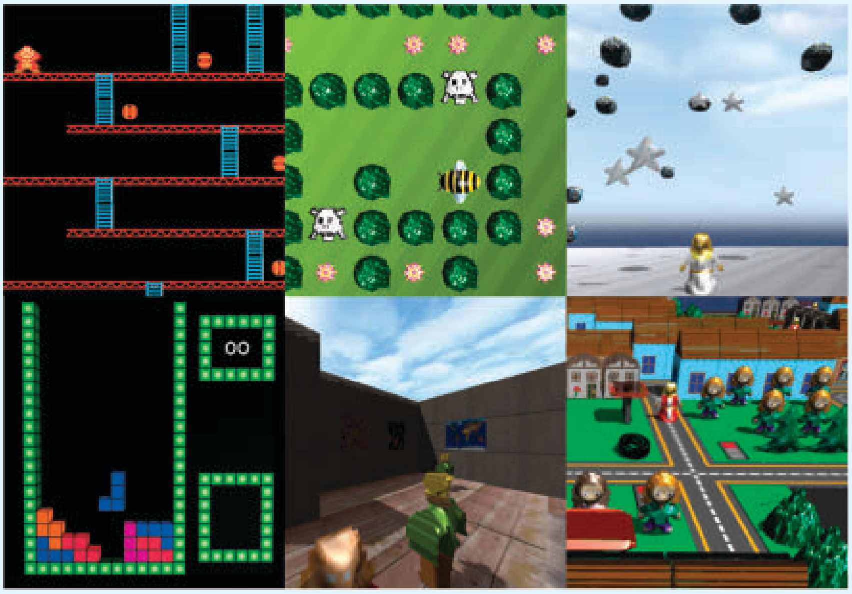
On September 17, 2017, in four states, seven instructors began to teach 26 CS courses with approximately 25 pre-service teachers each. All 600+ pre-service teachers must take two courses: Introduction to Computer Science and Computer Science Didactics. This totals four credit hours, resulting in over 25,000 contact hours. The Introduction to Computer Science course, which finished in December 2017, was based on three core concepts. Each concept is described here and includes a brief description of our preliminary assessment:

1. ***Learning and Motivation Strategy: Scalable Game Design.*** The strategy is to teach computational thinking through a series of computational thinking patterns[8] that gradually—hence the notion of scalability—expose students to increasingly sophisticated game design and STEM simulation building challenges, following the Zones of Proximal Flow strategy.[1]

The "scalable" aspect of the course worked well resulting in all pre-service teachers (students of the school of education) developing the necessary skills to program at least a basic game. Before the course, the acceptance of game design as learning strategy by teachers

[a] AgentCubes online is a 3D computational thinking tool with more than one million student projects created in 2017 in over 180 countries.

Figure 2. AgentCubes games designed and programmed by pre-service teachers.



was one of our biggest concerns. Course satisfaction overall was high with many even mentioning in their course evaluation that they liked programming games. Figure 2 shows sample games and simulations produced by teachers such as 1980s-era arcade-style 2D games (Donkey Kong), puzzle games (Tetris), educational simulation games (honey bee pollination), 3D indoor games, and 3D outdoor games. To design these games students had to engage in an iterative computational thinking process.[7]

2. ***Tools to support computational thinkers in schools: Computational thinking tools.*** We used our AgentCubes computational thinking tool because it had established accessibility even with younger kids, independent of gender and race,[8] and it enabled kids to create interesting artifacts such as 3D games and STEM simulations. To cover the wide range of needs in elementary schools, we also included CS Unplugged activities, for classrooms lacking computers, and Processing, as brief exposure to more traditional textual programming.

There was a huge shift of perception regarding pre-service teachers' ability to program. Before the course essentially nobody had done any kind of programming nor did they think programming was particularly relevant to K–12 education. At the end of the first course all 600+ pre-service teachers had created and programmed multiple games and participated in multi week game programming group projects. Everybody had gone beyond the minimal basics of programming, for instance by programming collaborate ghosts in a Pac-Man like game. Also, learning was not reduced to CS in isolation but included uses of computation in other subject areas such math, music, art, and social sciences.

3. ***Course structure: The seven big ideas of computer science.*** The mapping of the Scalable Game Design strategy onto the computer science part of the Lehrplan 21 was straightforward. The three main Swiss Lehrplan 21 CS topics (data, algorithms, and systems) were identified as a subset of the seven big ideas found in the AP Computer Science Principles framework: creativity, abstraction, data, algorithms, programming, the Internet and global impact. Each Computer Science Principle idea was mapped onto a two-week block in the 14-week course.

In spite of its density, most students really enjoyed this course structure. Students took particular pleasure in the Computer Science Principles ideas that were either evidently relevant to their lives such as the Internet, or conveyed through engaging activities such as a CS unplugged role play of people acting as bubbles in bubble sort to understand the notion of an algorithm.

Scale or Fail?

The preliminary evaluation of the first part of the course (Introduction to Computer Science) suggests a shift in audience from *self-selected in-service teachers* to *all pre-service teachers* may not be as difficult as anticipated. However, to actually assess if we can teach these pre-service teachers to teach computational thinking to their students will not only require us to conduct the second part of the course (Computer Science Didactics) but also much more time to observe practical impact. We have an approach that has been successful in the past, and we have a broad, transformative experiment that could scale to an entire country. We have confidence that this approach will allow us to cross the CSed chasm. C

References

1. Basawapatna, A., Repenning, A., Koh, K.H., and Nickerson, H. The zones of proximal flow: Guiding students through a space of computational thinking skills and challenges. In *Proceedings of the International Computing Education Research* (ICER 2013), San Diego, CA, USA, 2013.
2. Bradshaw. P. and Woollard, J. Computing at school: An emergent community of practice for a re-emergent subject. In *Proceedings of the International Conference on ICT in Education*, Rhodes, Greece, 2012.
3. Brown, N.C.C., Sentance, S., Crick, T., and Humphreys, S. Restart: The resurgence of computer science in U.K. schools. *Trans. Comput. Educ. 14* (2014), 1–22.
4. Crick, T. and Sentance, S. Computing at school: Stimulating computing education in the U.K. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research* (Koli, Finland), 2011, 122–123.
5. Moore, G.A. *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers.* HarperBusiness, New York, 1999.
6. Papert, S. An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning 1* (1996), 95–123.
7. Repenning, A. Moving beyond syntax: Lessons from 20 years of blocks programing in AgentSheets. *Journal of Visual Languages and Sentient Systems 3*, (2017), 68–89.
8. Repenning, A. et al. Scalable game design: A strategy to bring systemic computer science education to schools through game design and simulation creation. *Transactions on Computing Education (TOCE) 15* (2015), 1–31.
9. Vogel, S., Santo, R., and Ching, D. Visions of computer science education: Unpacking arguments for and projected impacts of CS4All initiatives. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, Seattle, WA, USA.

Alexander Repenning (alexander.repenning@fhnw.ch) is Professor and Chair of Computer Science Education at FHNW, School of Education, Windisch, Switzerland, and Professor of Computer Science at the University of Colorado at Boulder, Boulder, CO, USA.

This research was supported by the U.S. National Science Foundation, the Swiss National Science Foundation, and the Hasler Foundation.

DOI:10.1145/3201606

Harold "Bud" Lawson

Viewpoint
The March into the Black Hole of Complexity

Addressing the root causes of rapidly increasing software complexity.

In June 2002, *Communications* published my Viewpoint "Rebirth of the Computer Industry," in which I expressed hope that the past complexity sins of the computer industry had been admitted and that something positive could happen.[5] Instead, complexity has increased at an accelerated and alarming rate. I pointed to many fundamental problems in my previous Viewpoint; here, I emphasize two aspects: the hardware-software mismatch and the state of affairs in developing and sustaining software systems. Both are contributing root causes introducing enormous risks and impacting digital-age safety, security, and integrity. If the world ever hopes to climb out of the black hole of software complexity it will involve addressing these aspects in a constructive manner.

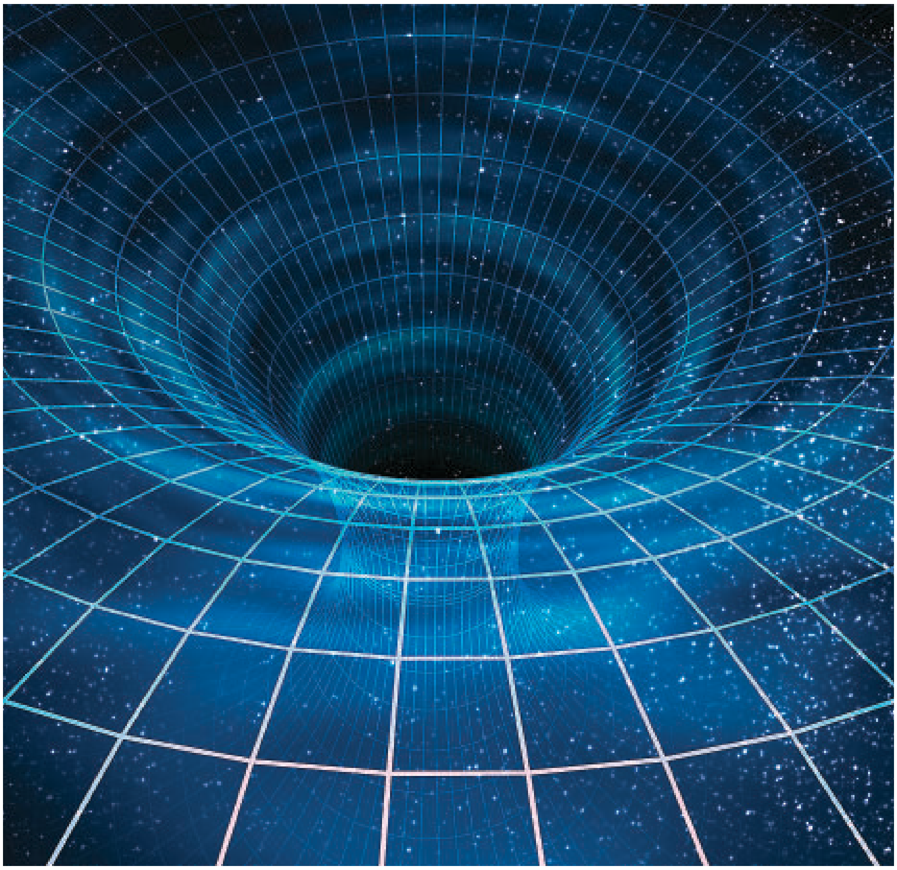


IMAGE FROM SHUTTERSTOCK.COM

Hardware-Software Mismatch and Consequences

During the late 1950s, Robert (Bob) Barton pioneered the idea of designing the hardware to accommodate the needs of software; in particular programming languages in the design of the Burroughs B5000. For example, he used the Łukasiewicz's concept of Reverse Polish Notation as the means of expression evaluation. There were several other software relevant innovations as well. Burroughs continued to develop the B5500 and 6500.[1] I fondly remember Burroughs provided a pedagogical game that one could clearly learn the underlying principles for Algol and Cobol program translation and execution.

However, Burroughs did not dominate the computer industry. IBM did. With the announcement of the IBM System/360 series in 1964, the IT World entered a new era. Clearly, there were significant advances in the 360, in particular the usage of solid-state logic and the control of most models by microprograms. While the series provided compatibility over a range of models at various price/performance levels, there was, in my opinion, a very serious problem. The Instruction Set Architecture (ISA) included 143 instructions, but it was not easy to generate efficient code from the various compilers for Fortran, Cobol, and the new language PL/I.

The problems with providing 360 systems software have been well documented by Fred Brooks in his book *The Mythical Man Month*.[2] I was a member of the Change Control Board for OS/360 as representative for PL/I. So, I observed firsthand the rapidly increasing complexity. The original concepts for OS/360 were clear and described in a small notebook. Finding that the instruction set was difficult to deal with, compiler and other system software projects requested exceptions, for example in calling conventions and parameter passing. As a result the documentation volume grew rapidly and eventually it became virtually impossible to follow all the changes. Entropy was a fact; very few people read anymore. To add to the complexity, a very complicated Job Control Language was provided.

Given the situation IBM discovered a new market. Customers had extreme difficulty in installing and operating their System/360s. So, IBM established the role of "Systems Engineers" with the altruistic goal of helping customers making their installations and applications operational. The bug-laden software resulted in fix after fix, each fix reconciling some problems but introducing new ones. This all produced an enormous revenue source from the selling of Systems Engineer's services.

I identify this as the beginning of the March into the Black Hole of Complexity. It created fantastic opportunities for consultants and start-up companies that made fortunes because they could handle some piece of the complexity. Now very significantly amplified by a wide variety of suppliers of complex software systems.

There were efforts in 1960s and early 1970s to return to the importance of the hardware-software relationship. I led a research group at IBM that developed ideas of T-Machines and E-Machines supported by microprograms. T-Machines that implemented an instruction set conducive to constructing compilers and based upon ideas from Digitek (at that time a supplier of compilers). Compiled programs executed by E-Machines that implemented programming language-relevant instruction sets in a manner similar to the Burroughs computers.

There were several others following similar ideas including Wayne Wilner with the microprogrammed Burroughs B1700[10] and the work of Glenford Myers on software-oriented architectures.[8] I was involved in two new microprogrammable architectures where there were plans to not only emulate existing machines, but to use the T- and E-Machine approach. First was the MLP-900 at Standard Computer Corporation.[7] Unfortunately, due to a change in management, only the prototype was produced, but it wound up on the original ARPA network in a dual configuration with a PDP-10 at USC Information Science Institute. It provided a microprogram research facility and was used during the 1970s. Datasaab in Sweden planned to license the MLP-900 to emulate a previous machine, but also to work toward implementing T- and E-Machines. There, I designed the Flexible Central Processing Unit, a 64-bit machine, with rather advanced microprogram features. It was microprogrammed to provide a compatible D23 system, but the true potential of using the FCPU for T- and E-Machine implementation did not transpire.[6]

Had these innovative "language-directed architectures" achieved wide market acceptance one wonders how computing would be these days? But as previously stated, IBM dominated. In the mid-1970s, the microprocessor showed up and radically changed hardware economics affecting the product offerings by all computer hardware suppliers. While the integrated circuit technology was a major achievement the establishment of a primitive ISA (Instruction Set Architecture) that has permeated in the X86 architecture has had an even more radical effect upon the hardware-software relationship than the 360. Generating code for these processors is highly complex and results in enormous volumes of code. As a result, true higher-level languages have often been put aside and the use of lower-level languages like C and C++ permeate. Another trend evolved in attempting to hide the complexity, namely via middleware where functions provided by higher-level abstractions are translated often to C code. Clearly middleware hides complexity, but it is also clear that it does not eliminate it. Finding bugs in this complex of software levels is a real challenge.

The March into the Black Hole of Complexity created fantastic opportunities for consultants and start-up companies.

The problems continued when IBM in an effort to capture the personal computer market agreed to use DOS and other Microsoft software. This was followed by the so-called WINTEL cycle. More powerful processors with more memory from Intel—and then new software functionality (often not really essential and not used) from Microsoft and then the next round.

During the 1980s there was a debate about the merits of the CISC (Complex Instruction Set of the X86 type) versus RISC (Reduced Instruction Set) architectures. While RISC architectures provided enhanced performance and the fact that higher-level functions can be achieved by subroutines, they do not directly address the hardware-software relationship. That is, there is a "semantic gap" between true higher-level languages and the ISA. The semantic gap refers to the level of cohesion between the higher-level language and the ISA. The T- and E-Machine approach described here has significantly reduced the semantic gap.

We now know that the world has been provided with an enormous amount of new functionality via both CISC and RISC processors; but with the nasty side effects of the unnecessary complexity in the form of enormous volumes of compiled code and middleware leading to bugs, viruses, hacker attacks, and so forth, to a large extent due to the enormous complexity. Given the current problems of cyber security, it is high time to seriously address the semantic gap and develop language-directed architectures that make software more understandable, maintainable, and protectable while also significantly reducing the amount of generated code.

Software Development, Deployment, and Sustainment

I have termed the unnecessary complexity that has evolved Busyware. Certainly it keeps vast numbers of consultants and teams of software engineers and programmers occupied. They should be focusing on producing good-quality software products (Valueware and Stableware), but find themselves often side-tracked into handling implementation complexities. While a resolution of the hardware-software mismatch would be a step in a positive direction, the scope and complexity of today's large software endeavors demand that the process of developing, deploying, and sustaining software must be improved. For example, today's operating systems and many advanced applications in the range of 50 to 100 million lines of code produced by a large group of software engineers and programmers have introduced significant problems of stability and maintainability.

The software engineering profession was established to improve capabilities in developing, deploying, and sustaining software. While early efforts focused on improvements in program structure, later developments have focused on the way of working. Many "gurus" have provided their own twist on best practices and methods that are often followed in a religious manner. As a result, a plethora of approaches have evolved. While this situation has made a lot of people rich, it certainly has contributed to additional complexities in selecting and applying appropriate practices and methods—moving us even deeper into the Black Hole of Complexity.

The scope and complexity of today's large software endeavors demand that the process of developing, deploying, and sustaining software must be improved.

In an effort to improve upon this alarming situation an international effort initiated by Richard Soley, Bertrand Meyer, and Ivar Jacobson resulted in the SEMAT (Software Engineering Method and Theory) organization. They observed that software engineering suffers from:

- The prevalence of fads more typical of a fashion industry than of an engineering discipline;
- The lack of a sound, widely accepted theoretical basis;
- The huge number of methods and method variants, with differences little understood and artificially magnified;
- The lack of credible experimental evaluation and validation; and
- The split between industry practice and academic research.

As a concrete step a team of international experts developed the Essence Kernel[4] that has become an OMG standard.[9] Essence provides:

- A thinking framework for teams to reason about the progress they are making and the health of their endeavors.
- A framework for teams to assemble and continuously improve their way of working.
- The common ground for improved communication, standardized measurement, and the sharing of best practices.
- A foundation for accessible, interoperable method and practice definitions.
- And most importantly, a way to help teams understand where they are, and what they should do next.

While developed for software engineering, when examining Essence it is obvious there are many ideas that can be applied on a wider scale. Certainly as we move into the era of the Internet of Things and cyber-physical systems the importance of organizing multidisciplinary teams and systems engineering become obvious. To meet this need, Ivar Jacobson and I have co-edited the book *Software Engineering in the Systems Context* where many well-known software and systems experts have provided their input.[3] Further, a call is made to extend the ideas from Essence to the systems engineering domain.

Conclusion

We are very deep in the Black Hole of Complexity and two important root causes to this situation have been identified. First, due to the mismatch between hardware and software that has resulted in complexities with the widescale usage of lower-level languages and middleware. Secondly, given the scope and complexity of today's software systems, we need to improve our approach to developing, deploying, and sustaining software systems that have become the most important and vulnerable elements of modern-day systems. Here, Essence provides an important step forward. Progress must be made in these two important aspects if the world is to avoid sinking further into the Black Hole of Complexity.

References

1. Barton, R. Functional design of computers. *Commun. ACM 4*, 9 (Sept. 1961).
2. Brooks, F. *The Mythical Man-Month*. Addison-Wesley, 1974.
3. Jacobson, I. and Lawson, H., Eds. *Software Engineering in the Systems Context, Volume 7*. Systems Series, College Publications, Kings College, U.K., 2015.
4. Jacobson, I. et al. *The Essence of Software Engineering: Applying the SEMAT Kernel*. Addison-Wesley, 2013.
5. Lawson, H. Rebirth of the computer industry. *Commun. ACM 45*, 6 (June 2002).
6. Lawson, H. and Magnhagen, B. Advantages of structured hardware. In *Proceedings of the 2nd Annual International Symposium on Computer Architecture*, 1975.
7. Lawson, H. and Smith B. Functional characteristics of a multi-lingual processor. *IEEE Transactions on Computers C-20*, 7 (July 1971).
8. Myers, G. SWARD—A software-oriented architecture. In *Proceedings of the International Workshop on High-Level Language Computer Architecture*, 1980.
9. OMG. *Essence—Kernel and Language for Software Engineering Methods*. Object Management Group, 2015.
10. Wilner, W. Design of the Burroughs B1700. In *Proceedings of the Fall Joint Computer Conference*, 1972.

Harold "Bud" Lawson (bud@lawson.se) is an ACM, IEEE, and INCOSE Fellow, IEEE Charles Babbage Computer Pioneer, and INCOSE Systems Engineering Pioneer.

DOI:10.1145/3199604

Margaret Martonosi

Viewpoint Science, Policy, and Service

Some thoughts on the way forward.

For many computer scientists, the thrill of impactful technical inventions and fast-moving innovation is what pulled us into this field originally and is what keeps us here. But as computer systems impact society ever more deeply, questions arise regarding the societal implications of the technologies we are building, and from there the conversation often shifts to questions of policy and regulation. When it comes to privacy, security, and other crucial issues, what should be expected of these computer systems that are increasingly ubiquitous? How should computer systems be managed and regulated? Just as importantly, who should make those decisions? How will policy experts grapple with developing regulatory and governance decisions about the deeply complex technologies we are developing, and who will help them understand what they need to know?

U.S. Department of State headquarters in Washington, D.C., USA.

PHOTO BY MARK VAN SCYOC/SHUTTERSTOCK.COM

From its roots supporting trajectory calculations and cryptography during the wars in the first half of the 20th century, computing technology has been deeply intertwined with policy and government issues. But the societal and personal impact of our inventions is becoming more and more apparent and urgent. Artificial intelligence and machine learning are now integral to a profound array of real-world applications, from autonomous vehicles to law enforcement. Internet of Things (IoT) devices and systems are being built and sold that affect human health and safety in myriad ways; they control the electrical grid, manage transportation, meter out medicine dosages, and control the front-door locks on houses. More than ever before, CS as a field faces deep responsibility for creating algorithms, devices, and systems that operate in a manner that is reliable, secure, ethical, and fair.

Simultaneously, the deceleration of Moore's Law and Dennard scaling mean that the underlying semiconductor improvements supporting our innovations are slowing and this scaling loss is leading to massive top-to-bottom shifts in how software and systems are designed, programmed, and operated. The low-level system designer's response has been to supplant software with increased use of specialized hardware accelerators for cryptographic routines, image analysis, and so forth. These hardware-accelerator-oriented approaches offer a viable path forward in the short-term, but are not without their challenges. Namely, although both hardware and software are susceptible to bugs and vulnerabilities, the bugs and vulnerabilities our systems "bake into" hardware are much more difficult, expensive, and time-consuming to fix.

So here we are, in a world where computing devices and systems have more societal impact than ever before, and yet they are simultaneously more difficult to test, debug, patch, or even *understand* than ever before. And in the midst of that, policymakers are—often with relatively little technical background—deciding what to do …

From August 2015 to April 2017, I was a Jefferson Science Fellow (JSF) engaged in computing and communications policy within the United States Department of State. For the first 12 months, I lived in Washington, D.C., and worked in-person at the Office of International Communications and Information Policy in the Economics Bureau at the State Department. After I returned to my university position in August 2016, I continued to work remotely on these issues for an additional eight months. The Jefferson Science Fellow program was established in 2003, acknowledging the strong and widespread impact of science and technology trends on foreign policy issues, and with the goal of augmenting the State Department's in-house science and technology expertise by bringing in tenured science and engineering professors for one-year fellowships. It is one of many efforts across the U.S. government to include scientists and technologists in the conversations regarding how "our" topics shape our world and therefore may require technical inputs for their policy and governance.

Sometimes when I mention to computer scientists that I worked at the State Department, they react in surprise, because it does not seem like a tech-heavy arm of our government. But as many news stories over recent years have highlighted, when foreign policy decisions involve topics like data privacy, security, encryption, Internet censorship, or other charged scientific topics like climate change, most of us would agree it is preferable to have those decisions guided by sound science and technological truth wherever possible.

Both domestically and abroad, computing technology is viewed in terms of both societal and economical benefits.

So what did I do as a JSF? I followed a range of computing and communications issues for my group, such as Internet of Things (IoT), smart cities, blockchain, and financial technologies. For these and other broader topics (computer security and privacy) I interpreted and explained the technology behind the topic, and I contributed to broader governmental and multistakeholder processes to formulate U.S. positions on the issues. I also participated in international meetings where these issues were discussed, and I spoke for the U.S. position on these issues, often negotiating with other countries whose beliefs in topics like Internet freedom differ strongly from our own.

Both domestically and abroad, computing technology is viewed in terms of both societal and economic benefits. Internet connectivity catalyzes better education and healthcare, and it also supports economic growth and entrepreneurship. Likewise, issues like harmonizing the communication spectrum are also important worldwide, as they fundamentally define our ability to use cellphones globally. But from the Internet writ large, to cellphones, to tiny IoT devices, different stakeholders can have widely disparate opinions about how they should be regulated and standardized. In my experience, I saw countries and companies pushing to develop broad technical standards requiring heavyweight cost accounting methods built into tiny IoT devices, while others preferred lighter weight methods that would avoid these. I also saw proposals of network protocols that encouraged inspection of packet contents—with considerable privacy and security implications—that concerned other meeting participants. These differences of opinion need to be negotiated; without some broader agreement, computer systems will lack the interoperability and ubiquity that makes them so vital in our world today.

On a nearly daily basis, a range of computing and telecommunications and digital economy issues are being discussed in multilateral forums including United Nations agencies, in the International Telecommunications Union (ITU), in the Organization for Economic Cooperation and Development (OECD), in the G7, and in other similar dialogs. Most researchers are surprised when I describe the *frequency* of intergovernmental policy meetings on computing topics. One of my goals as a JSF was to help ensure the U.S. policy approaches at these forums were technically sound and viable. My colleagues and I also advocated for approaches and meeting outcomes consistent with U.S. values regarding an open and interoperable Internet governed through multistakeholder processes, and a level playing field for global tech innovation. In addition to learning a great deal about technology and policy, the time I spent representing my country's values at international meetings also made—at a personal level—for a deeply fulfilling and patriotic year.

Although my JSF immersed me in *international* policy arenas, computing policy permeates many arenas across government, both domestic and broader. The technical and funding relationships between CS researchers and some agencies (Department of Defense, Department of Homeland Security (DHS), National Institute of Standards and Technology (NIST)) are familiar and long term. In addition, other agencies play increasing roles both in terms of funding and in terms of broader policy engagement. For example, the Department of Transportation has large projects related to smart cities and connected vehicles that will benefit greatly from CS input. Another example is increasing inter-agency attention on artificial intelligence policy issues over the past two years, including both technical aspects of trust and fairness, as well as broader issues of impact on the economy and labor markets. A further example is that as a country, we need to decide how to ensure a broad and diverse cross-section of our K–12 student population has access to computing education and affordable in-home broadband Internet access. Given the ubiquity of computing in society and in our economy, there is a corresponding ubiquity of computing-related policy questions in our government. A fundamental issue is: ***Who** will be answering these questions?*

Three years ago when I decided to join the State Department, I found it a very hopeful sign that I and many other science and technology researchers were working in Washington either as JSFs or "on detail" within the Office of Science and Technology Policy (OSTP), or as IEEE or AAAS fellows across government.

Now I am much less optimistic. The position of Science Adviser to the President has not been filled and much of OSTP is quite depleted overall. The State Department's reorganization and hiring freeze has left many leadership positions unfilled. This includes the State Department position of Science and Technology Adviser to the Secretary (STAS), which became vacant in July 2017; the Department's leadership has not committed to refilling it. The status of the JSF program in which I participated is also unclear. The program continues but with fewer than half as many fellows, and with placements only at USAID, not at State. More broadly, the news is filled with stories about the diminishment of science within offices at EPA, NOAA, and other agencies.

Talking with colleagues right now about government engagement, I am met with a wide range of responses. Immediately after the U.S. 2016 election and 2017 inauguration, a conversation about my role as a JSF would often be met with "How could you continue given the new administration … ?". On the other hand, the change of U.S. administration has not lessened the impact of computing on today's world, and therefore the urgency of computing policy discussions has not decreased either. In terms of specific technical policy topics (like spectrum coordination) the U.S. position has not changed much. Some see (as I often was able to) that as long as one is being listened to, and as long as one is not being asked to go against ethics or beliefs, then having sensible tech-savvy people continue to engage with any administration is a good thing.

If technologists do not speak up to educate policymakers, bad policy will result, and we will have to live with it.

When I ended my service in April 2017, it was four months earlier than my original one-year renewal for remote work, but six months after election day. In April, my office at the State Department included many of the same capable and devoted civil servants, covering many of the same issues, as one year earlier before the election. So, my decision to end my service early was, to my considerable surprise, more about how best to have effective impact than about the presidential administration per se, despite my disagreements with it. In essence, my decision was that there is a wide range of impactful and much-needed ways to help steer our policymakers in the right direction. Mailing back my badge in April was not the end of my work in that regard, but just the finalization of a decision to find other ways to do so.

How can computer scientists help shape policy for the better and ensure maximum societal benefit from our breakthroughs in computer science? First and foremost, there are many important policy issues—IoT security, network neutrality, trustworthy and fair AI, drones, and more—where computing practitioners and researchers can and should weigh in from the technical side. In the U.S., the FCC, NTIA, NIST, and others are often inviting stakeholders and the general public to participate in policymaking processes on these and other topics. Just in the past month, I have cringed at statements I have seen (on policy-oriented public email lists and in public meetings) made by policymakers with unclear appreciation for how security patches might be applied to IoT systems, or for how AI algorithms work, and for where opacity or unfairness might lead to undesirable results. If technologists do not speak up to educate policymakers, bad policy will result, and we will all have to live with it, as people and as technologists. There are also many lawmakers in the U.S. and elsewhere who welcome input on technical issues; working with them and their staff to explain technical issues and their implications is another way to be more engaged. In addition to issues being covered at the U.S. federal level, there are also state and local issues around data privacy, CS education, and other topics that are of critical importance. Finally, working with the press—both technical and general—is another avenue for our knowledge to be parlayed into broader understanding and action.

I particularly implore my *academic* colleagues to speak. While larger tech companies have dozens of public policy advocates engaged full-time in these conversations, the longer-term and more company-neutral technical viewpoints from academia must be heard as well. This includes advocacy for research and research funding, of course. Importantly though, it also goes well beyond funding. The academic CS community must commit time and energy to advocacy for tech policy paths that are technically rational, that benefit society, and that are decoupled from the motives of a particular company or its shareholders.

At this moment in time, with computing central to our world and yet science often being undervalued or even denigrated by some policymakers, computer scientists must continue to pursue ways and expand our efforts to share our technical knowledge for the good of society. Think of one way to share your technical knowledge with the broader public or government today, and continue with these steps each week. Eventually we will either have governance and policymaking that does not make a computer scientist cringe, or we will at least know that it is not for lack of our attention. C

Margaret Martonosi (mrm@princeton.edu) is an ACM Fellow, the H.T. Adams '35 Professor of Computer Science and the Director of the Keller Center for Innovation in Engineering Education at Princeton University, Princeton, NJ, USA.

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.

For more information: www.acm.org/academic-dept-membership

acm
Association for
Computing Machinery

DOI:10.1145/3154011

Article development led by acmqueue
queue.acm.org

Expert-curated guides to the best of CS research.

BY MALTE SCHWARZKOPF

Research for Practice: Cluster Scheduling for Datacenters

THIS INSTALLMENT OF Research for Practice features a curated selection from Malte Schwarzkopf, who takes us on a tour of distributed cluster scheduling, from research to practice, and back again. With the rise of elastic compute resources, cluster management has become an increasingly hot topic in systems R&D, and a number of competing cluster managers including Kubernetes, Mesos, and Docker are currently jockeying for the crown in this space. Interested in the foundations behind these systems, and how to achieve fast, flexible, and fair scheduling? Malte's got you covered!

—*Peter Bailis*

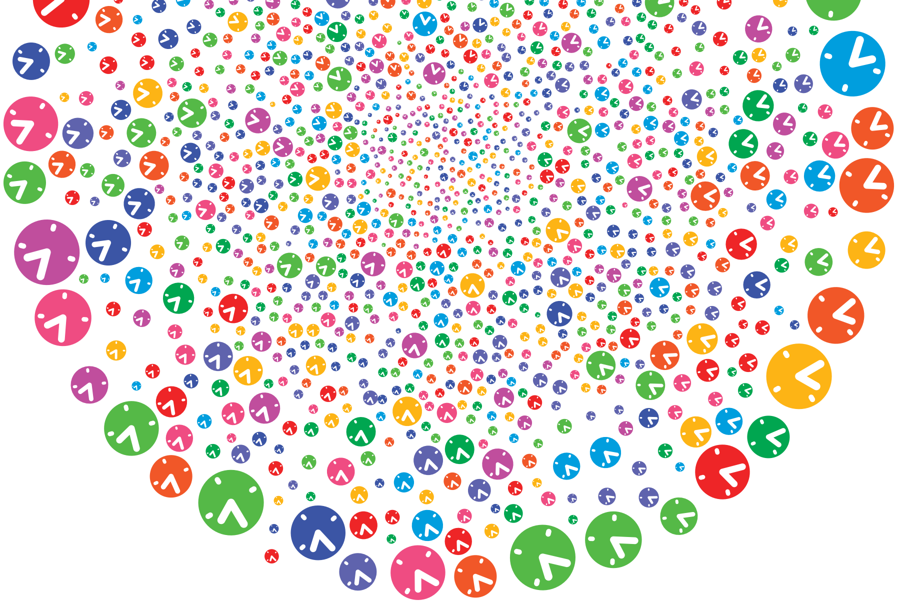
Peter Bailis is an assistant professor of computer science at Stanford University. His research in the Future Data Systems group (futuredata.stanford.edu) focuses on the design and implementation of next-generation data-intensive systems.

Increasingly, many applications and websites rely on distributed backends running in cloud datacenters. In these datacenters, clusters of hundreds or thousands of machines run workloads ranging from fault-tolerant, load-balanced Web servers to batch data-processing pipelines and distributed storage stacks.

A *cluster manager* is special "orchestration" software that manages the machines and applications in such a datacenter automatically: some widely known examples are Kubernetes, Mesos, and Docker Swarm. Why are cluster managers needed? Most obviously because managing systems at this scale is beyond the capabilities of human administrators. Just as importantly, however, automation and smart resource management save real money. This is true both at large scale—Google estimates that its cluster-management software helped avoid building several billion-dollar datacenters—and at the scale of a startup's cloud deployment, where wasting hundreds of dollars a month on underutilized virtual machines may burn precious runway.

As few academic researchers have access to real, large-scale deployments, academic papers on cluster management largely focus on *scheduling* workloads efficiently, given limited resources, rather than on more operational aspects of the problem. Scheduling is an optimization problem with many possible answers whose relative goodness depends on the workload and the operator's goals. Thinking about solutions to the scheduling problem, however, has also given rise to a vigorous debate about the right architecture for *scalable* schedulers for ever larger clusters and increasingly demanding workloads.

Let's start by looking at a paper that nicely summarizes the many facets of a full-fledged industry cluster manager, and then dive into the scheduler architecture debate.



Google's "Secret Sauce"

Verma, A. et al.
Large-scale cluster management at Google with Borg. In *Proceedings of the 10th European Conference on Computer Systems*, 2015, 18:1–18:17; http://dl.acm.org/citation.cfm?id=2741964.

This paper is mandatory reading for anyone who wants to understand what it takes to develop a full-fledged cluster manager and to deploy it effectively. Borg handles all aspects of cluster orchestration: it monitors the health of machines, restarts failed jobs, deploys binaries and secrets, and oversubscribes resources just enough to maintain key SLOs (service-level objectives), while also leaving few resources to sit idle.

To achieve this, the Borg developers had to make many decisions: from choosing an isolation model (Google uses containers), to how packages are distributed to machines (via a torrent-like distribution tree), how tasks and jobs find each other (using a custom DNS-like naming system), why and how low-priority and high-priority workloads share the same underlying hardware (a combination of clever oversubscription, priority preemption, and a quota system), and even how to handle failures of the Borgmaster controller component (Paxos-based failover to a new leader replica). There is a treasure trove of neat tricks here (for example, the automated estimation of a task's real resource needs in §5.5), as well as a ton of operational experience and sound distributed-system design.

A related *Queue* article describes how Borg and Omega, also developed at Google, have impacted Kubernetes, an open source cluster manager developed with substantial inspiration from Borg. The features Kubernetes offers are currently far more limited than Borg's, but Kubernetes catches up further with every release.

The actual *scheduling* of work to machines comprises only one subsection of the Borg paper (§3.2), but it is both challenging and crucially important. Many standalone papers have been written solely about this scheduling problem, often proposing improvements that seemingly might apply to Borg. Unfortunately, they can at times be quite confusing: some describe schedulers that operate at a higher level than Borg, or ones designed for workloads that differ from Google's mix of long-running service jobs and finite-runtime batch-processing jobs. Fortunately, the next paper describes a neat way of separating some of these concerns.

▸ https://dl.acm.org/citation.cfm?id=2898444

▸ https://dl.acm.org/citation.cfm?id=2465386

Many Schedulers: Offers or Requests?

Hindman, B. et al.
Mesos: A platform for fine-grained resource sharing in the datacenter. In *Proceedings of the Usenix Conference on Networked Systems Design and Implementation*, 2011, 295–308; http://static.usenix.org/events/nsdi11/tech/full_papers/Hindman_new.pdf.

Mesos was the first academic publication on modern cluster management and scheduling. The implementation is open source and appeared at a time when Borg was still unknown outside Google. The Mesos authors had the key insight, later reaffirmed by the Borg paper, that dynamically sharing the underlying cluster among many different workloads and frameworks (for example, Hadoop, Spark, and TensorFlow) is crucial to achieving high resource utilization.

Different frameworks often have different ideas of how independent work units (tasks) should be scheduled; indeed, the frameworks that Mesos targeted initially already had their own schedulers. To arbitrate resources among these frameworks without forcing them into the straitjacket of a single scheduling policy, Mesos neatly separates two concerns: a lower-level *resource manager* allocates resources to frameworks (for example, subject to fairness constraints), and the higher-level *framework schedulers* choose which specific tasks to run where (for example, respecting data locality preferences). An important consequence of this design is that Mesos can support long-running service tasks just as well as short, high-throughput batch-processing tasks—they are simply handled by different frameworks.

Mesos avoids having a complex API for frameworks to specify their resource needs. By inverting the interaction between frameworks and the resource manager: instead of frameworks *requesting* resources, the Mesos resource manager *offers* resources to frameworks in turn. Each framework takes its pick, and the resource manager subsequently offers the remaining resources to the next one. Using appropriately sized offers, Mesos can also enforce fairness policies across frameworks, although this aspect has in practice turned out to be less important than perhaps originally anticipated.

The Mesos offer mechanism is somewhat controversial: Google's Omega paper observed that Mesos must offer all cluster resources to each framework to expose the full knowledge of existing state (including, for example, preemptible tasks), but that a slow framework scheduler can adversely affect other frameworks and overall cluster utilization in the absence of optimistic parallel offers and a conflict-resolution mechanism. In response, the Mesos developers devised concepts for such extensions to Mesos.

The multischeduler design intro-

IMAGE BY AHA-SOFT

Scheduling is an optimization problem with many possible answers whose relative goodness depends on the workload and the operator's goals.

duced by Mesos has been quite impactful: many other cluster managers have since adopted similar architectures, though they use either request-driven allocation (for example, Hadoop YARN) or an Omega-style shared-state architecture (for example, Microsoft's Apollo and HashiCorp's Nomad).

Another deliberate consequence of the offer-driven design is that the Mesos resource manager is fairly simple, which benefits its scalability. The next two papers look deeper into this concern: the first one proposes an even simpler design for even greater scalability, and the second suggests that scaling a complex scheduler is more feasible than widely thought.

▸ http://mesos.apache.org/
▸ https://dl.acm.org/citation.cfm?id=2465386
▸ https://dl.acm.org/citation.cfm?id=2523633
▸ http://bit.ly/2sqMUmP
▸ https://www.nomadproject.io/

Breaking the Scheduler Further Apart

Ousterhout, K. et al.
Sparrow: Distributed, low-latency scheduling. In *Proceedings of the Symposium on Operating Systems Principles*, 2013, 69–84; http://dl.acm.org/citation.cfm?id=2522716.

Even within what Mesos considers framework-level tasks, there may be another level of scheduling. Specifically, some data-analytics systems break their processing into many short work units: Spark, for example, generates *application-level* "tasks" that often run for only a few hundred milliseconds (note that these are different from the *cluster-level* tasks that Borg or Mesos frameworks place!). Using such short tasks has many benefits: it implicitly balances load across the workers that process them; failing tasks lose only a small amount of state; and straggler tasks that run much longer than others have smaller absolute impact.

Shorter tasks, however, impose a higher load on the scheduler that places them. In practice, this is usually a framework-level scheduler, or a scheduler within the job itself (as in standalone Spark). With tens of thousands of tasks, this scheduler might get overwhelmed: it might simply be unable to support the decision throughput required. Indeed, the paper shows that the centralized application-level task scheduler within each Spark job scales to only about 1,500 tasks per second. Queueing tasks for assignment at a single scheduler hence increases their overall "makespan" (the time between task submission and completion). It also leaves resources idle while waiting for new tasks to be assigned by the overwhelmed scheduler.

Sparrow addresses this problem in a radical way: it builds task queues at each worker and breaks the scheduler into several distributed schedulers that populate these worker-side queues independently. Using the "power of two random choices" property, which says that (under certain assumptions) it suffices to poll two random queues to achieve a good load balance, Sparrow then randomly places tasks at workers. This requires neither state at the scheduler, nor communication between schedulers—and, hence, scales well by simply adding more schedulers.

This paper includes several important details that make the random-placement approach practical and bring it close to the choices that an omniscient scheduler would make to balance queue lengths perfectly. As an example, Sparrow speculatively enqueues a given task in several queues, spreading its bets across multiple workers and smoothing head-of-line blocking from other straggler tasks. It can also deal with placement constraints and offer weighted fair sharing across multiple jobs that share the same workers.

Sparrow could in principle be used as a cluster-level scheduler, but in practice works best when it load-balances application-level tasks over long-running workers of a single framework (which, for example, serves queries or runs analytics jobs). At the cluster-scheduler level, task startup overheads normally make tasks below tens of seconds in duration impractical because package distribution, container launch, among others already take several seconds. Consequently, the open source Sparrow implementation supplies a Spark application-level scheduler plug-in.

Finally, while Sparrow's randomized, distributed decisions are scalable, they assume that tasks run within fixed-size resource slots and that queues of equal length amount to equally good choices. Several follow-up papers improve more along these dimensions while maintaining the same distributed architecture for scalability and fault tolerance. One paper, however, looks at whether Sparrow-style wide distribution is really required for scalability.

▸ https://github.com/radlab/sparrow

Can We Have Quality and Speed?

Gog, I. et al.
Firmament: Fast, centralized cluster scheduling at scale. In *Proceedings of the Usenix Conference on Operating Systems Design and Implementation*, 2016, 99–115; https://www.usenix.org/system/files/conference/osdi16/osdi16-gog.pdf.

Distributed decisions improve scalability and fault tolerance, but schedulers must make them in the presence of reduced (and only statistically sampled) information about cluster state. By contract, centralized schedulers, which make all decisions in the same place, have the information to apply more sophisticated algorithms—for example, to avoid overloaded machines. Of course, this applies both at the cluster level and to application-level schedulers.

This paper sets out to investigate whether—fault-tolerance benefits notwithstanding—distribution is indeed necessary for scalability. It notes that it is crucial to amortize the cost of decisions over many tasks, especially if the scheduler supports features that require reconsidering the existing placements, such as task preemption. Think about it this way: If the scheduler picked a task off a queue, looked at a whole bunch of machines, and did some complex calculations just to decide where to put this single task, it cannot scale well to many tasks.

Firmament generalizes the Quincy scheduler, a cool—and sometimes-overlooked—system that uses a min-cost, max-flow constraint solver to schedule batch workloads. The constraint solver always schedules the *entire* cluster workload, not just waiting tasks. Because min-cost, max-flow solvers are highly optimized, their algorithms amortize the work well over many tasks.

Applied naively, however, the Quincy approach cannot scale to large workloads over thousands of machines—the constraint-solver runtime, which dominates scheduling latency, would be unacceptably long. To fix this, Firmament concurrently runs several min-cost, max-flow algorithms with different properties and solves the optimization problem *incrementally* if possible, refining a previous solution rather than starting over.

With some additional tricks, Firmament achieves subsecond decision times even when scheduling a Google-scale cluster running hundreds of thousands of tasks. This allows Sparrow-style application-level tasks to be placed within hundreds of milliseconds in a centralized way even on thousands of machines. The paper also shows there is no scalability-driven need for distributed cluster-level schedulers, as Firmament runs a 250-times-accelerated Google cluster trace with median task runtimes of only four seconds, while still making subsecond decisions in the common case. The simulator and Firmament itself are open source, and there is a plug-in that allows Kubernetes to use Firmament as a scheduler.

The Firmament paper suggests we need not compromise on decision quality to solve a perceived scalability problem in cluster scheduling. Nevertheless, Sparrow-style distributed schedulers are useful: for example, a fault-tolerant application-level load balancer that serves a fixed set of equally powerful workers might well wish to use Sparrow's architecture.

▸ https://dl.acm.org/citation.cfm?id=1629601

▸ https://github.com/camsas/firmament

▸ https://github.com/camsas/poseidon

Research to Practice

What does all this mean for you, the reader? For one, you are almost certainly already using applications that run on Borg and other cluster managers every day. Moreover, if you are running a business that computes on large data or runs web applications (or, especially, both!), you will probably want the automation of a cluster manager. Many companies already do so and run their own installations of Mesos or Kubernetes on clusters of VMs provisioned on the cloud or on machines on their own premises.

The problem of scaling cluster managers and their schedulers to very large clusters, however, is one that most readers won't have to face: only a few dozen companies run such large clusters, and buying resources on AWS (Amazon Web Services) or Google's or Microsoft's clouds is the easiest way to scale. In some cases, scheduler scalability can also be an issue in smaller clusters, however: if you are running interactive analytics workloads with short tasks, a scalable scheduler may give you better resource utilization.

Another important aspect of the scheduling problem is that cluster workloads are quite diverse, and scheduling policies in practice often require substantial hand-tuning using placement constraints. Indeed, this is what makes cluster scheduling different from the multiprocessor scheduling that your kernel does: while most applications are fine with the general-purpose policies the kernel applies to assign processes to cores, current cluster-level placement policies often do not work well for all workload mixes without some manual operator help.

Future Directions

Since it is challenging for humans to develop scheduling policies and good placement heuristics that suit all (or even most) workloads, research on policies that help in specific settings is guaranteed to continue.

Another approach, however, may also be viable. Recent, early results suggest that the abundant metrics data and the feedback loops of cluster-scheduling decisions are a good fit for modern machine-learning techniques: they allow training neural networks to automatically learn custom heuristics tailored to the workload. For example, reinforcement learning can effectively learn packing algorithms that match or outperform existing, human-specified heuristics, and a neural network outperforms human planners in TensorFlow's application-level operator scheduling.

Therefore, future research may raise the level of automation in cluster management even further: perhaps the cluster scheduler will someday learn its own algorithm.

▸ https://dl.acm.org/citation.cfm?id=3005750

▸ https://arxiv.org/abs/1706.04972

Editor's Note: To read this article with embedded hyperlinks, visit https://queue.acm.org/detail.cfm?id=3173558.

Malte Schwarzkopf is a post-doctoral associate in the Parallel and Distributed Operating Systems (PDOS) Group at MIT, Cambridge, MA, USA.

DOI:10.1145/3190566

Article development led by acmqueue
queue.acm.org

Automated canarying quickens development, improves production safety, and helps prevent outages.

BY ŠTĚPÁN DAVIDOVIČ AND BETSY BEYER

Canary Analysis Service

IN 1913, SCOTTISH physiologist John Scott Haldane proposed the idea of bringing a caged canary into a mine to detect dangerous gases. More than 100 years later, Haldane's canary-in-the-coal-mine approach is also applied in software testing.

In this article, the term *canarying* refers to a partial and time-limited deployment of a change in a service, followed by an evaluation of whether the service change is safe. The production change process may then roll forward, roll back, alert a human, or do something else. Effective canarying involves many decisions—for example, how to deploy the partial service change or choose meaningful metrics—and deserves a separate discussion.

Google has deployed a shared centralized service called Canary Analysis Service (CAS) that offers automatic (and often autoconfigured) analysis of key metrics during a production change. CAS is used to analyze new versions of binaries, configuration changes, dataset changes, and other production changes. CAS evaluates hundreds of thousands of production changes every day at Google.

CAS requires a very strict separation between modifying and analyzing production. It is a purely passive observer: it never changes any part of the production system. Related tasks such as canary setup are performed outside of CAS.

In a typical CAS workflow (shown in Figure 1), the rollout tool responsible for the production change deploys a change to a certain subset of a service. It may perform some basic health checks of its own. For example, if pushing a new version of an HTTP server causes a process restart, the rollout tool might wait until the server marks itself as able to serve before proceeding. (This may also inform the deployment speed of the production change. This rollout tool behavior is not specific to canarying.)

This subset of production now constitutes the *canary population*. By conducting an A/B test compared to a control population, CAS answers the question, "Is the canary meaningfully worse?" The control population is a (possibly strict) subset of the remainder of the service. Importantly, CAS is not trying to establish absolute health.

The population should be as fine-grained as possible. For example, an application update can use a global identifier of that particular process, which at Google would be a BNS (Borg Naming Service) path. A BNS path is structured as /bns/<cluster>/<user>/<job name>/<task number>. The job name is a logical name of the application, and task number is the identifier of a particular instance.[2] For a kernel update, the identifier might be machine hostname: clearly, multiple processes can

IMAGE BY NUCLEARTIST

run on the same machine, but (modulo virtualization) you are limited to a single running kernel, so granularity is defined at the machine level. Granularity allows the caller to slice and dice the overall service with no restrictions at runtime, and it makes a static preexisting canary setup unnecessary.

Once the canary population is set up, the rollout tool requests a verdict from CAS. This request specifies the canary and control populations, as well as a time range for each member of that population. An entity's canary status can be ephemeral: the canary only became a canary at some specific point in time; before that time, it did not have the tested production change.

The request also contains a reference to a user-supplied configuration, if one exists. As discussed in the following section, CAS tries to provide value, even without external configuration, by enforcing things we hold generally true.

CAS provides a point-in-time verdict after evaluation: a simple PASS or FAIL, meaning the system is performing either the same or in a dangerously anomalous way. (A third option, NONE, is also possible if underlying infrastructure was unavailable and CAS could not reach a verdict. Clients commonly treat this the same as if they could not reach CAS.) The signal must be clear and unambiguous for the rollout tool to take an action such as rolling forward, rolling back, or alerting a human. CAS intentionally does not provide a confidence score, p-value, or the like: that would imply that the rollout tool has logic to determine when to take a real-world action. Keeping this decision centralized allows better reuse and removes the risk of creating artificial confidence scores from a meaningless heuristic.

Default tooling integration and zero configuration option. CAS quickly gained extensive coverage across all of Google by integrating with all major tools used to change production, including tools to roll out new binaries, configurations, and data sets. Widespread integration required a conservative integration approach: in some cases, concessions in the analysis quality had to be made in favor of not inconveniencing users.

Yet another barrier to entry was removed by not requiring a canary setup in order to start using CAS. If the user does not specify a configuration, default analyses are performed across metrics that can be reasoned about across the board. CAS auto-discovers features of canaried systems, such as whether the binary is C++ or Java, or which RPC (remote procedure call) methods receive significant traffic, then chooses analyses to run (for example, RPC error ratio) for those features. Google's infrastructure homogeneity makes this largely successful.

Figure 1. Diagram of typical CAS workflow.

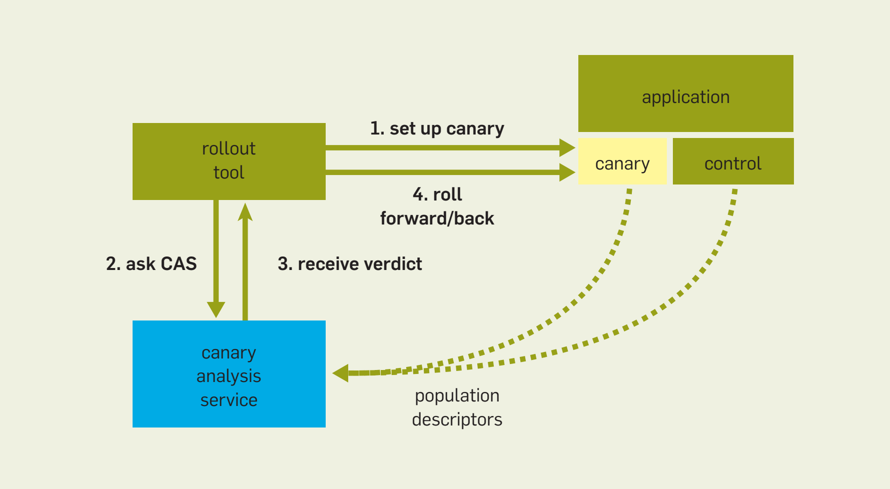
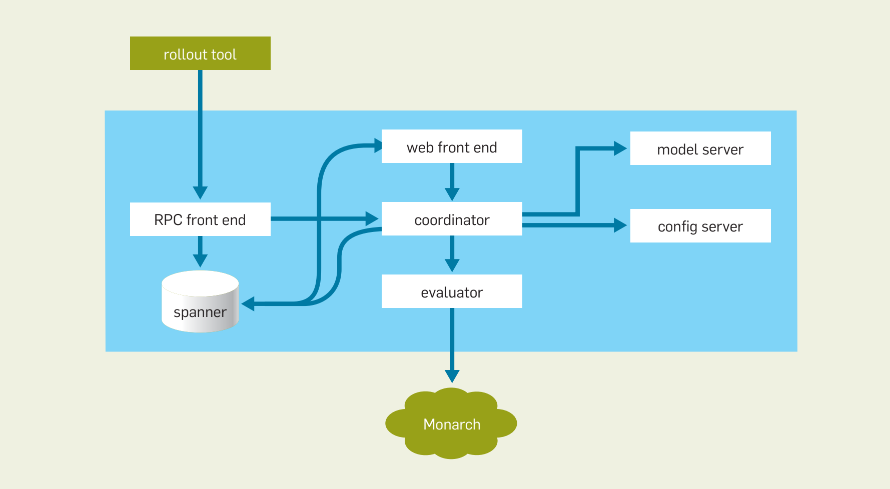


Figure 2. Diagram of CAS main components.



Service Design

CAS's relatively simple high-level interactions are enabled by a fairly complex system under the hood.

Public API in detail. CAS's API has two RPC calls: Evaluate() and GetResult(). Evaluate() is given one or more trials and returns a unique string identifier, Evaluation ID, which is a fully qualified URL to the CAS user interface. This simple trick has made it quite trivial to insert these links into various rollout tools, since it means they do not need any additional client-side logic to figure out how to turn an identifier into a URL. Trials are pairs of canary and control populations and the time range during which they should be compared; if the end time of the time range is left unset, CAS is free to decide how much time the evaluation needs. This means striking a balance between delaying the evaluation too much and not having enough data to reach a meaningful conclusion. In practice, at least five minutes of data are required. The call is reasonably fast (typically under a second), and the API retains the resulting data indefinitely (or at least until garbage collection of evaluations that are clearly no longer relevant). The client sends one RPC for a logical evaluation.

The CAS API does not promise that the evaluation will start when the

`Evaluate()` call is sent. Instead, analysis starts on `GetResult()`, since that indicates someone is interested in the result. As an optimization, the analysis actually starts on `Evaluate()`, but in order to set appropriate expectations with the client, this optimization is not part of the API definition.

`GetResult()` takes one parameter: the Evaluation ID. This RPC blocks until the analysis process is finished, which can take between a few seconds and a few minutes after the end time of the request; `GetResult()` is idempotent.

For the sake of reliability, CAS developers designed the system with two calls. This setup allows the system to resume processing a request without requiring complex client cooperation. This reliability strategy played out in practice when a bug in a library made all CAS processes crash every five to 10 minutes. CAS was still able to serve all user requests, thanks to the robust API.

There are some obvious alternatives to this design. CAS developers decided against using a single long-running RPC: since these calls are fundamentally point-to-point connections between two Unix processes, disruption (for example, because one process restarted) would lead to a full retry from the client side. The original design doc included a large number of options, each with trade-offs tied to nuanced properties of Google's infrastructure and requirements.

Evaluation structure. While the RPC returns only a simple PASS/FAIL verdict, the underlying analysis consists of several components.

The lowest-level unit is a *check*, a combination of time series from the canary population, time series from the control population, and a statistical function that turns both time series into an unambiguous PASS/FAIL verdict. Some example checks might be:

- Crash rate of the canary is not significantly greater than the control.
- RPC error ratio is not significantly greater than the control.
- Size of dataset loaded in memory is similar between canary and control.

As mentioned in the API description, each evaluation request can define multiple trials (that is, pairs of canary and control populations). Evaluation of each trial results in a collection of checks. If any check in any trial fails, the entire evaluation is declared a failure, and FAIL is returned.

CAS's relatively simple high-level interactions are enabled by a fairly complex system under the hood.

Currently, trials are implemented to be fairly independent, though a given evaluation request might have multiple trials if they look at two related but different components. For example, consider an application with a front end and a back end. Changes on the front end can trigger bad behavior on the back end, so you need to compare both:

- The canary front end to the production front end.
- The back end receiving traffic from the *canary* front end to a back end receiving traffic from the *production* front end.

These are different populations, possibly with different metrics, but failure on either side is a potential problem.

Configuration structure. What exactly does a user-defined configuration entail? While the design phase of CAS involved lengthy philosophical discussion about the nature of configuration, the primary aim was simplicity. The CAS developers did not want to force users to learn implementation details to encode their high-level goals into a configuration. The intent was to ask users only a few questions, as close to the user's view of the world as possible.

The individual checks that should be executed for each matching trial define what information is needed. For each check, the user specifies:

- What it should be called.
- How to get the time series for the particular metric.
- How to turn these time series into a verdict.

The user can also include optional pieces of information, such as a long-form description.

Monarch is the typical source of monitoring data for time series.[1] The user specifies an abstract query, and the canary and control populations are determined at *runtime* in the RPC that requests evaluation. CAS has a flexible automatic query rewrite mechanism: at runtime, it rewrites an abstract query to specialize it to fetch data only for a particular population. Say a user configures a query, "Get CPU usage rate." At runtime, CAS rewrites that query as "Get CPU usage rate for job foo-server replicas 0, 1, 2." This rewrite happens for both the canary population and the control, resulting in two queries.

It is possible, although uncommon, to specify different queries for the canary and the control. The queries are still subject to rewriting, which guarantees that they will fetch data only for the objects that are actually being evaluated.

To simplify configuration, there are also *common queries*. These are canned queries curated by the CAS team, such as crash rate, RPC server error ratio, and CPU utilization. These offer known semantics, for which CAS can provide better quality analysis.

Finally, there needs to be a way to turn the time series (possibly multiple streams) obtained by running the Monarch query for canary and control populations into an unambiguous verdict. The user can choose from a family of tests. Some tests (such as Student's t-test) have a clear statistical origin, while others contain custom heuristics that attempt to mimic how a human would evaluate two graphs.

As we will discuss, automatic analyses are applied if a user chooses the default configuration, as well as on user-supplied queries if the user does not specify a statistical test.

System Components and Request Flow

Figure 2 illustrates the components of the CAS system. This section describes the role of each component and the CAS request flow.

Spanner database. The Spanner database is a shared synchronization point for the evaluation flow; almost all components write to it. It is the canonical storage for evaluation progress and final status.

RPC front end. The rollout tool sends `Evaluate()` calls to the RPC front end, which is intentionally very simple. The front end generates a unique identifier for the evaluation, stores the entire evaluation request in the database (with the unique identifier as primary key), and returns the identifier.

`GetResult()` calls also land on the RPC front end, which queries the database to see if a coordinator is already working on the evaluation. If so, the RPC front end sends an `AwaitEvaluation()` RPC to the coordinator, which blocks until the evaluation is complete. If the coordinator is not tracking the evaluation (for example, if a restart resulted in lost state) or if no coordinator is assigned, the RPC front end chooses a coordinator, stores that information in the database, and calls `AwaitEvaluation()`. These retries are limited.

Checkpoints occur after a coordinator receives a fully qualified configuration and asynchronously as evaluators return check-evaluation requests.

If the evaluation has already finished, the RPC front end does not contact the coordinator and immediately returns the results from the database to the caller.

It is very cheap for the RPC front end to handle parallel `GetResult()`s. Selecting one coordinator avoids duplication of expensive work unless the client requests two duplicate and independent evaluations.

Coordinator. The coordinator keeps all evaluations it is currently processing in memory. Upon `AwaitEvaluation()`, the coordinator checks whether the evaluation is being processed. If so, the coordinator simply adds this RPC to the set of RPCs awaiting the result.

If the evaluation is not being processed, the coordinator transactionally takes ownership of the evaluation in the database. This transaction can fail if another coordinator (for whatever reason, such as a race condition) independently takes ownership, in which case the coordinator pushes back to the RPC front end, which then contacts the new canonical coordinator.

Upon receiving a new evaluation, the coordinator does the following:

1. Retrieves fully qualified and unambiguous expanded configuration from the config server. The coordinator now has the full set of all checks to run.
2. Fans out each check to evaluators.
3. Calls the model server to obtain predicted behavior for checks, simultaneously reporting the results of the checks in the current evaluation.
4. Responds to all waiting `AwaitEvaluation()` RPCs with the final verdict.

The coordinator checkpoints progress to the database throughout. Checkpoints occur after a coordinator receives a fully qualified configuration and asynchronously as evaluators return check-evaluation requests. If the coordinator dies, a new one takes over, reads progress from the database, and continues from the last coordinator's checkpoint.

Configuration server. The configuration server looks up and fully expands a configuration that matches an evaluation.

When the configuration is explicitly referenced in a request, lookup is trivial. If the configuration is not explicitly referenced, a set of automatic lookup rules search for the user's *default config*. These lookup rules are based on features such as who owns the canaried service.

The CAS-submitted configuration is generic: it might say something like "Fetch HTTP error rate," without specifying where to fetch the error rate. In the typical flow, the rollout tool identifies the current canary and passes this information along to CAS when the evaluation is requested. As a result, the configuration author cannot necessarily predict the canary population.

To support this flexibility, the configuration server expands configuration and canary/control population definitions to specify exactly what data is requested. For example, the user's "Fetch HTTP error rate" becomes "Fetch HTTP error rate from these three processes for canary data, and from these ten processes for control data." From a user's point of view, after configuring the generic variant, the "right thing" happens automatically, removing any need to define a dedicated canary setup before canarying (although users can define such a setup if they have other reasons to do so).

Besides evaluations, the configuration server also receives configuration updates, validates updates for correctness and ACLs (access control lists), and stores these updates in the database.

Evaluator. The evaluator receives a fully defined configuration (after the expansion already mentioned) for each check, with each check in a separate RPC. The evaluator then:

1. Fetches time series for both canary and control data from the appropriate time series store.
2. Runs statistical tests to turn the resulting pair of sets of time series into a single PASS/FAIL verdict for each statistical test (pair because of canary/control; sets because it is possible, for example, to have a time series per running process and have many processes in the canary or control groups).

If a user configures a statistical test, then the evaluator runs only that test. If the user opts for autoconfiguration, however, the evaluator may run dozens of tests with various parameters, which generate data that feeds into the model server.

The evaluator returns the data from tests and any potential metadata (such as errors talking to time-series stores) to the coordinator.

Model server. The model server performs automatic data analysis. After evaluation, the coordinator asks the model server for predictions. The request contains information about the evaluation and all observed verdicts from the evaluator.

For each observed verdict, the model server returns its expected verdict for that particular evaluation. It returns this information to the coordinator, which ignores results of statistical functions for predicted failures when deciding the overall verdict. If the model server predicts failure because said failure is typical behavior, this behavior is deemed a property of the evaluated system and not a failure of this particular canary evaluation.

Autoconfiguration

Canarying properly is a complex process, as the user needs to accomplish these nuanced tasks:

- Correctly identify a meaningful canary deployment that creates a representative canary population with respect to the evaluation metrics.
- Choose appropriate evaluation metrics.
- Decide how to evaluate canaries as passing or failing.

CAS eases the burden by removing the most daunting of these tasks: evaluating what it means for a time-series pair to pass or fail. CAS builds upon the underlying argument that running reliable systems should not require in-depth knowledge of statistics or constant tuning of statistical functions' parameters.

CAS uses behavior learning that is slightly different from the general problem of anomaly detection for monitoring. In the CAS scenario, you already know that a service is being changed, and exactly where and when that change takes place; there is also a running control population to use as a baseline for analysis. Whereas anomaly detection for monitoring triggers user alerts (possibly at 4 A.M.), bad CAS-related rollouts are far less intrusive—typically resulting in a pause or a rollback.

Users can opt out of autoconfiguration by specifying a test and its parameters manually.

Online behavior learning. In the simplest terms, we want to determine the typical behavior of the system being evaluated during similar production changes. The high-level assumption is that bad behavior is rare.

This process takes place online, since it must be possible to adapt quickly: if a behavior is anomalous but desirable, CAS fails the rollout; when the push is retried, CAS needs to adapt.

Adaptive behavior poses a risk if a user keeps retrying a push when an anomaly is actually dangerous: CAS eventually starts treating this risky behavior as the new norm and no longer flags it as problematic. This risk becomes less severe as the automation becomes more mature and reliable, as users are less inclined to blindly retry (assuming an incorrect evaluation) and more inclined to actually debug when CAS reports a failure.

Offline supporting processes can supplement the standard online learning.

Breakdown of observations. Intuitively, you know that comparing the same metrics across different binaries may yield different results. Even if you look at the same metric (RPC latency, for example), a stateful service such as BigTable may behave quite differently from a stateless Web search back end. Depending on the binary being evaluated, you may want to choose different parameters from the statistical tests, or even different statistical tests altogether.

Rather than attempting to perform in-depth discovery of potential functional dependencies, CAS breaks down observations across dimensions based upon past experiences with running production systems. You may well discover other relevant dimensions over time.

Currently, the system groups observations by the following factors:

- *Data source.* Are you observing process crash rate, RPC latency, or something else? Each data source is assigned a unique identifier by fingerprinting the configuration and some minor heuristics to remove common sources of unimportant differences.

- *Statistical function and parameters.* This could mean, for example, a t-test with significance level of 0.05. Each distinct statistical function and parameter set is assigned a unique identifier.
- *Application binary.*
- *Geographical location.* This refers to locations of the canary and control.
- *Process age.* Has the process recently restarted? This helps distinguish a configuration push (which might not restart the process) from a binary update (which likely would).
- *Additional breakdowns, such as different RPC methods.* For example, reading a row in BigTable may behave very differently from deleting the entire table. This breakdown depends on the supplied metric.
- *Time of observation.* This is kept at daily granularity for system efficiency.

These factors combine with the count of each observed verdict to make a *model*. A model knows *only* identifiers—it has no understanding of the data source, statistical functions, or their parameters.

Prediction selection. All models pertaining to a particular binary are fetched across all statistical functions for which there is an observation, and across all data sources.

For each statistical function and each data source, the weighted sum of the previously observed behaviors is calculated for each possible result. Similarity is weighted both by heuristic similarity of features (process age and geographical location) and by the age of the model. Because additional breakdowns such as RPC methods do not have a usable similarity metric, the additional matching breakdowns are simply filtered in, with no further weighting.

For a single statistical function and a single data source, we generate a score for each possible verdict (PASS, FAIL, or NONE). We calculate this score from a weighted sum of past observations. Weighting is based upon factors like age of the observation and similarity of the observation to the current situation (for example, do both observations pertain to the same geographic location?).

Each statistical function has a minimum pass ratio. The ratio *sum*[*PASS*] / (*sum*[*PASS*] + *sum*[*FAIL*] + *sum*[*NONE*]) must be greater than the minimum for a PASS prediction. Otherwise, the prediction is FAIL.

This ratio allows CAS to impose a notion of strictness on various functions, while being tolerant of "normal" volatile behavior. For example, consider two statistical functions: one that tolerates only 1% deviation between canary and control, and one that tolerates 10%. The former can be given a very high minimum pass ratio, and the latter a lower one. If the metric fluctuates more than 1% in normal operation, CAS quickly learns that behavior and stops flagging it. If that fluctuation is a one-off, CAS flags it, the system recovers, and over time CAS relearns that normal behavior includes only deviations under 1%. CAS intentionally takes longer to learn normal behavior for larger tolerated fluctuations, so in this example, CAS will learn at a slower rate for the 10% case.

Bootstrapping. When a user initially submits a configuration that evaluates a metric, no past behavior exists to use for prediction. To bootstrap such cases, CAS looks for past evaluations that *could have* used this config and runs those evaluations to collect observations for the model server. With enough recent evaluations, CAS will already have useful data the first time a user requests an evaluation.

If such bootstrapping is not possible, the model server reverts to the most generous behavior possible.

Arbitrary input analysis. The behavior-prediction mechanism is also the first attempt at *arbitrary input analysis*, which allows modeling behavior for tests when there is no prior knowledge of what they are about.

When a user configures canarying on RPC error ratio, CAS knows in advance that the values are between 0.0 and 1.0, and that higher is worse. For a user-supplied query against the monitoring data, CAS has no such knowledge and can only apply a battery of tests and observe the differences.

Despite some significant issues, as we will discuss, the CAS development team chose this approach because they were confident that it would have relatively few unexpected risks. It still greatly improves automated canarying. The developers are actively working on improvements.

Future Work

Time series aggregate models. While the meta-analysis of the results of hardcoded statistical functions has worked well for the initial launch of automatic configuration, this approach is crude and inflexible. Rather than storing results of statistical tests without any knowledge about the time series that caused them, CAS could store data about the time series.

Each statistical function that CAS supports requires different data from the time series. We could attempt to extract constant-size aggregate views on this data, one for each statistical test. For example, a student's t-test view on the time series could be the mean value for both populations, the population sizes, and variance estimation.

This aggregated view from many past observations would allow synthesizing a single test for each statistical function, with the correct parameters chosen based on past data and some policy.

This work would essentially replace half of the current autoconfiguration system.

Observation breakdowns turned out to be the biggest contribution of the model server to CAS as a whole, so the development team plans to expand this feature. Adding more breakdowns entails additional computational/storage costs and, therefore, needs to be undertaken carefully given CAS's large scale.

While CAS currently has breakdowns based on the object of evaluation, this could be expanded to breakdowns by type of canarying. Anecdotally, there have been major differences in canary behavior when observed using before/after tests versus simultaneous tests of two populations. The size of the canary population in relation to the control population and the absolute sizes of the populations can also provide meaningful breakdowns.

Future work could determine if these additional breakdowns are worthwhile, and at what granularity to perform them. Automatically generated decision trees may also be an option.

Priming with steady state data. CAS sees only production changes. Currently, it does not learn that a particular metric is erratic even in steady state.

Data about metric behavior outside of production changes could be used to define the typical noise in the data. CAS would fail a canary only if the deviation is above this typical noise level. The noise data could come from analyzing only the control population for every evaluation, because the control population is expected to have no production changes.

Known Issues

Same environment overfitting. CAS autoconfiguration's most significant issue is overfitting data when there is already a rich history of past observations in exactly the same environment. In this scenario, only the historical data of that environment is used.

This behavior has some caveats. Consider a rollout of a new version of a system that takes twice as long to handle each RPC call but does a significantly better job. CAS would flag the longer RPC handling time as anomalous behavior for each geographical location of the rollout, causing the release owner undue hardship. The mitigation is to adjust the heuristics carefully in selecting relevant environments to include data beyond the perfect match.

User mistrust. CAS is useful but far from perfect. It has experienced incidents when users disregarded a canary failure and pushed a broken release. User mistrust in complex automation is at the root of many of these issues.

The CAS developers are tackling this mistrust by explicitly explaining, in human-friendly terms that do not require knowledge of statistics, why CAS reaches a particular conclusion. This includes both textual explanation and graphical hints.

Relative comparisons only. Because the model server stores only the outcomes of statistical functions without knowing the input values, CAS does not know the typical values for a time series.

Not knowing the semantics of the data implies that the tests being run are purely relative comparisons, such as having a t-test with null hypothesis that the metric did not increase by more than 5%. While relative comparisons are easy to reason about, they behave extremely poorly if the provided time series value is typically zero, or if a large relative change occurs in absolute numbers too small to be important to the service owner.

Observation breakdowns turned out to be the biggest contribution of the model server to CAS as a whole, so the development team plans to expand this feature.

This is a significant limitation of the mechanism. While it has not had much practical impact in real-world operation, especially given existing trivial workarounds, it merits improvement. Numerous improvements can be made to this mechanism, some quite simple. In addition to the future work mentioned previously, candidates include standard deviation analysis and looking at past observed behavior of the metric.

Scale limitations on the input values. As CAS uses only a hard-coded set of statistical functions and their parameters, the system is somewhat inflexible about analyzing inputs outside of the expected input scale. For example, if the range of 1% through 100% difference is covered, what about the systems and metrics where a difference of 200% is normal? What if even a 1% difference is unacceptable?

CAS developers did not anticipate this to be a significant limitation in practice, which thankfully proved true. Most metrics meriting canary analysis turn out to contain some noise; conversely, most of our A/B testing hopes to see little difference between the two populations, so large differences are unexpected and therefore noticed.

Lessons Learned

Good health metrics are surprisingly rare. The best way to use CAS is to employ a few high-quality metrics that are clear indicators of system health: suitable metrics are stable when healthy, and they drastically change when unhealthy.

Often, the best canarying strategy is to choose metrics tied to SLOs (service-level objectives). CAS automatically integrates with an SLO tracking system to apply servicewide SLOs and some heuristics to scale them appropriately to the canary size.

Setting an SLO is a complex process connected to business needs, and SLOs often cover an entire service rather than individual components. Even if a canary of a single component misbehaves in the extreme, its impact on a service's overall SLO can be small. Therefore, key metrics need to be identified (or introduced) for each component.

It's tempting to feed a computer all the metrics exported by a service. While Google systems offer vast amounts of telemetry, much of it is useful only for debugging narrow problems. For example, many BigTable client library metrics are not a direct indication that a system is healthy. In practice, using only weakly relevant metrics leads to poor results. Some teams at Google have performed analysis that justifies using a large number of metrics, but unless you perform similarly detailed data analysis, using only a few key metrics yields much better results.

Perfect is the enemy of good. Canarying is a very useful method of increasing production safety, but it is not a panacea. It should not replace unit testing, integration testing, or monitoring.

Attempting a "perfectly accurate" canary setup can lead to a rigid configuration, which blocks releases that have acceptable changes in behavior. When a system inherently does not lend itself to a sophisticated canary, it's tempting to forego canarying altogether.

Attempts at hyper-accurate canary setups often fail because the rigid configuration causes too much toil during regular releases. While some systems do not canary easily, they are rarely *impossible* to canary, though the impact of a having a canary process for that system may be lower. In both cases, switching to a strategy of gradual onboarding of canarying, starting with low-hanging fruit, will help.

Impact analysis is very hard. Early on, the CAS team asked, "Is providing a centralized automatic canarying system worth it?" and struggled to find a answer. If CAS actually prevents an outage, how do you know the impact of the outage and, therefore, the impact of CAS?

The team attempted to perform a heuristic analysis of production changes, but the diverse rollout procedures made this exercise too inaccurate to be practical. They considered an A/B approach where failures of a subset of evaluations were ignored, passing them in order to measure impact. Given the many factors that influence the magnitude of an outage, however, this approach would not be expected to provide a clear signal. (Postmortem documents often include a section such as "where we got lucky," highlighting that many elements contribute to the severity of the outage.)

Ultimately, the team settled upon what they call *near-miss analysis*: looking at large postmortems at Google and identifying outages that CAS *could* have prevented, but did not prevent. If CAS did not prevent an outage because of missing features, those features were identified and typically implemented. For example, if CAS could have prevented a $10M postmortem if it had an additional feature, implementing that feature proves a $10M value of CAS. This problem space continues to evolve, as we attempt other kinds of analyses. Most recently, the team has performed analysis over a (more homogeneous) portion of the company to identify trends in outages and postmortems, and has found some coarse signal.

Reusability of CAS data is limited. CAS's immense amount of information about system behaviors could potentially be put to other uses. Such extensions may be tempting at face value, but are also dangerous because of the way CAS operates (and needs to operate at the product level).

For example, the CAS team could observe where canaries behave best and recommend that a user select only that geographical location. While the recommended location may be optimal *now*, if a user followed the advice to canary only in that location, the team's ability to provide further advice would lessen. CAS data is limited to its observations, so behavior at a local optimum might be quite different from the global optimum.

Conclusion

Automated canarying has repeatedly proven to improve development velocity and production safety. CAS helps prevent outages with major monetary impact caused by binary changes, configuration changes, and data pushes.

It is unreasonable to expect engineers working on product development or reliability to have statistical knowledge; removing this hurdle—even at the expense of potentially lower analysis accuracy—led to widespread CAS adoption. CAS has proven useful even for basic cases that do not need configuration, and has significantly improved Google's rollout reliability. Impact analysis shows that CAS has likely prevented hundreds of postmortem-worthy outages, and the rate of postmortems among groups that do not use CAS is noticeably higher.

CAS is evolving as its developers work to expand their scope and improve analysis quality.

Acknowledgments

A great number of people contributed key components of this work. Thanks to Alexander Malmberg, Alex Rodriguez, Brian O'Leary, Chong Su, Cody Smith, Eduardo Blanco, Eric Waters, Jarrod Todd, Konstantin Stepanyuk, Mike Ulrich, Nina Gonova, Sabrina Farmer, Sergey Kondratyev, among others. Also, thanks to Brian O'Leary, and Chris Jones for technical review. C

Related articles
on queue.acm.org

Fail at Scale
Ben Maurer
http://queue.acm.org/detail.cfm?id=2839461

The Verification of a Distributed System
Caitie McCaffrey
https://queue.acm.org/detail.cfm?id=2889274

Browser Security:
Lessons from Google Chrome
Charles Reis, Adam Barth, Carlos Pizano
http://queue.acm.org/detail.cfm?id=1556050

References

1. Banning, J. Monarch, Google's planet-scale monitoring infrastructure. Monitorama PDX 2016; https://vimeo.com/173607638.
2. Van Winkel, J.C. The production environment at Google, from the viewpoint of an SRE, 2017. https://landing.google.com/sre/book/chapters/production-environment.html.

Štěpán Davidovič is a Site Reliability Engineer at Google, where he works on internal infrastructure for automatic monitoring. In previous Google SRE roles, he developed Canary Analysis Service and has worked on AdSense and many shared infrastructure projects.

Betsy Beyer is a technical writer for Google Site Reliability Engineering in New York, NY, USA, and the editor of *Site Reliability Engineering: How Google Runs Production Systems*. She has previously written documentation for Google's Data Center and Hardware Operations teams and lectured on technical writing at Stanford University.

DOI:10.1145/3174795

Article development led by acmqueue
queue.acm.org

Praise matters just as much as money.

BY KATE MATSUDAIRA

How Is Your Week Going So Far?

I HAVE TO say, this week I am walking on sunshine. I am getting a lot done and feeling really good while getting it all done. Which is pretty surprising, since earlier this week, I felt completely overwhelmed and a little defeated.

I had been feeling overwhelmed with the large amount of work on my plate (both at home and at the office), every new task added to the list just made me feel more and more tired. Less and less excited. More and more overwhelmed.

But then something kind of amazing happened. And it was amazing because it was so small. I got to work, and shortly thereafter, I received an email message highlighting some recent wins that came out of work I had just done.

In fact, the message included this day-maker: "*Amazing job on the presentation!!!!!!!*"

Oh yeah, those are *seven* exclamation points.

Ever wondered how to make your team more productive, more excited, and more motivated? It's really simple. It's so ridiculously simple.

Seven exclamation points completely changed the trajectory of my week.

You had better believe my spirits were lifted and I kept working even harder after hearing that my work was not only appreciated, but that it also helped us achieve some goals.

Why did this work?

Because praise is one of the most meaningful ways to connect with the people on your team and motivate them to do more amazing work.

Nobody comes to work to do a bad job. Most of us are doing our best.

Even so, it's rare that we hear how our work is being received. We assume if we hear nothing that it means we are not in trouble, which is good. But it's not great.

A Gallup study found that more than two-thirds of employees do not receive any praise in a given week.

Which is surprising, given the research that shows getting "praise or recognition for good work" increases revenue and productivity 10% to 20% and that those feeling unrecognized are three times more likely to quit in the next year.

Praise Is Difficult

Giving praise is difficult. It can be awkward. It can feel unnecessary.

You might think, "My team already knows I think the work they do is awesome."

And you know what? You might be right. They might already know you appreciate them. But that does not counteract their need to hear that you still think they are awesome.

It's not just the knowledge that your boss values you and your work that matters. Hearing it, out loud, for specific projects is what really matters. It is what sustains people. It is what motivates them.

Hearing praise releases oxytocin in our brains, a hormone that fuels trust and bonding. Simply put, hearing how much our work is appreciated makes us want to do more to repeat that feeling by pleasing the people we work with.

In fact, when I think about my past, one of my biggest motivations for being amazing in previous roles was being recognized for being amazing.

The recognition and approval I received from my leaders and peers were just as important as the raises and promotions I received for being great at my job.

Praise matters just as much as money. So, how do you do it right?

How to Praise Your Team Effectively

Valuable praise has the same three elements. If you add these together, the praise you are giving will be meaningful and motivating to your team.

It's like a super-simple math equation for motivation: to be effective, praise must be frequent, specific, and strategic.

Frequent. When you do not praise your team regularly, they do not know where they stand with you. They may make assumptions based on limited information such as your demeanor in a meeting or a face you made in passing. When your team has little to go on (or they hear from you only when things are wrong), they do not have enough information to know you (secretly) appreciate their work.

Never forget that as a manager, your opinion matters to your team and they are constantly looking to you for information about their status.

Plus, negative comments last a lot longer in our brains than positive ones. This is why frequent praise matters.

It has been said that it takes six positive interactions to overcome one negative interaction—keep that in mind, especially if you are a hard-driving manager who demands the best. Make sure your team hears more of the good than the bad.

Specific. How many email messages have you gotten with a "Thanks!" or "Good job" tacked onto the end of it? It doesn't quite have the enthusiastic effect the sender probably meant for it to have.

As a manager, when you praise your team, you need to tell them *exactly* what you liked in their work in order for it to have any value to them. Was it the way they commented their code? Did they give a detailed, efficient, and prompt answer in a support question? Were they able to take control of a bad situation and get everyone quickly working toward a good solution?

If you acknowledge specifically what you liked about what they did, they will know that you really paid attention and they will know exactly what to do to be praised again.

Researchers have found that the highest driver of work engagement is whether workers feel their managers are genuinely interested in them and their well-being. Think about how many times in your own career you have said to yourself things like, "I don't think they even notice what I do."

Be clear about praising specific work that you are grateful for or that has had a big impact. This will go a long way toward fighting burnout and building an amazingly motivated team (especially if you take the time to look for unsung and overlooked heroes on projects).

Strategic. Are you convinced praise is a good thing? Well, it gets better. You can actually use praise to develop your people and build a more amazing team.

To do this, choose a skill you want each team member to add or improve on. Work on this with each person, and any time you see improvement or good work, praise the person specifically for it.

You will see that person light up and keep getting better until reaching the level you want. Occasionally praising the things you know that someone has always been good at will also keep that person from feeling like no one notices his or her ongoing hard work.

What you reward and recognize is what you get. If you do not recognize anything, the bar will lower to see what gets noticed (or what they can get away with). When you do praise and reward your team, you raise the bar based on what gets praised.

Can You Be the Manager You Always Wished You Had?

None of us hears "thank you" or "awesome job" enough at work. Being the person who praises other people is an amazing person to be, especially when you follow this formula for making your praise ridiculously effective.

What could you accomplish if you had the best team in your company? Imagine what you could do if you had a team that was so successful and so motivated that you could take a long vacation without worrying about what was going on at the office?

Stop thinking about it, and start doing. Set a reminder on your calendar to give more praise every week. C

Related articles on queue.acm.org

The Debugging Mindset
Devon H. O'Dell
https://queue.acm.org/detail.cfm?id=3068754

The Paradox of Autonomy and Recognition
Kate Matsudaira
https://queue.acm.org/detail.cfm?id=2893471

Broken Builds
Kode Vicious
https://queue.acm.org/detail.cfm?id=1740550

Kate Matsudaira (katemats.com) is an experienced technology leader. She has worked at Microsoft and Amazon and successful startups before starting her own company, Popforms, which was acquired by Safari Books.

IMAGE BY KAPITOSH

Trust Extension as a Mechanism for Secure Code Execution on Commodity Computers
Bryan Jeffrey Parno
Text Data Management and Analysis
A Practical Introduction to Information Retrieval and Text Mining
ChengXiang Zhai
Sean Massung
The Sparse Fourier Transform
Theory and Practice
Haitham Hassanieh
Frontiers of Multimedia Research
Shih-Fu Chang
Editor
Smarter than Their Machines
Oral Histories of Pioneers in Interactive Computing
John Cullinane
The VR Book
Human-Centered Design for Virtual Reality
Jason Jerald, Ph.D.
Verified Functional Programming in Agda
Aaron Stump
A Framework for Scientific Discovery through Video Games
Seth Cooper
An Architecture for Fast and General Data Processing on Large Clusters
Matei Zaharia
The Handbook of Multimodal-Multisensor Interfaces
Volume 1
Foundations, User Modeling, and Common Modality Combinations
Sharon Oviatt
Björn Schuller
Philip Cohen
Daniel Sonntag
Gerasimos Potamianos
Antonio Krüger

DOI:10.1145/3132699

A teacher and students coding together make explicit the unwritten rules of programming.

BY JOSH TENENBERG, WOLFF-MICHAEL ROTH, DONALD CHINN, ALFREDO JORNET, DAVID SOCHA, AND SKIP WALTER

More Than the Code: Learning Rules of Rejection in Writing Programs

LEE SHULMAN, A PAST PRESIDENT of the Carnegie Foundation for the Advancement of Learning, identifies the signature pedagogies of professions as those characteristic forms of teaching and learning that "define what counts as knowledge in a field and how things become known."[13] If there is a signature pedagogy for computing, it is surely the writing of code together between a teacher and students: The teacher poses a problem, solicits input from students about how to write the code to solve this problem, and weaves together the suggestions from different students into a coherent whole. This pattern is so pervasive that anyone who has spent time in a computer science classroom has participated in this pedagogical form.

Given its ubiquity, one would think this signature pedagogy is well understood. Yet our search of the educational literature yielded not a single prior research study focused on this signature pedagogy of computer science. Our aim here is to fill this gap. In particular, we provide an empirical account of the joint—irreducibly social—work that programming students and teachers perform. We ask: What is it that occurs when a teacher and students write code together? What is *learn-able* about that joint work? And how can it further inform computing teaching and learning practice?

Programming as a Social Practice

In addressing these questions, our study approaches programming as a *social* practice. Almost 30 years ago, Sherry Turkle and Seymour Papert[17] described the common conception of programming as one that is fundamentally formal: "The prevailing image of the computer is that of a logical machine, and . . . programming is seen as a technical and mathematical activity." No one represents this view better than Edsger Dijkstra,[4] who described programs as "rather elaborate formulae from some formal system" and "an abstract symbol manipulator which can be turned into a concrete one by supplying a computer to it." Accordingly, the activity of programming is a technical exercise of the individual programmer who produces a correct program from a specification. According to this view, it is unsurprising that much computing education research has been devoted to categorizing and quantifying student errors and mis-

» key insights

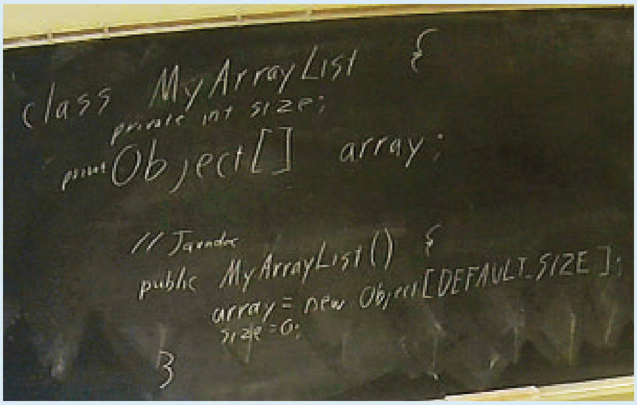
- Programming can be viewed as a social practice structured by tacit "rules of the game" rather than a formal exercise linking specifications to code.
- An empirical investigation of a joint code-writing session between a teacher and students shows how rules of programming are uncovered in the reasons given for rejecting proposed lines of code.
- While the code for a program is routinely captured in normative forms of instruction and student notes, the rejection rules generally go undocumented.

conceptions in comparison to the "correct," or canonical, behavior of experts in carrying out this formal activity.[8,14]

Over the past several years, however, a different view of programming has emerged, one that sees programming as *inherently* social. In a 2016 *Communications* article, Kafai[6] states that "[c]oding was once a solitary, tool-based activity. Now it is becoming a shared social practice. Participation spurred by open software environments and mutual enthusiasm shifts attention from programming tools to designing and supporting communities of learners." This shift of attention from the isolated individual to participation in social practice has led to considerable research advances in the educational literature over the past several decades.[10,11,16] Research in computing education is beginning to appear that draws on sociocultural theories of cognition and learning and their methods of empirical investigation. This is consistent with a 2014 report,[2] which states that "many questions [in computing education] remain unanswered and would benefit from contemporary research in the learning sciences in sociocultural and situated learning, distributed and embodied cognition, as well as activity, interaction and discourse analysis."[2] It is in this gap in the literature on the social practices within computing classrooms that this study is situated. An important question to consider, then, is: What does it mean for a practice to be social rather than individual? A practice is social because practitioners

IMAGE BY KOSTSOV

Figure 1. Presented code from first part of the episode.



```
class MyArrayList {
  private int size;
  private Object[] array;

  // Javadoc
  public MyArrayList() {
    array = new Object[DEFAULT_SIZE];
    size = 0;
  }
```

act as if they were following specific rules of the game, and an observer can read these rules off from the practice of the game even if the players never articulated or are aware of the rules.[18] It is with this definition of social practice as the joint work of making rules present in and through practice that we turn to an analysis of joint code writing.

Analysis of a Joint Code-Writing Episode

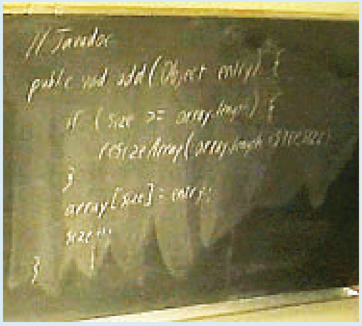
To illustrate how learning to program can be approached as social practice, and how such an approach can be relevant to computing education, we present and analyze an episode from a data structures course taught at a small private university in the U.S. The episode has been culled from a database that includes video recordings, photographs, ethnographic observations, and interviews from the entire course, which met for 50 minutes three times per week over a 15-week term. The students and teacher also met for an additional session each week for two hours in a large computer laboratory. Audio-visual recordings were made in both settings from two cameras placed on a ceiling-mounted overhead projector in the center of the room, one pointing to the front of the room and one pointing to the side, augmented by and later synchronized with audio recorders placed around the room.

Chalk and talk. The episode begins 21 minutes into a class session during the second week of the term. In attendance are 30 students occupying most of the 37 seats in the room. The teacher, Alan (a pseudonym), is in his fifth year of teaching in higher education, doing the data structures course for the third time. Alan stands at the front of the classroom, in front of a blackboard of three equal sections that spans the entire front wall. He begins the episode saying,[a] "So the question is (0.4) how do you make something like this (1.6) if you were to code up your own ArrayList (0.5) which incidentally is this week's homework assignment (2.9) how would you do it (2.8) now one approach you could do [Erases center board, writes 'class MyArrayList {' at the top of this section]."

Alan makes two statements that have the syntactic form of questions. Yet in neither case does a student respond. For the second question, Alan responds to his own question and begins a presentation that lasts almost nine minutes, with only two students speaking during that time for a total of less than 20 seconds. Code is presented as a fait accompli, a rationale preceding each line of code (such as "because constructors have no return types"). The teacher speaks and writes, whereas the students align their bodies and gaze in the direction of the teacher and what he is writing on the board, occasionally looking down to their desks as they write notes. Figure 1 shows the final code filling the blackboard's center section at the end of this chalk-and-talk session of nine minutes, 10 seconds. This session, in which the teacher writes while providing extensive verbal accounts, is typical of most of the class sessions recorded.

Thus far, there is little that might be thought of as the *collective* writing of code that includes students as well as the teacher. Alan's next statement appears to be more of the same. He stands in front of the code he has just produced, faces the class,

Figure 2. Final code for `add` method.



```
// Javadoc
public void add (Object entry) {
  if  ( size >= array.length ) {
    resizeArray(array.length + STEPSIZE)
  }
  array[size] = entry;
  size++;
}
```

[a] The numbers in parentheses indicate the length in seconds of a pause in speech. Words inside parentheses indicate transcriber difficulty hearing the speaker, with the parenthesized words being the transcriber's best guess at what was said. Statements inside square brackets are transcriber comments.

and says, “all right (0.3) how would I create an add method (6.8) I want to add a new object to myArrayList how would I do that (4.2).” There are several pauses during this turn, and two statements that have the grammatical form of a question (“how would I”). As the pause extends at the end of this statement, it affords room for anyone to speak, and the shape of the discussion will be determined by what the actors in the setting actually do. It is not clear at this point what constitutes an appropriate response or who should provide it. For example, it may be another rhetorical question to be answered by the teacher. It is only in their unscripted and irreducibly social transactions that the teacher and the students together work out what this lesson is to be as it unfolds.

Participative code writing. A significant shift in the lesson occurs when the silence at the end of Alan’s utterance is broken. A student says, “well you’re going to want your add [Alan turns toward board] (0.4) uh method [Alan starts walking to right-hand third of the blackboard] to be public so you’re going to want some javadoc,” followed by Alan saying “right” and writing “`// javadoc`” at the top of this section of the board. Taken together as a unit, these first three turns at talk may be seen as a variant of what in educational research has become known as the “initiation-response-evaluation” pattern, or IRE.[7] In the original version as reported in the literature, the student provides an answer to a query that is then evaluated. But in the variant here, the three-turn sequence is an initiation to a social practice (here, writing code), an instance of this practice, and feedback as to how well the newcomer has done.[12] From the student perspective, these are first instances where they live and indeed partake in such cultural practices as writing code. Cultural practices and the languages that go with them have been called “language games.”[18] Ludwig Wittgenstein insists on the fact that language games may be learned before and sometimes “without ever learning or formulating rules.” Any game may be learned by participating in it and then being told a rule as part of feedback on why a move does not work. In this case, “the rule may be an aid in teaching the game.” In saying “right,” Alan makes it evident for everyone to see that the student has acted in a way that is consistent with the cultural rules of writing code, without those rules being explicitly stated in advance.

In the next turn-pair, a student says, “and (0.4) uh it’s going to need to take an argument” after which Alan writes the word “`public`” on the next line on the blackboard. This turn-pair can be seen as the response-evaluation part of the IRE pattern, with the initiation—the “I” part—implicit in the evaluation—the “E” part—of the prior IRE. But it is only with this new student response and teacher evaluation that the teacher’s evaluation in the prior IRE can be seen as initiating the response and evaluation that follow. That is, this IRE speech pattern is neither the teacher’s nor the students’ alone but is irreducibly spread across them; it is a structure of participation in this game. Only in the student response can the statement preceding and following be seen as initiation and evaluation; and likewise, only in the teacher initiation and evaluation is the student’s utterance seen as a response. This pattern of linked, cascading IREs continues through almost all of this eight-minute joint-code-writing episode, with the evaluation part including either an explicit or implicit solicitation for further student response.

In the subsequent turn sequence, the student continues his earlier utterance, saying “which I believe will be an object you could just call it add” and Alan saying “okay (0.5) so I’m going to make public [writes and speaks the following words at the same time] void add [writes ‘`(`’] and it’ll [writes ‘`Object`’ as he speaks the rest of his turn] take an object what do you want to call that object.” Alan’s writing and corresponding speech do three things at the same time: acknowledge what the student offered, provide a positive evaluation in the writing, and correct and elaborate what was said while putting it into code. The teacher, in his actual practice, writes what is “right.”

Rejection rules. Thus far in this episode, all student moves have been marked as correct, with corresponding writing on the board. The next several turn-pairs reveal what happens when student moves do not follow the rules of the unstated but present game. In the turn just finished, Alan ends with an explicit initiation in the grammatical form of a question, “what do you want to call that object.” Another student responds with “uh submit,” which is met with no writing and Alan’s comment, “submit’s a little strange, submit’s a verb (2.4) any other ideas for what we what’s a good name for this object that we’re passing in (2.4) [turns from student to others in class].” Another student says, “my object,” which again is not written on the board but is instead given an explanation for

Table 1. Classification of rules offered as reasons for rejecting student responses.

Example response-evaluation pair	Rule type
Student: submit **Alan:** submit’s a little strange submit’s a verb	Naming convention
Student: you want to try and do it without resizing first **Alan:** and if we don’t have enough room (0.3) we will do the resizing (1.1) and THEN we’ll add the thing and so that in the end when we add we will be guaranteed that we have the space	Strategic, concerning order of operations
Student: you need to check if there’s value (0.3) in the (0.5) slot nine tenth (0.4) tenth slot from the beginning **Alan:** but we don’t know if there’s 10 slots because it could have already been resized	Counterfactual or “what could be” reasoning, in that *some* possible earlier state of the program could cause a problem for the student’s proposal.
Student: creates (size) (size) **Alan:** yes but let’s not do that quite yet	Confirms the student’s response but does not write, making this rejection temporary, a “not yet but soon.”
Student: would it just be size plus one **Alan:** it would just be size … because if the size …	Logic error

the nonacceptance: "eh my object's a little bit too vague."

This code-writing session continues for another 23 turn-pairs, for seven minutes, 40 seconds total, with contributions from eight students; Figure 2 shows the resulting code. Generalizing across all of these response-evaluation pairs in the episode, accepted moves are always translated into correct code and written on the board with no explanation, while rejected moves are (in every case but one) met with explanations for their rejection and not written on the board. We summarize with this simple rule: What is right is written, what is not right is explained.

In looking at these explanations for rejection ("submit's a verb," "my object's too vague"), they can be heard as rules that have been violated in the student contribution to the game (writing code), for example, that appropriate identifiers for this parameter should not be a verb, should not be vague. Yet there are qualitatively different kinds of reasons given for rejection from one case to another. Although we do not show the entire transcript here, Table 1 lists various types of reasoning about programs the teacher makes visible in his evaluations of student responses across the entire episode.

These rules evidence Alan's expertise as a computer programmer and as a teacher. He plays his part in what Collins et al.[1] call "cognitive apprenticeship," where expertise is made visible in an appropriate context, so students, who are indeed participating in the cognitive practice, can see instances thereof. In this sense, the "rejection rules" Alan makes visible are significantly *unlike* the kind of accounting students do with one another when they pair-program, where negative criticism and explicit justification are virtually nonexistent.[9] And although various other forms of peer learning (such as teamwork, group work in a "flipped" classroom, and Peer Instruction[3]) often provide "a natural context for elaborating one's own reasoning,"[15] they may nonetheless fail to provide a context through which expert reasoning (such as these rejection rules) can be made available to students.

Toward a signature pedagogy. But it would be a mistake to see this making expert work visible as Alan's accomplishment alone, as if Alan is the sole active participant, pouring programmer wisdom into passive student containers. For in the first part of this episode, when Alan was presenting code as something already accomplished and written (Figure 1), he was writing on the board what he was explaining and explaining each statement he wrote; that is, the earlier part of this episode was governed by a different normative rule. And this earlier kind of classroom behavior—this chalk and talk—is so mundane it goes unnoticed as a pedagogical choice, replicated in countless online video tutorials for introductory programmers, with a talking head and a writing hand at the whiteboard, as well as in introductory programming textbooks and the lecture slides that accompany them as canned teacher resources. Only what is correct is given, accompanied by detailed explanation.

Table 2. Number of students writing at the end of Alan's writing or speaking turns, out of 14 students observable in the view frame of the side-facing camera.

	Number of students writing
Alan says:	
submit's a little strange submit's a verb	8
eh my object's a little bit too vague	5
and if we don't have enough room (0.3) we will do the resizing (1.1) and THEN we'll add the thing and so that in the end when we add we will be guaranteed that we have the space	5
but we don't know if there's ten slots because it could have already been resized	3
yes but let's not do that quite yet	3
it would just be size … because if the size …	5
Alan writes:	
public void add (Object entry) {	12
if (size >= array.length) {	12
resizeArray(array.length + STEPSIZE)	12
array[size] = entry;	11
size++;	12

In the presentational forms of instruction, the ways a program and writing code can go wrong are missing. However, the ways things can go wrong are even more essential to knowing a practice.[5] These ways become apparent in the second part of the episode, where students and teacher together make visible and audible two types of instances. When students' programming moves are appropriate, they are accepted without actually stating the rules, but when the moves are not part of the game, they are rejected and an associated rule is articulated. Here, students are part of playing a game, living the practice of coding; and with some practice, they will be playing the game in its solitaire variant or with other learners outside the classroom. There is a simple but easily overlooked fact about programs for why such rules of rejection might be important to know and share: The number of programs that is not correct—syntactically, semantically, or pragmatically—for any given problem is far, far greater than the number of correct programs for the problem. Students here are active, partnering in the elicitation and production of rejection rules. Alan does not simply enumerate lists of such rules based, perhaps, on past sessions with other students, to be memorized and indexed by the current students in the room. Rather, Alan responds with specific rules particularized to the responses these students are making at this moment with respect to the program at hand.

Rules of Rejection and Their Learn-Ability

Throughout this study, we have shown how the living praxis of writing programs collectively between students and teacher can reveal much more than the code that is written. Looking at the code in Figure 1 and Figure 2, there is little that distinguishes them from one another. Yet the *work* that characterizes joint code writing (Figure 2) but not teacher presentation (Figure 1) is in the student offerings of code that are rejected and the teacher rationales for the rejections. It is in that joint work that opportunities for learning computing praxis, the actual doing of it, exist. Yet in the classroom studied—we suspect in most classrooms where joint code writing occurs—no persistent traces (such as in the form of documents) are made by the teacher of this most important feature of the joint code writing. What students themselves are writing cannot be discerned from the video recordings. But what is clear is that more students are writing more of the time after the teacher writes code on the board than after the teacher provides a reason for rejecting a student offer and does no writing, as evidenced in Table 2.

In noting this key feature of joint code writing, it can more easily be made salient to new teachers of programming, so the learn-ability of these moments may be made more explicit in the teaching practice. We hyphenate learn-ability to highlight that in these moments the very ability for learning is in the visibility of the practice as it is being constituted by those present. It is easy for teachers and students alike to become fixated on the end product, the final code, making such code the sole focus of instruction, all the while losing sight of the accounts that experienced programmers can, but not necessarily will, provide for all of the wrong turns novices inevitably make as they struggle to develop a program for a given problem. In addition, to support learn-ability in this pedagogical praxis, these rules of rejection are likely to be at least as important to preserve in some persistent form as the code itself, whether as written annotations to the code, a table to summarize the joint work, or some other form. And for this task, it is possible that new technological tools can also be employed, such as the teacher using a group-editable document like a Google Docs document for writing the code, where the students, in real time and in full view of one another, provide the reasons for rejection as comments within the document.

It is easy to view code as a purely formal object and conceive of its production as a straightforward matter of applying generative rules and routines to produce correct code to specification. With such a perspective, we can study expert behavior in isolation to try to determine such rules and routines and then seek in novices those deviations and missteps that lead them to failure. Here, we have taken a different perspective, inquiring into the actual work students and teachers do together in a signature pedagogy of the programming practice.

In examining an episode of joint code writing as a social accomplishment, we find what has been hiding in plain sight but viewed as unremarkable and hence unremarked upon in the literature. The *rules for rejection* of code are as important as the generative rules and routines for producing correct code, since the space of incorrect code is vastly larger than the space of correct code. What might be viewed as failure on the part of students who offer pieces of code that are rejected can instead be seen as learnable moments, at least as productive as those offers of their code judged correct. This pedagogical form makes explicit the reasons for rejection that so many other pedagogical forms leave unexplained. In so doing, it makes the instructor's cultural knowledge explicit and visible to students at a particular point in time.

Acknowledgments

We wish to thank Natalie Jolly for reading early drafts of this article and the Helen Riaboff Whiteley Center of the Friday Harbor Laboratories of the University of Washington for providing a facilitative writing environment as we completed an early draft.

References

1. Collins, A., Seely Brown, J., and Holum, A. Cognitive apprenticeship: Making thinking visible. *American Education 15*, 3 (1991), 6–11.
2. Cooper, S., Grover, S. Guzdial, M., and Beth Simon, B. A future for computing education research. *Commun. ACM 57*, 11 (Nov. 2014), 34–36.
3. Crouch, C. and Mazur, E. Peer instruction: 10 years of experience. *American Journal of Physics 69*, 9 (2001), 970–977.
4. Dijkstra, E.W. On the cruelty of really teaching computer science. *Commun. ACM 32*, 12 (Dec. 1989), 1398–1404.
5. Garfinkel, H. *Ethnomethodology's Program: Working Out Durkheim's Aphorism.* Rowman & Littlefield, Lanham, MD, 2002.
6. Kafai, Y.B. From computational thinking to computational participation in K-12 education. *Commun. ACM 59*, 8 (Aug. 2016), 26–27.
7. Mehan, H. *Learning Lessons: Social Organization in the Classroom.* Harvard University Press, Cambridge, MA, 1979.
8. Miller, C.S. Metonymy and reference-point errors in novice programming. *Computer Science Education 24*, 2–3 (2014), 123–152.
9. Murphy, L., Fitzgerald, S., Hanks, B., and McCauley, R. Pair debugging: A transactive discourse analysis. In *Proceedings of the Sixth International Workshop on Computing Education Research*, 2010, 51–58.
10. Roth, W.-M. and Jornet, A.G. Situated cognition. *WIREs Cognitive Science 4* (2013), 463–478.
11. Roth, W.-M. and Lee, Y.-J. Vygotsky's neglected legacy: Cultural-historical activity theory. *Review of Educational Research 77*, 2 (2007), 186–232.
12. Roth, W.-M. and Radford, L. Re/thinking the zone of proximal development (symmetrically). *Mind Culture and Activity 17*, 4 (2010), 299–307.
13. Shulman, L. Signature pedagogies in the professions. *Daedalus 134*, 3 (2005), 52–59.
14. Spohrer, J.C. and Soloway, E. Novice mistakes: Are the folk wisdoms correct? In *Studying the Novice Programmer*, J.C. Spohrer and E. Soloway, Eds. Lawrence Erlbaum, Hillsdale, NJ, 1989, 401–416.
15. Teasley, S. Talking about reasoning: How important is the peer in peer collaboration? In *Perspectives on Socially Shared Cognition*, L. Resnick, J. Levine, and S. Teasley, Eds. American Psychological Association, Washington D.C., 1991, 361–384.
16. Tenenberg, J. and Maria Knobelsdorf, M. Out of our minds: A review of sociocultural cognition theory. *Computer Science Education 24*, 1 (2014), 1–24.
17. Turkle, S. and Papert, S. Epistemological pluralism and the revaluation of the concrete. In *Constructionism*, I. Harel and S. Papert, Eds. Ablex Publishing Company, Norwood, NJ, 1991.
18. Wittgenstein, L. *Philosophical Investigations / Philosophische Untersuchungen.* Blackwell, Oxford, U.K., 1997.

Josh Tenenberg (jtenenbg@uw.edu) is a professor in the Institute of Technology at the University of Washington, Tacoma, WA, USA.

Wolff-Michael Roth (mroth@uvic.ca) is Lansdowne Professor of Applied Cognitive Science in the Faculty of Education at the University of Victoria, British Columbia, Canada.

Donald Chinn (dchinn@uw.edu) is an associate professor in the Institute of Technology at the University of Washington Tacoma, WA, USA.

Alfredo Jornet (a.g.jornet@iped.uio.no) is a postdoctoral researcher in the Department of Education at the University of Oslo, Oslo, Norway.

David Socha (socha@uw.edu) is an associate professor in the School of Science, Technology, Engineering and Mathematics at the University of Washington Bothell, Bothell, WA, USA.

Skip Walter (skip.walter@fticonsulting.com) is the chief product officer of the FTI Consulting Technology Segment in Seattle, WA, USA.

DOI:10.1145/3199477

The U.S. State Department's Internet Freedom agenda is being adapted to help them communicate without DNS and IP address filtering.

BY RICHARD R. BROOKS, LU YU, YU FU, OLUWAKEMI HAMBOLU, JOHN GAYNARD, JULIE OWONO, ARCHIPPE YEPMOU, AND FELIX BLANC

Internet Freedom in West Africa:

Technical Support for Journalists and Democracy Advocates

IN DEVELOPED COUNTRIES, Internet penetration is near saturation and population growth is stagnant. In contrast, the African population is young and growing quickly. UNICEF estimates that by the end of the century, 40% of the world's population will be African.[a] Where Africa in May 2016 had 16% Internet penetration, the McKinsey Global Institute predicted

» key insights

- West African governments try to restrict access through Internet blackouts and invasive surveillance, and pro-democracy movements use the Internet to promote free and fair elections.
- Innovative technologies for avoiding detection produced by criminal botnets are being adapted by legitimate network services to increase network privacy and security.
- What we viewed as a purely technical project helped launch a grassroots movement promoting freedom of expression, transparency, and democracy in West Africa that subsequently also influenced the social, economic, and political frameworks there.

[a] https://www.unicef.org/publications/files/UNICEF_Africa_Generation_2030_en_11Aug.pdf

Opposition supporters protest at the Place de la Nation in Burkina Faso's capital Ouagadougou, November 2, 2014.

IMAGE BY ISSOUF SANOGO/AFP/GETTY IMAGES

in 2013 that by 2025 the penetration rate will be approximately 50%[b] and that 600 million Africans will be using the Internet,[c] producing approximately $75 billion in annual e-commerce activity and contributing $300 billion to African GDP.

West Africa is a diverse region in sub-Saharan Africa, including both the Sahel desert and lush rain forests. Many local languages from distinct language groups are spoken, along with the former colonial languages, including French, Spanish, Portuguese, and English. The region includes thriving democracies like Ghana (with press freedom ranked by Reporters Without Borders better than France and the U.K.) and repressive regimes like Equatorial Guinea (with press freedom ranked by Reporters Without Borders at the level of Cuba, Eritrea, Iran, and North Korea). The majority of the West African population lives in countries that do not allow effective freedom of expression.[d,e]

This article discusses Internet freedom in West Africa. In April 2016, we completed a project sponsored by the U. S. State Department's Bureau of Democracy, Human Rights, and Labor, whose goal was to promote online freedom of expression by West African activists. To this end, we implemented a distributed proxy network and held

[b] https://www.mckinsey.com/industries/high-tech/our-insights/lions-go-digital-the-internets-transformative-potential-in-africa

[c] According to http://www.internetworldstats.com/stats.htm, there are approximately 320 million Internet users in North America and approximately 630 million Internet users in Europe.

[d] https://freedomhouse.org/sites/default/files/FH_FTOP_2016Report_Final_04232016.pdf

[e] https://rsf.org/en/ranking

annual training sessions. Our proxy counters Internet censorship and surveillance, following a design loosely modeled on criminal botnets.

Our training sessions provided participants the skills they needed to protect their freedom of expression, bringing together a regional community of bloggers, technologists, journalists, and democracy advocates. Trainees from multiple countries found they were facing similar problems.

Internet Freedom

Though freedom of expression is guaranteed by Article 19 of the United Nations Universal Declaration of Human Rights (http://www.un.org/en/universal-declaration-human-rights/index.html), the world's ability to access information and openly express opinions is tenuous and unevenly distributed. The non-governmental organization (NGO) Freedom House's 2016 report on press freedom said that global press freedom was at its lowest point in 12 years.[f]

Non-democratic countries with poor human rights records maintain power by tightly controlling information, limiting their populations' ability to share opinions, organize, and create democratic alternatives. For these governments, the traditional press is easier to control than the Internet. They can physically intervene in print and broadcast media operations. Newspaper distribution networks are expensive to maintain and easily disrupted. In almost every country, radio and television broadcasters are controlled, or licensed, by the government.

In contrast to traditional media, it is inexpensive and less risky to put a web server online in another country. In countries where the population does not trust traditional media, social media has emerged as an alternative. As more voices become available to the population, fearing a loss of control, repressive governments invest in technologies for shriveling and censoring Internet traffic. Firewalls can block the domain name system (DNS) and/or Internet Protocol (IP) addresses of offending news sites (such as *The New York Times*). Internet surveillance tools using deep packet inspection (DPI) can block network sessions containing sensitive keywords. DPI tools are often considered "dual-use," along with legitimate network management, making it difficult to regulate the export of these technologies.

Repressive governments subject dissenting voices to denial-of-service (DoS) attacks that are inexpensive,[g] difficult to attribute, and make objectionable viewpoints unavailable. Governments hire cheap, unskilled laborers to flood websites with either pro-government "50-cent army" or abusive troll comments. If all else fails, a government can simply shut off the Internet and other telecommunications technologies during politically sensitive times. In 2017, the governments of Cameroon, the Democratic Republic of Congo, and Gabon all used Internet blackouts as a political tool.

To protect the freedom of expression guaranteed in Article 19, NGOs and Western governments try to foster Internet freedom. Technical tools, like The Onion Router, or Tor,[h] Psiphon,[i] uProxy,[j] and Lantern,[k] provide proxy services for evading national firewalls. Trainers teach at-risk populations to use the Internet securely, avoid surveillance, and circumvent censorship. NGOs lobby governments and international groups to put in place laws and policies to safeguard the public's freedom of expression. Our project promoted Internet freedom within West Africa, producing a censorship-circumvention tool and building a West African user community. This article documents our experiences.

West African press freedom. As shown in the table here,[l] freedom of expression in Africa is being confronted by special challenges. Approximately 40% of the countries in sub-Saharan Africa (over 38% of the population) live in countries whose press freedom

West African freedom surveys populations from the *CIA World Factbook*, 2016 (https://www.cia.gov/library/publications/the-world-factbook/); RSF columns from Reporters Sans Frontières, 2010 and 2017 (https://rsf.org/en/ranking); and FH FIW columns from Freedom House Freedom in the World, 2010 and 2017 (https://freedomhouse.org/report-types/freedom-world).

Country	Population	2010 RSF	2017 RSF	2010 FH FIW	2017 FH FIW	Users
Benin	10,741,458	70	78	Free	Partly Free	8
Burkina Faso	19,512,533	49	42	Partly Free	Partly Free	5
Camer.	24,360,803	129	130	Not free	Not Free	8
Chad	11,852,462	112	121	Not Free	Not Free	6
Cote d'Ivoire	23,740,424	118	81	Not Free	Partly Free	39
Congo (Kinshasa)	81,331,050	148	154	Not Free	Not Free	2
Djibouti	846,687	110	172	Partly Free	Not Free	3
Equat. Guinea	759,451	167	171	Not Free	Not Free	1
Gabon	1,738,541	107	108	Not Free	Not Free	1
Gambia	2,009,648	125	143	Partly Free	Not Free	15
Guinea	12,093,349	113	101	Not Free	Not Free	4
Liberia	4,299,994	84	94	Partly Free	Partly Free	2
Mali	17,467,108	26	116	Free	Partly Free	1
Niger	18,638,600	104	61	Partly Free	Partly Free	1
Nigeria	186,053,386	146	122	Partly Free	Partly Free	4
Sierra Leone	6,018,888	91	85	Partly Free	Partly Free	3
Senegal	14,320,055	93	58	Partly free	Partly Free	7
Togo	7,756,937	60	86	Partly Free	Partly Free	12

RSF = Reporters Without Borders
FH FIW = Freedom House Freedom In the World

[f] https://freedomhouse.org/sites/default/files/FH_FTOP_2016Report_Final_04232016.pdf

[g] Reports (2016) suggest distributed DoS (DDoS) attacks can be ordered online for as little $5 per hour; https://www.incapsula.com/blog/unmasking-ddos-for-hire-fiverr.html

[h] https://www.torproject.org/

[i] https://psiphon3.com/en/index.html

[j] https://www.uproxy.org/

[k] https://getlantern.org/

[l] Population statistics from *The CIA World Factbook* (https://www.cia.gov/library/publications/the-world-factbook/); RSF rankings from Reporters Without Borders, 2010 and 2017, with 1 as best (such as Norway) and 180 as worst (such as North Korea); and qualitative rankings from Freedom House Freedom in the World, 2010 and 2017 (https://freedomhouse.org/report-types/freedom-world).

is ranked by Freedom House as "not free," and more than 61% of the population lives in countries ranked as "partly free." The West African countries with the lowest press-freedom rankings are The Gambia and Equatorial Guinea.[m] The Gambia is a small, English-speaking country surrounded by larger, French-speaking Senegal. Many Gambian journalists have lived in exile to avoid persecution. Its 2016 election, which removed the former strongman, may yet change its ranking. Equatorial Guinea is a small, Spanish-speaking country between Cameroon and Gabon, with large oil reserves producing revenues that go mainly to the ruling family. Trainees told us that Equatorial Guinea blocks access to social media.

Freedom House gave both countries the lowest possible ranking for political rights in 2016. Independent of our work, a pan-African group augmented Article 19 by drafting a comprehensive African Declaration on Internet Rights and Freedoms[n] to put in place African norms in support of online freedom of expression.

Other countries in the region have relatively positive rankings for their press freedom. We had participants from Benin, which is ranked as "partly free," with one of the better rankings in West Africa (Reporters Sans Frontières ranks Benin next to Italy). Training participants from Benin still feared legal proceedings meant to intimidate political speech. During our training sessions, 2012 to 2016, the status in some countries improved. Côte D'Ivoire (Ivory Coast) moved from "not free" to "partly free," and Senegal's rankings improved. The only available statistics for Internet Freedom in Africa can be found in Freedom House's series of Internet Freedom Reports,[o] which increased the number of African countries it covers from six in 2011 to 16 in 2016. Unfortunately, only two West African countries—The Gambia and Nigeria—were included, thus limiting our ability to provide objective demographics here.

Activist community. We worked with bloggers, journalists, and activists. Bloggers use multiple platforms to express opinions and inform local populations about political, economic, and ecological developments. Journalists were dedicated to their profession in situations with limited monetary reward and real physical danger. Some of those we trained were international correspondents working in West Africa for international broadcasters, including representatives of the Committee for the Protection of Journalists, International Federation of Journalists, and local journalist unions. The "users" column in the table lists the number of participants from each country,[p] not including international correspondents.

Our activist community included human rights and technical activists. The region has a growing open source and maker community that is politically engaged, promoting economic, social, intellectual, and political democratic development in the region. Approximately 15% of our participants were female, and approximately 20% were primarily technical activists; the remaining 80% were about evenly split between bloggers and professional journalists. These numbers are inexact in part because participants were not easily categorized, and some had their own businesses providing both content and technical services.

Internet influence. All participants used the Internet, including social-media platforms, giving them a strong voice. The government of The Gambia recognized the power of the Internet in 2013 by passing a law that punishes its use to "spread dissatisfaction with the government" with fines over $100,000 and 15 years in jail.[q] The influential Balai Citoyen[r] (Burkina Faso) and Y'en a Marre[s] (Senegal) movements had used the combined influence of musicians and web activists to bring about free and fair democratic elections since 2011. Repressive governments and free-expression activists alike were aware of the power of the Internet and new media, using them to advance their agendas.

Internet censorship and attacks on free speech in West African countries have not attracted as much attention as censorship in countries like China and Iran. Only limited information has been available regarding network surveillance and censorship in West Africa, let alone use of censorship-circumvention tools there. Although our project was not intended to collect statistics, we learned the reality of the situation from the trainees. We generated reports that circulated among the human rights community. The fact that Freedom House increased the number of sub-Saharan countries in its *Freedom on the Net* reports[t] since 2010 indicates increased awareness by the international community of the situation in the region.

Our Project

Clemson University and Syre Inc. designed our project to adapt the U.S. State Department's Internet Freedom agenda to the needs of West Africa. We recruited the NGO Internet Without Borders (Internet Sans Frontières) as a liaison with the human rights community in sub-Saharan Africa. The project had two main objectives: develop secure messaging tools based on author Brooks's research at Clemson University tailored to local needs; and provide bilingual (English and French) training for the West African user community.

Proxy networks. Tools are available for circumventing censorship, many providing proxy connections to Internet users. A local client initiates a connection to a remote server through an encrypted "tunnel," and the remote computer executes actions requested by the local host, returning results to the local host through the tunnel. We now discuss the Tor, Psiphon, Lantern, and uProxy proxy systems. Proxy networks and virtual private networks (VPNs) help users circumvent surveillance and censorship but are not perfect solutions:

National firewall. A national firewall can track remote connections, detect DNS/IP addresses used by proxies, and block suspect addresses. Censors use DPI to identify addresses with suspect content;

[m] https://freedomhouse.org/sites/default/files/FH_FTOP_2016Report_Final_04232016.pdf

[n] http://africaninternetrights.org

[o] https://freedomhouse.org/report-types/freedom-net

[p] See the table for demographic and human rights data on the countries discussed here.

[q] https://freedomhouse.org/report/freedom-net/freedom-net-2015

[r] English translation: "A citizen's broom sweeps clean."

[s] English translation: "We are fed up."

[t] https://freedomhouse.org/report/freedom-net/freedom-net-2015

Proxy connections. Proxy connections can be security risks at both ends. Clients can have sessions spied on by the proxy server. Servers can be made responsible for client actions that seem to be from the local machine; and

Latency and jitter. Increased latency and jitter hinder user acceptance.[4] Even users aware of local censorship and surveillance risks tend to use faster direct Internet connections.

Each proxy has its own strategy for overcoming these drawbacks. Psiphon[u] is a one-hop proxy with multiple modes, including one that hides its use of encryption. It avoids nation-state firewall blocking by running a large, international park of proxy nodes that are difficult to enumerate and providing access options that obfuscate the connection. The proxy nodes are provided by Psiphon, making Psiphon potentially liable for criminal abuse. (An online system allows Chinese users to rank the speed and stability of existing proxy solutions, https://cc.greatfire.org/en; from here on, we include, in parentheses, its ranking of each proxy as of June 2017, if the proxy was present in the survey.) For example, Psiphon (11) users have to trust Psiphon not to spy on proxy connections and exploit session information.

The Tor[v] (9) proxy network tunnels connections through three, separately encrypted hops. To protect user privacy, entry into the Tor network is normally through a small number of trusted guard nodes. Tor provides advice to help exit-node providers minimize their legal risk, primarily by telling ISPs in advance that the node is a Tor exit node. Exit nodes have been used to spy on users in the past. It is unwise to send personally identifiable information through Tor. Tor's use of two additional network connections to increase anonymity adds more latency and jitter than one-hop proxies. Many countries block Tor. Iran has blocked SSL/TLS connections, which block Tor. China blocks connections to IP addresses that run Tor. China looks for the TLS cipher lists that indicate Tor use. Some countries actively probe and blacklist nodes they suspect of providing access to Tor. Tor counters such blocks by maintaining a set of reserve (bridge) addresses that become available as needed, though this set of nodes sometimes is scarce. Tor is also implementing pluggable transport[w] (PT) layers that modify the network transport layer and disguise Tor traffic. Unfortunately, each PT is usually supported by only a small number of bridges; a PT can also produce a fingerprint that can be detected.

Non-democratic countries with poor human rights records maintain power by tightly controlling information, limiting their populations' ability to share opinions, organize, and create democratic alternatives.

uProxy[x] is a browser extension for Chrome and Firefox that allows users to share their Internet connection with others. It was developed by Google Ideas, now the Jigsaw subsidiary of Alphabet, in conjunction with Lantern and the University of Washington. uProxy functionality is roughly similar to CGI-Proxy we used in our system. Ideally, a friend of a user in a repressive country would volunteer to provide a friend with a proxy connection. In this scenario, the friend risks being potentially responsible for illegal activities done by the proxy client, and the client could be spied on by the friend. Alternatively, the proxy connection can be through a commercial provider. This second scenario is basically equivalent to using a commercial VPN. As with Psiphon, the user has to trust the commercial VPN. Many users do not have friends available in countries outside the firewall, and countries with national firewalls often block the service providers (such as Github and Gmail) uProxy relies on.

Lantern[y] (6) is a product of Brave New Software, a nonprofit providing a distributed proxy, bootstrapping initial connections through Google Talk servers, but does not provide anonymity, aiming instead to provide efficient access to websites. If a site is not blocked locally, Lantern will load the material directly and not use the proxy. If a webpage is blocked locally, Lantern will retrieve the webpage through a proxy connection. Lantern maintains a distributed set of proxies for the user. Users can allow their connection to be shared. All traffic passing through the Lantern peer-to-peer system is encrypted. The distributed na-

[u] https://psiphon3.com/en/index.html
[v] https://www.torproject.org/
[w] https://www.torproject.org/ docs/pluggable-transports
[x] https://www.uproxy.org/
[y] https://getlantern.org/

ture of Lantern reduces, but does not eliminate, the risks of proxy use. Since only part of the session would be sent through an individual proxy exit node, the likelihood that an exit node would be blamed for the acts of a malicious user are reduced. Similarly, the amount of information an exit node can harvest from a naive user is reduced.

Our proxy design. We developed and deployed a network of peer-to-peer proxies for our user community that included journalists, human rights activists, political dissidents, and technology activists from the region. Our technical goal was to adapt tools used in the botnet community to avoid DNS and IP-address filtering. Many botnets remain active for years despite our best efforts to stop them.

Unlike Tor, Psiphon, and Lantern, which are open to the public, our tool resembles uProxy in that it is deployed by a small, trusted, authorized user community. It is similar to Lantern in that our clearinghouse maintains a dynamic list of proxies available for immediate use. Unlike other proxies, we vetted the people invited to our training sessions, and they helped define the rules we enforce in maintaining the network. To reduce the risk of using a proxy, we did the following:

Informed users of risks. We explained the risks involved in being a proxy server, with users allowed to opt out of being proxies for others;

Established a trustworthy user community. We provided the system to a small set of users, all individually vetted by our partners. Most were professional journalists, well-known bloggers, and/or human rights activists. All had strong professional credentials;

Protected privacy. We limited access to the network to only authenticated users and kept no records of user sessions;

Adopted community-defined standards. We enlisted users in defining the code of conduct to be respected by the user community;

Recognized political boundaries. We maintained a matrix segregating countries by security agreements and shared infrastructure, with proxy nodes chosen only from countries not friendly with the local users' governments; and

Created a sense of community. We had the user community meet at training sessions, with individual users deciding whether or not to be a proxy server after talking face-to-face with potential proxy clients.

Our users made fully informed decisions as to whether or not to share their network connection. The risk of acting as a proxy in this setting is less than with Tor and similar to sharing a network connection with a colleague through uProxy. It is difficult to compare this risk with the risk of being an exit node for Lantern, where users provide small slices of their bandwidth to strangers. With our system, users provide a vetted colleague with an entire session. We are the only proxy we are aware of that automatically blocks the use of proxy servers when the political stance of the exit server's country could pose a risk.

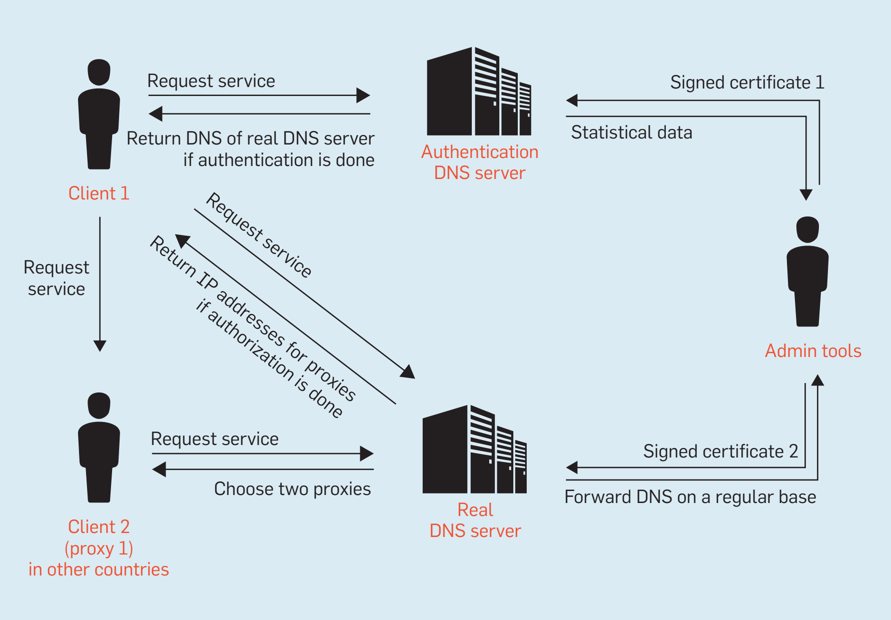
We did not conduct extensive performance comparisons between our tool and the other proxy networks. It is probably fair to assume that the connection speed and jitter of the one-hop proxies, including ours, are roughly equivalent. The observed network throughput of the connection is one factor we considered in choosing the proxy location. Worth noting is that the performance of our tool in Africa was quite different from when we tested it in North America and Europe. The Internet in Africa uses less wired infrastructure, and a number of 4G wireless providers compete for business in the urban centers.

To access our network, clients use the network protocol in Figure 1 to find the address of a remote proxy. They have a dynamically updated list of DNS names used to connect to our proxy network. It attempts to open a Secure Shell (ssh) session through a DNS tunnel to our authentication node. Password-less ssh credentials verify that the connection is from one of our authenticated users. When authentication succeeds, the local node receives the DNS name of a proxy clearinghouse node.

The local node opens a second DNS tunnel to the clearinghouse and uses Secure Copy through the DNS tunnel to retrieve the IP address of the node/proxy it can use to access the Internet for the current session. Our protocol design and implementation included a number of innovations adapted from criminal botnets to counter Internet censorship, especially DNS/IP filtering. The following sections describe the techniques we used to avoid tracking and detection.

To establish secure communication to our system's authentication servers, we had to bypass firewalls and network filters and found that many malicious botnets use DNS to communicate covertly.[1] DNS is of interest for several reasons: it is globally deployed and used; its filtering typically blocks attempts to connect to a blacklisted set of sites; and its

Figure 1. Nodes find their remote proxy partner using DNS tunneling to access a proxy clearinghouse hidden by a fast flux connection.


Request service
Return DNS of real DNS server if authentication is done
Authentication DNS server
Signed certificate 1
Statistical data
Client 1
Request service
Return IP addresses for proxies if authorization is done
Request service
Admin tools
Request service
Choose two proxies
Real DNS server
Signed certificate 2
Forward DNS on a regular base
Client 2 (proxy 1) in other countries

packets and records are rarely validated by the ISP, allowing DNS servers to be impersonated. These factors make DNS suited for use as a covert communication channel.

Besides DNS tunneling, we also adapted "fast flux" ideas created by botnets to protect our users. The term fast flux refers to frequent redefinition of the IP addresses affiliated with a DNS name. In current botnets, one symbolic DNS name is affiliated with a large number of IP addresses. The IP addresses are given "short time to live" values and swapped out frequently, generally less than three minutes. The result is a DNS name that cannot be reliably tied to any computer through its IP address (see Figure 2).

Nodes in the fast flux tend to work as proxies for a "mothership" that wants to be hidden, effectively avoiding detection and tracking by providing a moving target. This is largely why botnets have been so difficult to stop, even when law enforcement and tech vendors might conspire to track down and neutralize them.[5] Our approach applied this concept to our authentication and proxy-distribution servers to add an additional layer of redundancy and survivability. The proxy-distribution server determined what proxy would be used by the client. As with botnet fast flux, the servers moved frequently to different physical and logical locations on the Global Environment for Network Innovations (GENI) network.[z]

We also regularly changed the server's domain names. In practice, we chose them from Latin-alphabet-language words taken from Wikipedia. One alternative to this approach would be to algorithmically generate domain names[3] using an algorithm like the one in Fu et al.[2] To further obscure the network, we used dynamic DNS services to register our domain names, allowing individuals to register, at no cost, subdomains to any of a large set of volunteer root domains. This is useful, since the root domains are quite varied and have no direct connection to our project.

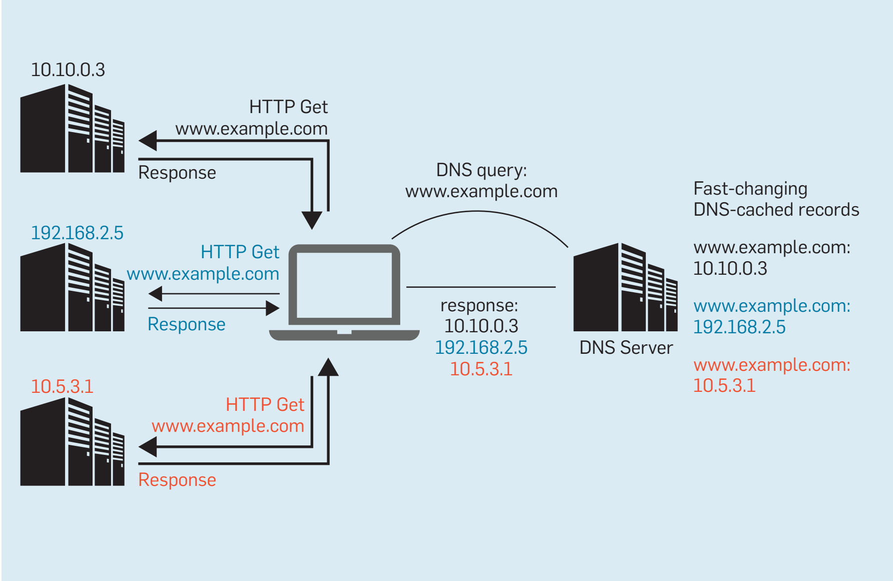
Criminal botnets are known to have been "sinkholed," or a law-enforcement agency anticipates the domain name that will be used and then registers that name. This allows law enforcement to identify and isolate the infected nodes, effectively dismantling the botnet. We used the following strategies to avoid sinkholing:

Refreshed DNS names. Each node regularly received, during its session, a list of DNS names the authentication server would use in the future when the current DNS name would no longer be available.

Hidden Tor service. We maintained a Tor hidden service with a user forum. Should users be disconnected from the service, we would provide a script on the forum that would provide the system with the current list of DNS names.

In practice, we had no difficulty with sinkholing and never had to use the second approach.

Figure 2. Fast flux.



Hardened environment. We gave users a hardened networking environment consisting of a bootable, encrypted Linux USB drive (Linux Mint-aa[aa]), a set of scripts that create and remove a temporary environment on Windows (encrypted using 7-Zip[ab]), or an Android app. When using the first two, no software would be installed on the user's machine, and care would be taken to avoid leaving data traces for later forensic analysis. Users had to safeguard only the encrypted USB drive. Unlike the proxy tools discussed earlier, we did not assume the user computer could be kept secure from local authorities.

The hardened environment would automatically launch a browser using CGIProxy[ac] to access a remote proxy node launched in the hardened environment. CGI-proxy connections all use TLS, thus limiting DPI's ability to identify suspect communications.

Client hardening encrypted the work environment so even if the software would be lost or fall into the wrong hands, the risk of a user data breach and disclosure of the proxies would be greatly reduced. A strong password was required for users to extract data and use our tool.

Should hostile users acquire a copy of our tool, the amount of damage they could inflict on the system would be limited. Our user community became rather close, and it is quite likely we would have been informed of arrest, detention, or physical threats, in which case we would have disabled the user's access to the system. Even if we were unaware of a user's compromise, our matrix of political alliances and shared infrastructure would have guaranteed a

[z] https://www.geni.net/#

[aa] Linux Mint provides full disk encryption to counter viruses and keyloggers on users' laptop computers; https://www.linuxmint.com/

[ab] 7-Zip is portable software for compressing or zipping files secured with encryption; http://www.7-zip.org/

[ac] CGI-proxy is a tool comparable to uProxy that lets nodes without web servers function as a proxy for others; https://www.jmarshall.com/tools/cgiproxy/

compromised node could connect only to nodes of no possible interest to local authorities. Any attempt to enumerate network addresses providing proxy access would have never provided information of interest to a local regime. To the best of our knowledge, this is a unique feature of our approach.

Peer-to-peer proxies. The list of proxy nodes we maintain includes only those that are currently active. In addition to proxy servers in Africa, we maintained at least four active proxy servers on the GENI network for approximately 50 users. The clearinghouse kept up-to-date information on the quality of the network connections to proxy nodes that let us do load balancing. Members of our user community also acted as proxies for each other, as shown in Figure 1.

We gave each user the choice of whether or not to act as a proxy server. This option had not been anticipated at the beginning of the project but was requested by users during training. We originally assumed solidarity with the community would lead it to provide secure connections for each other. Many expressed concern over how authorities could misuse information harvested through eavesdropping on proxy sessions on their nodes.

Internet Without Borders (http://internetsansfrontieres.org/) developed a matrix identifying countries with either mutual defense agreements or the same telecommunications providers. Proxy connections are made only through countries that are not friendly with local government and various providers. We did this to protect our user community. Proxy connections were encrypted while passing from the client through the network to the proxy but in clear text when leaving the proxy. By forcing connections through a country not aligned with the home country, it becomes functionally impossible for the home country's political authorities to survey the session. Should a node become compromised and used to harvest the addresses of our users, the home country authorities would be able to harvest only the IP addresses of users in countries with which they *do not* have friendly relations. The user community grew close and was often aware when a member was detained, thus allowing us to remove that user's credentials from the authorization node.

Repressive governments and free-expression activists alike were aware of the power of the Internet and new media, using them to advance their agendas.

Lessons learned. During deployment, we learned a number of important lessons:

Qualitatively different. The Internet in Africa is qualitatively different from the Internet in Europe and the U.S. Wired connections are rare and power disruptions are common. Most connections use 4G wireless in urban areas;

Test as soon as possible. Start testing the tool in the local environment as soon as possible. Our first version, which we tested in Europe and the U.S., delivered extremely poor quality of service on African networks;

Enlist local technologists. Enlist local technologists into the project for testing early in the process if possible. Once we began using colleagues in Abidjan and Abuja to test our system, we were able to find timing errors more quickly;

Use local technologists. Use local technologists to support other users. Many web activists who were part of co-working spaces made themselves available on short notice to help journalists;

Listen to the local participants. Listen to the local participants to learn their security problems, many of which we could not have anticipated. We had assumed users would be eager to serve as proxies for their colleagues, but many were, in fact, hesitant. This is reasonable for people living under authoritarian regimes, and we were naive not to have foreseen it;

Apply local lessons. We adapted war games from African training and used it as a class exercise for our college students who found ways to evade our surveillance we had not anticipated. We added those tools to the next set of training sessions; and

Make no assumptions. Do not make assumptions about the local security situation based on rules of Western law enforcement. Warrantless searches occur. Some users had problems with informers within their own local networks. The political situation can be changed for the better by the local population when it has access to information. Participants in our training were in groups (rap singers and web activists working together) that even managed to bring about regime change, removing entrenched governments from power.

Fadel Barro (2-L), a leader of Le Y'en a Marre (We're Fed Up) movement, and Oscibi Johann (2-R), a leader of Burkina Fasos Le Balai Citoyen (Citizens Broom), at a press conference in Kinshasa.

PHOTO BY FEDERICO SCOPPA/AFP/GETTY IMAGES

Proxy system comparison. Such systems vary according to how proxy nodes are chosen. Tor and Lantern users rely on public proxy nodes, though public proxy nodes have been used to spy on users. Psiphon runs its own proxies and has access to incoming and outgoing traffic. uProxy forces users to find their own proxy nodes. We were able to provide users with a community of professionally vetted colleagues they could meet face to face.

Proxies also route traffic differently. Psiphon connects users directly with nodes located mainly in Western countries. When not using the connection of a friend, uProxy uses cloud connections through Digital Ocean (sites in North America, Europe, Bangalore, and Singapore), Facebook, Github, or Google. Such a connection can be difficult to access from countries with active censorship; for example, many are blocked in China. Tor routes are chosen from nodes distributed throughout the world. Users can specify preferred nodes (and therefore countries) for entry and exit. Lantern's routing assumes individuals are *not* being targeted. Proxy routes include nodes in the local country. We assumed our users *were* being targeted. We routed traffic to proxy nodes located at either U.S. research universities or in a West African country not allied with a local government.

To the best of our knowledge, ours is the only proxy that explicitly considers political tensions in choosing how to route proxy traffic. The way Tor, Psiphon, Lantern, and uProxy maintain direct connections to proxy nodes has made them vulnerable to traffic fingerprinting, blacklisting, and active probing. Our use of fast flux was different from these existing tools. By frequently changing the DNS and IP addresses associated with our proxy, it should have been more difficult to use such techniques to disable our system. However, traffic fingerprinting would still be possible for identifying our use of DNS tunneling, and our use of DNS tunneling required very few, small messages. To date, this has not been a problem.

Training sessions. We sought out local activists who were most capable of contributing to our training sessions. We posted advertisements, used social networks, and took advantage of our connections within the African diaspora to recruit a diverse set of participants. Our plan was to have two sessions each year from 2014 to 2016, one in Abidjan, Côte D'Ivoire, and one in Paris, France. We chose Abidjan, since it is a major commercial center for West Africa with excellent travel connections. As Abidjan is a member of the Economic Community of West Africa, citizens of almost all countries in our target area did not need a visa to travel there. It also has a stable political climate. Training in Abidjan was held at a computer training facility at Université Felix Houphouet Boigny.

As the former colonial power, France still has strong cultural and economic ties to most West African countries. There is also a large African diaspora in greater Paris, and it is not unusual for African dissidents to come to France as political refugees. For the third year, we held two training sessions in Abidjan but dropped the one in Paris. During the first two years, we had already educated the members of the diaspora in France who were most influential and were having difficulty getting visas approved to travel to France to participate.

Participants. We received a large number of applicants who recognized the importance of secure Internet use for their own projects. Although most were natives of West Africa, a few were also from Europe. The Europeans worked for NGOs involved in the region or for international organizations or were journalists working for international broadcasters. The training groups included members of the International Federation of Journalists, the Committee for the Protection of Journalists, and several NGOs that preferred to not be identified. Most of the African participants were either journalists or bloggers, many also influential activists.

One participant had set up an online election-monitoring system that was largely responsible for his country's first peaceful democratic transition of power. Another worked with a group of rap musicians and tech activists who had mobilized their local populations to protest a planned change in the local constitution that would have let the local strongman remain in power for more than 27 years. Enough protesters took part to convince the country's army to ask the strongman to leave the country, leading to a free and fair election.

Participants not able to take part in the training included a blogger from Mali who continued reporting from his city even while it was occupied by Al Qaeda and a journalist working in the Central African Republic during a violent civil conflict between Christians and Muslims.

Participants reported a number of threats to Internet freedom in the region:

In Gambia. Journalists would be held by the National Intelligence Agency until they allowed access to their email messages;

In Togo. Journalists worried that communications networks would be shut down during elections and journalists detained following sensitive mobile phone conversations;

In the public interest. Some activists were jailed for putting online apolitical information that was clearly in the public interest; and

Forced to flee. Following training, at least three activists were forced to flee their country of origin due to threats of imprisonment or physical harm due to their online presence. Other participants helped them find safe haven in other countries.

There is a very active maker community in West Africa, including a number of free-software activists. The local tech community is socially engaged, creating maker spaces that promote technical literacy within the region. By bringing these local technicians into our training sessions, we were able to provide the democracy advocates local contacts who could provide them technical support as needed.

Curriculum. The training curriculum concentrated on Internet freedom. We surveyed the global situation, discussed surveillance and censorship technologies, and taught the user community the necessary skills. In addition to teaching them to use our proxy, we tutored them in the use of Tor, Psiphon, and encrypted email.

In the second and third years, we added new topics and deployed a Friendica open source social-network site at a .onion address on the dark web. We found that providing the community a private, secure forum it could reach only through Tor helped its members understand the tool would give them access to items unavailable through normal means. Previously, the students had noticed Tor's latency more than its strong points. Once they were used to using Tor for communicating within the community, such communication became a habit. We found that people teaching the use of privacy tools should introduce them in ways that emphasize their unique abilities. Otherwise, students would be more likely to notice some deficiency of the user interface, like, say, latency.

War (role-playing) game. We developed role-playing game scenarios where trainees would have to cooperate and share information to win. A detailed introduction of the game is available online,[ad] and we used the game as a final exam for the training. The training personnel acted as the "national intelligence agency" that would block access to Internet sites and sniff the network for "evidence." Concrete evidence of users accessing "politically sensitive" information would cause them to be "imprisoned," or expelled from the game.

The trainees were divided into groups, with players in the same group working together. Each player would choose a role in the game by picking a piece of paper from a hat. Each team included an "agent provocateur" who would inform the "national intelligence agency" of suspicious activity. The first scenario involved reporters trying to collect information about corrupt officials and the second an armed insurgency resembling Boko Haram.

The game scenarios required trainees to apply the tools they had been given without help from the instructors. They indeed had to prove their ability to outwit them. We found this to be very useful, as it was popular with the trainees, allowing them to gain confidence in their ability to use the tools. And embedding "informers" in the scenarios was an essential aspect of the game, forcing users to think about the security of their internal communications and seriously contemplate possible threat models.

After using these scenarios for instructing journalists, the author Brooks integrated them into his computer-engineering security course at Clemson University. In addition to it being a useful exercise for the course, his students managed to find some tools (notably anonymous chat services) for use in the war game he had not previously considered. The insights he gained from his students became part of the following year's security seminar in Abidjan.

Training surveys. An anonymous survey (bilingual in English and French) was performed at the end of each training session. Participant satisfaction with the training scored an average of 4.4 out of 5 on our five-point scale.[ae] Participants rated both the appropriateness and effectiveness of the project at 4.6. They disagreed strongly (1.4) with the idea that Internet freedom is not a problem in West Africa. The main complaint was the fact that the training facilities were not adequate (3.69 out of 5), which is understandable given the limited funds, time, personnel, or a combination of reasons. The most frequent complaint about the training facilities was the quality of the local Internet connection. We received the following user suggestions for improvement: provide information on mobile phone security; provide information on telephone wiretaps; provide guidance on how to collect information on state surveillance infrastructure; and provide guidance on how to work around Internet blackouts.

Africtivistes Movement

During the final year of the course, many trainees, who had initially met at our training sessions, worked together to create the Africtivistes movement[af] and League of African Bloggers and Cyber-Activists for Democracy, which held its first annual meeting in Dakar, Senegal, November 25, 2015. The initial team was lead by Cheikh Fall (@cypher007), who helped *Y'en a Marre* put in place an election-monitoring tool in Senegal, Justin Yarga (@y_jus), who worked as a web liaison for Balai Citoyen as it led pro-democracy protests in Burkina Faso, and Aisha Dabo (@mashanubian), a Gambian journalist. They assembled 150 activists from 35 countries representing the major online movements in sub-Saharan Africa. Attendees included Youssou N'Dour, a major world-music star who was a former Minister of Culture in Senegal and current minister-adviser to the President of Senegal. We provided onsite security training for the Africtiviste delegates and helped them set up their own dark web forum.

The Africtivistes movement is coordinating national pro-democracy activities into a pan-African force. We were able to help many of them simply by giving access to other people facing similar problems. The Africtivistes group is currently working to help a number of national actions, including:

#Sassoufit. Trying to convince the President of Congo Brazzaville to respect his country's constitution and enforce the term limit of 30 years on the current president;

#Article59 Togo. Trying to convince the government of Togo to respect

[ad] https://clemson.box.com/s/4knwbrq4j27zn0w4iig2at0wanymza4t

[ae] All questions used a five-point scale ranging from strong disagreement (1) to strong agreement (5);

[af] http://www.africtivistes.org/!/

term limits, as written, in its national constitution;

Benin Vote 2016. Trying to establish an online election-monitoring system to make the country's presidential election more transparent;

#StopBokoHaram. Protesting expansion of the Boko Haram insurrection into Cameroon and persuade the international community to intervene; and

#Mauritanie. Protesting actions taken to imprison human rights activists arrested for working against the modern slave trade in Mauritania.

The Africtivistes movement is ongoing and has brought together the sub-Saharan human rights, blogging, journalism, and tech communities into a common front. Plans are underway to create a new generation of security training sessions where local trainees will train yet others to use the tools we would provide. Remote technical support will be provided by author Brooks's team at Clemson.

Conclusion

Our project ended in the spring of 2016, with its technical products having been taken over by the Internet Without Borders NGO that had expressed interest in deploying our tool for other user communities. Our training sessions were successful in many ways, some we could not have foreseen:

Influential activists. A large number of influential activists in the region are now aware of the larger international struggle for Internet freedom;

Trainees. A number of trainees used our materials and tools to hold their own local training sessions to spread their new knowledge;

Local participants. We trained local participants on a range of tools, including ours, for using the Internet securely while avoiding censorship and surveillance; and

Like-minded colleagues. Many participants connected with like-minded colleagues throughout the region with whom they could collaborate.

Plans are under way to expand this work by having Internet Without Borders deploy our technology to support other user groups and Africtivistes creating a new set of training sessions derived from our original curriculum.

Before working with local activists, we were unable to find reliable documentation as to Internet censorship and surveillance in the region. Since then, numerous reports have indicated many of the more authoritarian countries in the region have purchased and deployed sophisticated network-surveillance tools from companies in Western democracies. We have found no documentation on Chinese involvement in Internet surveillance in the region. On the contrary, China is investing in local networking infrastructure, and we have anecdotal evidence of individual Chinese citizens helping the African population learn to evade censorship of social media.

The Internet in Africa is qualitatively different from the Internet in North America and Europe. There is much less wired infrastructure, and most users rely on 4G wireless links. The reliability of network and electrical infrastructure is not assured. We found it difficult to assure the performance of our tools without them being tested in the region. On the other hand, the McKinsey Group has estimated that in 2025 the Internet in Africa will involve approximately 600 million users buying $75 billion in e-commerce goods and services, and the Internet will add approximately $300 billion to the region's economy.[ag]

Demographically and financially, the sub-Saharan Internet is growing, and we are witnessing an ongoing struggle between authoritarian governments and local democracy activists taking place largely over the Internet. While in many ways the Internet is helping the pro-democracy forces, it is also helping keep non-democratic governments in place.

Acknowledgments

We would like to acknowledge the support of the U.S. State Department's Bureau of Democracy, Human Rights, and Labor in part for this work through Internet Freedom for West Africa grant number SLM-AQM-12GR1033. The statements in this article are the opinions of the authors and do not reflect the positions of the U.S. government or the U.S. Department of State. We also wish to acknowledge the support we received from our seminar participants and the Fondation Robert Fiadjoe pour le Qualité and to thank the editor and reviewers for their thoughtful reviews and suggestion of improvements to this article, greatly improving its quality.

[ag] https://www.mckinsey.com/industries/high-tech/our-insights/lions-go-digital-the-internets-transformative-potential-in-africa

References

1. Dietrich, C.J., Rossow, C., Felix, Freiling, C., Bos, H., Van Steen, M., and Pohlmann, N. On botnets that use DNS for command and control. In *Proceedings of the Seventh European Conference on Computer Network Defense* (Gothenburg, Sweden, Sept. 6–7). IEEE Computer Society Press, 2011, 9–16.
2. Fu, Y., Yu, L., Hambolu, O., Ozcelik, I., Husain, B., Sun, J., Sapra, K., Du, D., Beasley, C., and Brooks, R. Stealthy domain-generation algorithms. *IEEE Transactions on Information Forensics and Security 12*, 6 (June 2017), 1430–1443.
3. Hagen, J. and Luo, S. *Why Domain-Generating Algorithms?* Trend Micro, Aug. 18, 2016; http://blog.trendmicro.com/domain-generating-algorithms-dgas
4. Roberts, H., Zuckerman, E., and Palfrey, J. *Circumvention Landscape Report: Methods, Uses, and Tools.* The Berkman Center for Internet & Society at Harvard University, Cambridge, MA, 2007; http://cyber.harvard.edu/sites/cyber.harvard.edu/files/2007_Circumvention_Landscape.pdf
5. Silva, S.S.C., Silva, R.M.P., Pinto R.C.G., and Salles, R.M. Botnets: A survey. *Computer Networks 57*, 2 (Feb. 2013), 378–403.

Richard Brooks (rrb@g.clemson.edu) is a professor of computer engineering in the Holcombe Department of Electrical and Computer Engineering at Clemson University, Clemson, SC, USA.

Lu Yu (lyu@g.clemson.edu) is a postdoctoral fellow of computer engineering in the Holcombe Department of Electrical and Computer Engineering at Clemson University, Clemson, SC, USA.

Yu Fu (fu2@g.clemson.edu) is a staff engineer at Palo Alto Networks, Palo Alto, CA, USA.

Oluwakemi Hambolu (ohambol@g.clemson.edu) is a Ph.D. student in the Holcombe Department of Electrical and Computer Engineering at Clemson University, Clemson, SC, USA.

John Gaynard (jgaynard@gmail.com) has taught innovation at ESIEE Engineering School in Paris, France, and the OU Business School, U.K., and spent much of his professional career consulting on strategic and telecoms issues in French West Africa.

Julie Owono (julie@internetsansfrontieres.org) is a lawyer and Executive Director of Internet Without Borders (https://internetwithoutborders.org/).

Archippe Yepmou (archippe@internetsansfrontieres.org) is a musical composer and President of Internet Without Borders (https://internetwithoutborders.org/).

Félix Blanc (fb.blanc@gmail.com) is head of public policy in Internet Sans Frontières (https://internetwithoutborders.org/) and a research fellow in the Center for Technology and Society in the Law Department of the Getulio Vargas Foundation, Rio de Janeiro, Brazil.

Watch the authors discuss their work in this exclusive *Communications* video. https://cacm.acm.org/videos/internet-freedom-in-west-africa

DOI:10.1145/3197544

The data comes from multiple optimal sources in parallel, helping reduce addressing and data-acquisition latency.

BY XIAONAN WANG

Data Acquisition in Vehicular Ad Hoc Networks

WITH THE AMOUNT of multimedia data large and growing larger, low-latency data acquisition represents an important practical goal for emerging Internet of Vehicles applications. Multihoming could help reduce such latency because it could let a single node use multiple addresses to acquire data in parallel.

Network researchers are thus trying to extend multihoming to vehicular ad hoc networks (VANETs), aiming to reduce latency in the Internet of Vehicles. But in VANETs with multihoming, a vehicle must be able to perform n addressing processes to be configured with addresses with n global network prefixes (GNPs). And getting a vehicle to use addresses with different GNPs to acquire data in parallel through the standard communication models is a significant engineering challenge. Here, I propose an address-separation mechanism so vehicles can be configured with addresses with different GNPs in a single addressing process, extending the k-anycast model to help acquire data in parallel.

Vehicles on the road today include abundant computer processing and storage, producing demand for con-

» key insights

- Because there is so much multimedia data, low-latency data acquisition could help ensure vehicles get what they need.
- Acquiring multimedia data through vehicular ad hoc networks helps deliver the data to networked vehicles.
- The *k*-anycast model can be extended to vehicular ad hoc networks so they can acquire data in parallel and help reduce latency in data acquisition.

necting VANETs to the Internet so they can acquire a variety of multimedia data.[1,6,12] In multihoming, one IP domain is identified by n ($n \geq 2$) GNPs, and a node can be configured with n addresses with different GNPs.[7] A node would use these addresses to acquire data in parallel, thus reducing data-acquisition latency.[3–5] However, network researchers trying to extend multihoming to VANETs must first address two main technical challenges:[3]

Addressing. A node usually performs an addressing process with addresses including one GNP;[2] that is, a node must perform n addressing processes to be configured with addresses with n GNPs, leading to considerable addressing latency; and

Data acquisition. In unicast and anycast models, a node acquires data from a single provider. In multicasting, a multicast address works only as a destination address. Each destination multicast member receives a copy of data from a particular source, so a multicast member actually acquires data from a single provider. A node cannot use addresses with different GNPs to acquire data from providers in parallel via unicast, anycast, or multicast.

Figure 1. Architecture.

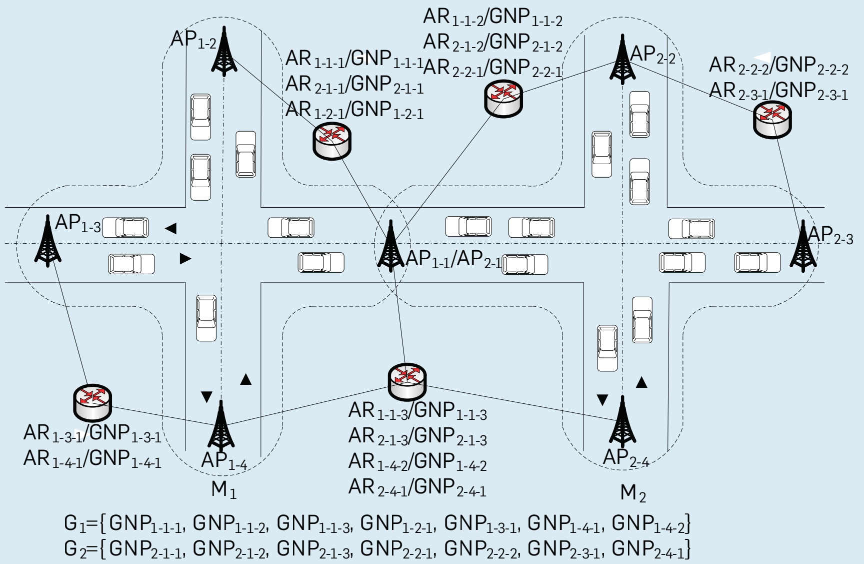
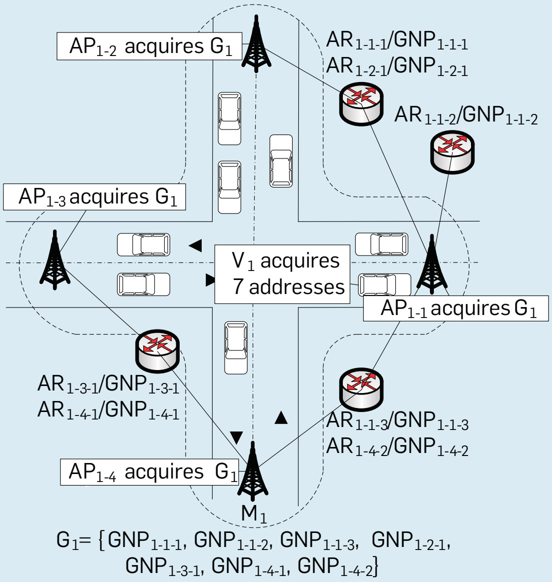

AR$_{1-1-2}$/GNP$_{1-1-2}$
AR$_{2-1-2}$/GNP$_{2-1-2}$
AR$_{2-2-1}$/GNP$_{2-2-1}$
AP$_{1-2}$
AR$_{1-1-1}$/GNP$_{1-1-1}$
AR$_{2-1-1}$/GNP$_{2-1-1}$
AR$_{1-2-1}$/GNP$_{1-2-1}$
AP$_{2-2}$
AR$_{2-2-2}$/GNP$_{2-2-2}$
AR$_{2-3-1}$/GNP$_{2-3-1}$
AP$_{1-3}$
AP$_{1-1}$/AP$_{2-1}$
AP$_{2-3}$
AR$_{1-3-1}$/GNP$_{1-3-1}$
AR$_{1-4-1}$/GNP$_{1-4-1}$
AP$_{1-4}$
M$_1$
AR$_{1-1-3}$/GNP$_{1-1-3}$
AR$_{2-1-3}$/GNP$_{2-1-3}$
AR$_{1-4-2}$/GNP$_{1-4-2}$
AR$_{2-4-1}$/GNP$_{2-4-1}$
AP$_{2-4}$
M$_2$
G$_1$={GNP$_{1-1-1}$, GNP$_{1-1-2}$, GNP$_{1-1-3}$, GNP$_{1-2-1}$, GNP$_{1-3-1}$, GNP$_{1-4-1}$, GNP$_{1-4-2}$}
G$_2$={GNP$_{2-1-1}$, GNP$_{2-1-2}$, GNP$_{2-1-3}$, GNP$_{2-2-1}$, GNP$_{2-2-2}$, GNP$_{2-3-1}$, GNP$_{2-4-1}$}

Figure 2. Addressing.


AP$_{1-2}$ acquires G$_1$
AR$_{1-1-1}$/GNP$_{1-1-1}$
AR$_{1-2-1}$/GNP$_{1-2-1}$
AR$_{1-1-2}$/GNP$_{1-1-2}$
AP$_{1-3}$ acquires G$_1$
V$_1$ acquires 7 addresses
AP$_{1-1}$ acquires G$_1$
AR$_{1-3-1}$/GNP$_{1-3-1}$
AR$_{1-4-1}$/GNP$_{1-4-1}$
AR$_{1-1-3}$/GNP$_{1-1-3}$
AR$_{1-4-2}$/GNP$_{1-4-2}$
AP$_{1-4}$ acquires G$_1$
M$_1$
G$_1$= {GNP$_{1-1-1}$, GNP$_{1-1-2}$, GNP$_{1-1-3}$, GNP$_{1-2-1}$, GNP$_{1-3-1}$, GNP$_{1-4-1}$, GNP$_{1-4-2}$}

Wang[9] proposed a k-anycast communication model in the IPv6 network with one GNP. In the k-anycast model,[9] one k-anycast group consists of k-anycast members that cooperate to provide data in parallel; that is, a user can acquire data from more than one member in parallel, greatly reducing data-acquisition latency. Due to the efficiency of the k-anycast model, vehicle-network researchers are looking to take advantage of the k-anycast idea. Addressing and data-acquisition latency can thus be reduced through multihoming and the k-anycast model. Based on my proposed architecture for VANET with multihoming, a vehicle can be configured with addresses with different GNPs through a single addressing process, substantially reducing addressing latency. Also based on my proposed architecture, the k-anycast model can be extended through a single GNP[9] to VANET with multiple GNPs so the vehicle would use addresses with multiple GNPs to acquire data from different k-anycast members in parallel, thus reducing data-acquisition latency.

There are two main differences between the data-acquisition mechanism I propose and the one suggested by the k-anycast mechanism:[9]

Multiple GNPs. The data-acquisition mechanism based on the i-anycast model[9] works in the IPv6 network with a single GNP, whereas the one I propose works in VANETs with multiple GNPs; and

A single GNP. In Wang,[9] the optimal k-anycast members that cooperate to provide data are selected based on one GNP, whereas the optimal k-anycast

Table 1. *k*-anycast address.

k-anycast ID	Reserved
w	128-*w*

Table 2. Content address.

k-anycast ID	Part ID set
w	128-*w*

members are chosen based on multiple GNPs.

Anycast is different from *k*-anycast. In anycast, a node acquires data from one anycast member, whereas in *k*-anycast, a node acquires data from more than one *k*-anycast member in parallel.

Architecture

In DAVM, a VANET consists of access points (APs) and vehicles and is connected to the Internet through access routers (ARs). The lanes enclosed by p_x ($p_x \geq 2$) APs construct a vehicular multihoming domain (VMD) M_x, and each AP is denoted by AP_{x-y} ($1 \leq y \leq p_x$). AP_{x-y} links with r_{x-y} ($1 \leq r_{x-y}$) AR(s) denoted by AR_{x-y-z} ($1 \leq z \leq r_{x-y}$) that identifies one GNP denoted by GNP_{x-y-z}. M_x can thus be defined by GNP set G_x, as shown in Equation (1). One vehicle in M_x can use each GNP in G_x to construct an IPv6 address and utilize the addresses with different GNPs to acquire data from providers in parallel. As shown in Figure 1, two VMDs, M_1 and M_2, are included. The lanes enclosed by AP_{x-y} ($x = 1$, $1 \leq y \leq 4$) form VMD M_1, which is defined by GNP set G_1, and the lanes enclosed by AP_{x-y} ($x = 2$, $1 \leq y \leq 4$) construct VMD M_2, which is defined by GNP set G_2.

$$G_x = \bigcup_{y=1}^{p_x} \bigcup_{z=1}^{r_{x-y}} GNP_{x-y-z} \tag{1}$$

In DAVM, one VMD is defined by a GNP set and the VMD-based architecture yields two main benefits:

Address separation. A single VMD can help achieve the proposed address-separation mechanism in VANET through multihoming to help reduce addressing latency. In the address-separation mechanism, a vehicle in one VMD is configured with a globally unique node ID through a single addressing process, then combines the node ID with each GNP in the GNP set defining the VMD to construct globally unique addresses with different GNPs. A vehicle can thus be configured with addresses with different GNPs through a single addressing process; and

In parallel. A VMD can help achieve the *k*-anycast model in the VANET with multiple GNPs to reduce data-acquisition latency; that is, a vehicle in one VMD can use the addresses with different GNPs to acquire data from different optimal *k*-anycast members in parallel.

Addressing

In order to reduce the address-configuration latency in VANET with multiple GNPs, I propose address separation as a way to achieve the addressing, whereby a vehicle in M_x performs only one addressing process to be configured with a globally unique node ID and is uniquely identified through this node ID during its lifetime. The vehicle then combines its node ID with each GNP in G_x to construct globally unique addresses with different GNPs. A vehicle can thus be configured with addresses with different GNPs through a single addressing process.

Node ID space. If a node ID is *w*-bits long (*w* is a positive integer) and the number of APs is 2^a ($1 \leq a < w-1$, *a* is a positive integer), then the node ID space $[1, 2^w-2]$ is divided into 2^a parts, with each part for one AP. The mth ($1 \leq m \leq 2^a$) AP's node ID $A(m)$ is shown in Equation (2), and the mth AP's node ID space $[L(m), U(m)]$ is shown in Equation (3) and Equation (4). Each AP thus has a unique node ID and maintains its globally unique node ID space.

$$A(m) = \begin{cases} 1; & m = 1 \\ (m-1) \cdot 2^{w-a}; & 2 \leq m \leq 2^a \end{cases} \tag{2}$$

$$L(m) = \begin{cases} 2; & m = 1 \\ (m-1) \cdot 2^{w-a} + 1; & 2 \leq m \leq 2^a \end{cases} \tag{3}$$

$$U(m) = \begin{cases} m \cdot 2^{w-a} - 1; & 1 \leq m \leq 2^a - 1 \\ m \cdot 2^{w-a} - 2; & m = 2^a \end{cases} \tag{4}$$

GNP set. In DAVM, an AP stores the GNP sets defining the VMDs it belongs to. In M_x, AP_{x-y} can acquire GNP_{x-y-z} by receiving a router advertisement from AR_{x-y-z}. When AP_{x-y} acquires the GNP set *x*–*y*, as shown in Equation (5), it then performs the three operations to acquire G_x:

$$G_{x-y} = \bigcup_{z=1}^{r_{x-y}} GNP_{x-y-z} \tag{5}$$

Broadcasts. AP_{x-y} sets G_x to G_{x-y} and broadcasts one *Neig-AP* message where the payload is G_{x-y-z}.

Performs operations. Following receipt of the *Neig-AP*, a vehicle or AP performs acquisition operations based on three cases:

Case 1. A vehicle outside an AP's communication range receives the *Neig-AP*, and the vehicle forwards the *Neig-AP* and repeats the operation;

Case 2. A vehicle within an AP's communication range receives the *Neig-AP*, then updates the destination address in the *Neig-AP* with the address of the AP, forwards the *Neig-AP*, and repeats the operation; and

Case 3. An AP receives the *Neig-AP*, then updates G_x by performing the union operation, as shown in Equation (6).

$$G_x = G_x \bigcup G_{x-y} \tag{6}$$

Ends. The process ends, as shown in Figure 2.

In this data-acquisition process, AP_{x-y} might employ a positioning method[10] to determine the VMD one *Neig-AP* comes from. As shown in Figure 2, AP_{1-1} in VMD M1 receives one *Neig-AP* from AP_{x-y} (x=1, $2 \leq y \leq 4$) and establishes GNP set G_1 defining M_1. Likewise, AP_{x-y} ($x = 1$, $2 \leq y \leq 4$) also acquires G_1 by receiving one *Neig-AP*. In this way, AP_{x-y} ($x = 1$, $1 \leq y \leq 4$) acquires G_1.

Address construction. A vehicle V_1 in M_x that begins to move uses a hardware ID (such as a media-access control, or MAC, address) as a temporary address and acquires a node ID from the nearest AP AP_{x-y} based on the following three-step process:

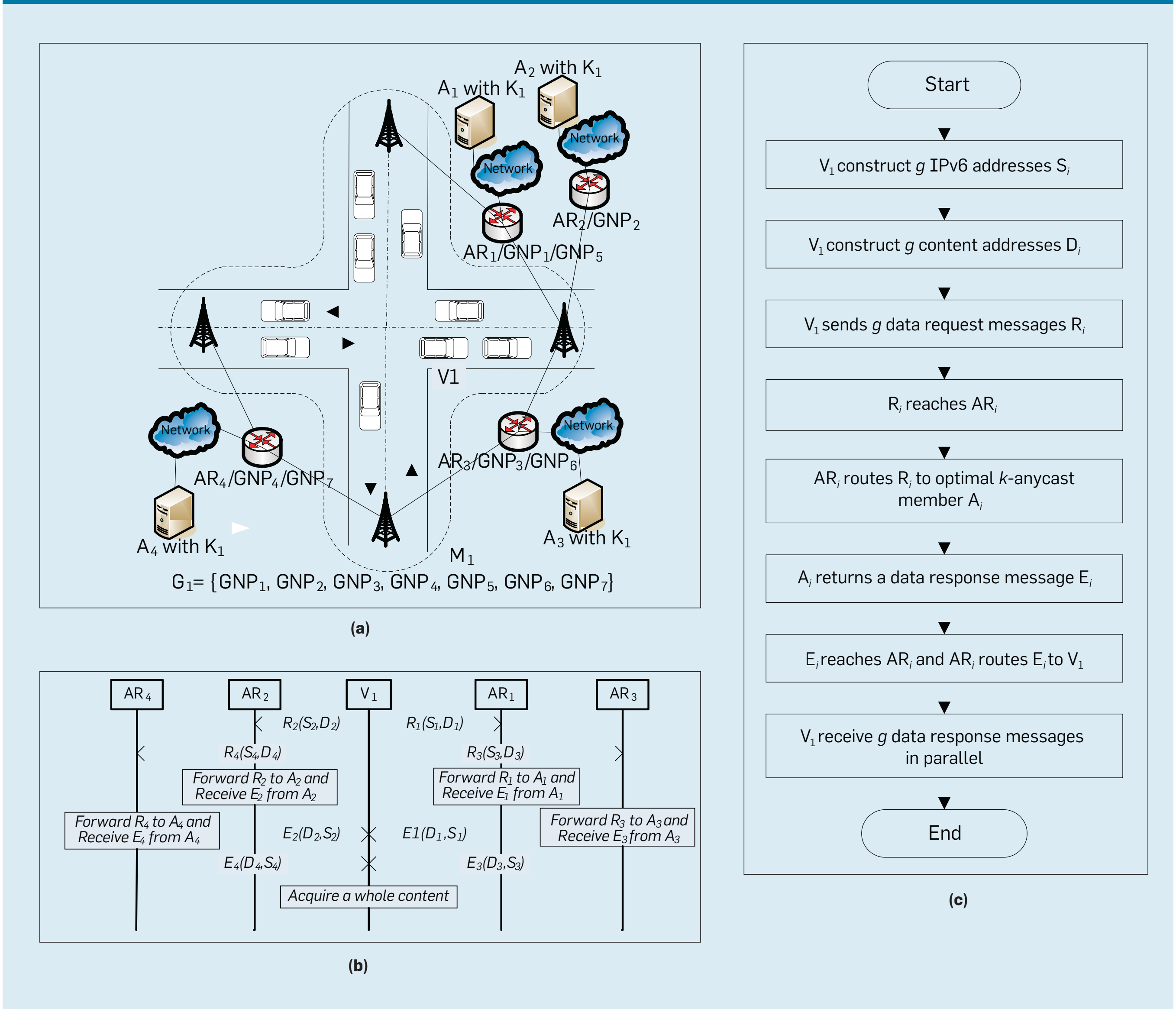
Sends. V_1 sends one *N-Req* message to AP_{x-y};

Marks the node. The AP_{x-y} receiving the *N-Req* returns one *N-Rep* message where the payload includes the assigned node ID and GNP set G_x defining M_x, then marks the node ID as "assigned"; and

Sets its node. The V_1 receiving the *N-Rep* sets its node ID to the node ID in the *N-Rep* and stores G_x, as shown in Figure 2.

Since AP_{x-y}'s node ID space is globally unique, the node ID that AP_{x-y} assigns to V_1 is also unique. A V_1 configured with a globally unique node ID then combines its node ID with each GNP in G_x to acquire a globally unique IPv6 address. In Figure 2, V_1 is located in M_1, which is defined by G_1. When V_1 acquires a unique node ID from AP_{1-1}, it then combines the node ID with each GNP in G_1 to acquire seven unique IPv6 addresses.

Figure 3. Data acquisition.


A2 with K1
A1 with K1
Network
Network
AR2/GNP2
AR1/GNP1/GNP5
V1
Network
Network
AR4/GNP4/GNP7
AR3/GNP3/GNP6
A4 with K1
A3 with K1
M1
G1= {GNP1, GNP2, GNP3, GNP4, GNP5, GNP6, GNP7}
(a)
AR4
AR2
V1
AR1
AR3
R2(S2,D2)
R1(S1,D1)
R4(S4,D4)
R3(S3,D3)
Forward R2 to A2 and Receive E2 from A2
Forward R1 to A1 and Receive E1 from A1
Forward R4 to A4 and Receive E4 from A4
Forward R3 to A3 and Receive E3 from A3
E2(D2,S2)
E1(D1,S1)
E4(D4,S4)
E3(D3,S3)
Acquire a whole content
(b)
Start
V1 construct g IPv6 addresses Si
V1 construct g content addresses Di
V1 sends g data request messages Ri
Ri reaches ARi
ARi routes Ri to optimal k-anycast member Ai
Ai returns a data response message Ei
Ei reaches ARi and ARi routes Ei to V1
V1 receive g data response messages in parallel
End
(c)

Data Acquisition

In the *k*-anycast model with a single GNP,[9] one *k*-anycast group consists of *k*-anycast members that cooperatively provide data in parallel. A user can thus acquire data from different *k*-anycast members in parallel, greatly reducing acquisition latency. In order to achieve the main DAVM objective of reduced data-acquisition latency, network researchers are trying to extend the *k*-anycast idea with one GNP[9] to VANET with multihoming so vehicles are able to use addresses with different GNPs to acquire data from different optimal providers in parallel.

In DAVM, a single *k*-anycast address defines one type of content, and all providers able to provide that content construct a *k*-anycast group uniquely specified by the *k*-anycast address. The *k*-anycast address structure consists of *w*-bit *k*-anycast ID field and reserved field whose value is zero (see Table 1). In DAVM, a particular content C is divided into *q* ($q \geq 2$) parts, with each part c_u ($q \geq u \geq 1$) uniquely identified by part ID d_u, as shown in Equation (7). A vehicle would use a content address to achieve *k*-anycast communications, with a content address consisting of the *k*-anycast ID and part ID set (see Table 2). The *k*-anycast ID specifies the type of desired content, and the part ID set indicates the specific parts of the content.

$$C=\bigcup_{u=1}^{q} c_u \qquad (7)$$

Vehicle V_1 is located in M_x, as defined by GNP set G_x, and acquires content C_1 identified by *k*-anycast address K_1 through the following five-step process:

Selects. V_1 selects *g* ($2 \leq g \leq |G_x|$) GNPs from G_x to construct *g* addresses denoted by S_i ($1 \leq i \leq g$) where the node ID is V_1's node ID and then constructs *g* content addresses denoted by D_i. In D_i, the *k*-anycast ID is the same as the ID in K_1, and the part ID set is P_i that defines the data parts B_i, as shown in Equation (8), where $c_{i\text{-}j}$ ($1 \leq j \leq |P_i|$) is the data part identified by element $d_{i\text{-}j}$ in P_i. This way, B_i satisfies Equation (9);

$$B_i=\bigcup_{j=1}^{|P_i|} c_{i\text{-}j} \qquad (8)$$

$$C_1 = \bigcup_{i=1}^{g} B_i \tag{9}$$

Sends. V_1 sends g data-request messages denoted by R_i in which the destination address is D_i and the source address is S_i;

Routs. Based on S_i, R_i is routed to AR_i, which specifies the GNP in S_i. Based on D_i, AR_i routes R_i to the optimal *k*-anycast member A_i with *k*-anycast address K_1. Based on P_i in D_i, A_i returns a data-response message E_i whereby the destination address is S_i and the payload is B_i;

Further routs. Based on the GNP in S_i, E_i is routed to AR_i and then, based on the node ID in S_i, E_i is routed to V_1; and

Receives data. V_1 can thus receive g data-response messages from different *k*-anycast members in parallel, as shown in Figure 3.

In Figure 3a, V_1 is located in M_1, which is defined by G_1 and connects with R_i ($1 \leq i \leq 4$), and the *k*-anycast group includes four members, A_i, that provide content C_1, as defined by *k*-anycast address K_1. C_1 is divided into fours parts, with each part defined by part ID d_i. V_1 constructs four content addresses D_i whereby the *k*-anycast ID is the same as the ID in K_1, and the part ID set is P_i. V_1 selects four GNPs, GNP_i, to construct four addresses, S_i. V_1 then sends four data-request messages R_i in which the source address is S_i and the destination address is D_i. Based on S_i, R_i reaches AR_i, which routes R_i to the optimal *k*-anycast member A_i. Based on P_i in D_i, A_i returns a data-response message E_i whereby the destination address is S_i and the payload is the content parts, B_i, as defined by P_i. Based on the GNP in S_i, E_i is routed to AR_i. Based on the node ID in S_i, E_i is routed to V_1, which receives different parts of C_1 from different *k*-anycast members in parallel.

Performance Evaluation

Following the earlier description of data acquisition, the addressing latency *TA* consists of the node ID request latency $T_{A\text{-}Req}$ and the node ID response latency $T_{A\text{-}Rep}$, as shown in Equations (10), (11), and (12), in which b is the data rate, t is the delay in transmitting a bit between neighbors, t_{Max} is the delay in transmitting a message with maximum size s_{Max} between neighbors, l is the distance between a vehicle and the nearest AP, and $s_{ID\text{-}Req}/s_{ID\text{-}Rep}$ is the size of an *N-Req*/*N-Rep*. Following the description of performance evaluation, the data-acquisition latency T_C consists of the data-request latency T_{Req} and data-response latency T_{Rep}, as shown in Equations (13), (14), and (15), in which l_i is the distance between AR_i and the nearest *k*-anycast member, l_i is the distance between AR_i and a particular vehicle, and s_{Req}/s_{Rep} is the size of a data request/response message. The notations used in DAVM are listed in Table 3.

Table 3. Notation.

Notation	Description
T	Addressing latency
$T_{A\text{-}Req}$	Node ID request latency
$T_{A\text{-}Rep}$	Node ID response latency
b	Data rate
t	Delay of transmitting a bit between neighbors
t_{Max}	Delay of transmitting a message with maximum size between neighbors
s_{Max}	Maximum size of a message
l	Distance between a vehicle and the nearest AP
$s_{ID\text{-}Req}/s_{ID\text{-}Rep}$	Size of one *N-Req*/*N-Rep*
T_C	Data acquisition latency
T_{Req}	Data request latency
T_{Rep}	Data response latency
l_i	Distance between AR_i and the nearest member
l_i'	Distance between AR_i and a vehicle
s_{Req}/s_{Rep}	Size of a data request/response message

Table 4. Simulation parameters.

Parameters	Values
Vehicle density	0.01vpm(vehicles/m)
Data rate	2Mbps
Number of lanes	2
Transmission radius	300m
Speed	20m/s
MAC	IEEE 802.11p
Mobility model	Freeway
Simulation time	*600s*
Rounds	20
Confidence level	0.95

$$T_A = T_{A\text{-}Req} + T_{A\text{-}Rep} \tag{10}$$

$$T_{A\text{-}Req} = \begin{cases} s_{IDReq}/b + t_{Max} \cdot (l-1)); \\ s_{ID\text{-}Req} > s_{Max} \\ t \cdot l \cdot s_{ID\text{-}Req}; s_{ID\text{-}Req} \leq s_{Max} \end{cases} \tag{11}$$

$$T_{A\text{-}Rep} = \begin{cases} s_{ID\text{-}Rep}/b + t_{Max} \cdot (l-1)); \\ s_{ID\text{-}Rep} > s_{Max} \\ t \cdot l \cdot s_{ID\text{-}Rep}; s_{ID\text{-}Rep} \leq s_{Max} \end{cases} \tag{12}$$

$$T_C = T_{Req} + T_{Rep} \tag{13}$$

$$T_{Req} = \begin{cases} \max_{i=1}^{g}(s_{Req}/b + t_{Max} \cdot (l_i + l_i' - 1)); \\ s_{Req} > s_{Max} \\ \max_{i=1}^{g}(t \cdot (l_i + l_i') \cdot s_{Req}); s_{Req} \leq s_{Max} \end{cases} \tag{14}$$

$$T_{Rep} = \begin{cases} \max_{i=1}^{g}(s_{Rep}/b + t_{Max} \cdot (l_i + l_i' - 1)); \\ s_{Rep} > s_{Max} \\ \max_{i=1}^{g}(t \cdot (l_i + l_i') \cdot s_{Rep}); s_{Rep} \leq s_{Max} \end{cases} \tag{15}$$

DAVM is evaluated in *ns*-2 using the simulation parameters in Table 4 in which the number of *k*-anycast members is equal to g. DAVM is compared with the addressing standard[2] and the data-acquisition scheme with a single GNP[9] called Data Acquisition One GNP, or DAOGNP, as shown in Figure 4 and Figure 5.

Figure 4. Addressing latency.

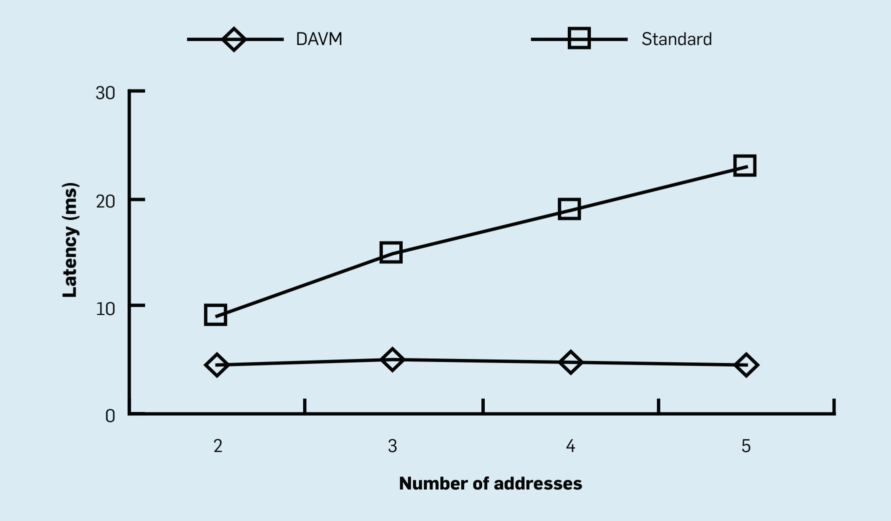
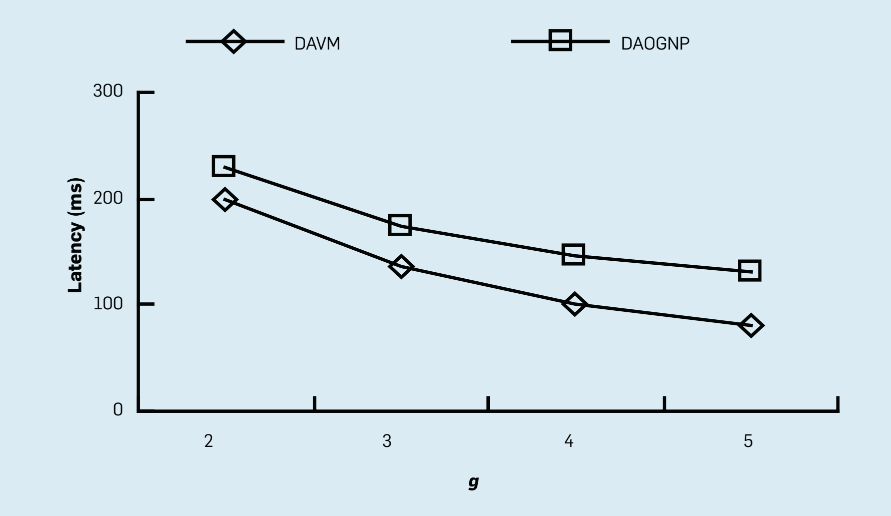


Figure 5. Data-acquisition latency.



As in Figure 4, with the increase in the number of addresses, the addressing latency in the standard increases, whereas the addressing latency in DAVM tends to be stable. Such stability follows from DAVM including an address-separation mechanism and a vehicle using a single addressing process configured with multiple addresses with different GNPs. As a result, addressing latency is only minimally affected by the number of addresses. In the standard, only a single addressing process is performed for each GNP, so the addressing latency grows with the number of addresses. As shown in Figure 5, with the increase in GNPs, the data-acquisition latency in both DAVM and DAOGNP decreases, but DAVM involves less data acquisition latency for two main reasons:

Based on multiple GNPs. In DAVM, the optimal *k*-anycast members that provide the data are selected based on multiple GNPs, whereas in DAOGNP, the optimal *k*-anycast member are selected based on a single GNP; and

Single address. In DAVM, a vehicle uses addresses with different GNPs to acquire data from optimal *k*-anycast members through different routing paths in parallel, whereas in DAOGNP, a node uses a single address with a single GNP to acquire data from relatively optimal *k*-anycast members.

Conclusion

My observation that multihoming and *k*-anycast can help lower data-acquisition latency to extend multihoming and *k*-anycast to VANET is what led me to propose DAVM as a way to reduce data-acquisition latency. My results show DAVM works for three main reasons:

- ▸ Extends multihoming to VANET;
- ▸ Extends the *k*-anycast idea with one GNP to VANET with multihoming so vehicles can use addresses with different GNPs to acquire data from multiple *k*-anycast members in parallel; and
- ▸ Provides the address-separation mechanism so vehicles can obtain multiple addresses with different GNPs through a single addressing process.

My future work will aim to take advantage of the powerful computing capabilities and abundant storage resources of APs to help improve addressing and data acquisition in VANET with multihoming.

Acknowledgment

This work is supported by the 333 Project Foundation (grant number BRA2016438), CERNET Innovation Project (grant number NGII20170106), and National Natural Science Foundation of China (grant number 61202440). C

References

1. Amadeo, M., Campolo, C., and Molinaro, A. Information-centric networking for connected vehicles: A survey and future perspectives. *IEEE Communications Magazine, 54*, 2 (Feb. 2016), 98–104.
2. Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and Carney, M. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6), RFC 3315.* Internet Engineering Task Force, Fremont, CA, 2003; http://www.ietf.org/rfc/rfc3315.txt
3. Gladisch, A., Daher, R., and Tavangarian, D. Survey on mobility and multihoming in future Internet. *Wireless Personal Communications 74*, 1 (Jan. 2014), 45–81.
4. Islam, S., Hashim, A.H.A., Habaebi, M.H., and Hasan, M.K Design and implementation of a multihoming-based scheme to support mobility management in NEMO. *Wireless Personal Communications 95*, 2 (Feb. 2017), 457–473.
5. Khatouni, A.S., Marsan, M.A., and Mellia, M. Video upload from public transport vehicles using multihomed systems. In *Proceedings of the IEEE Conference on Computer Communications* (San Francisco, CA, Apr. 10–14). IEEE Computer Society Press, 2016, 306–307.
6. Omar, H., Zhuang, W., and Li, L. Gateway placement and packet routing for multihop in-vehicle Internet access. *IEEE Transactions on Emerging Topics in Computing 3*, 3 (Mar. 2015), 335–351.
7. Troan, O., Miles, D., Matsushima, S., Okimoto, T., and Wing, D. *IPv6 Multihoming Without Network Address Translation, RFC 7157.* Internet Engineering Task Force, Fremont, CA, 2014; http://www.ietf.org/rfc/rfc7157.txt
8. Vegni, A.M. and Loscri, V. A survey on vehicular social networks. *IEEE Communications Surveys & Tutorials 17*, 4 (Apr. 2014), 2397–2419.
9. Wang, X. Analysis and design of a *k*-anycast communication model in IPv6. *Computer Communications 31*, 10 (Oct. 2008), 2071–2077.
10. Wang, X. and Zhong, S. Research on IPv6 address configuration for a VANET. *Journal of Parallel and Distributed Computing 73*, 6 (June 2013), 757–766.
11. Wang, X. and Zhu, X. Anycast-based content-centric MANET. *IEEE Systems Journal PP*, 99 (Nov. 2016), 1–9.
12. Zheng, Z., Lu, Z., Sinha, P., and Kumar, S. Ensuring predictable contact opportunity for scalable vehicular Internet access on the go. *IEEE/ACM Transactions on Networking 23*, 3 (Mar. 2015), 768–781.

Xiaonan Wang (nina_99999@163.com) is a professor in the Computer Science and Engineering Department of Changshu Institute of Technology, Jiangsu, Changshu, China.

ACM Books

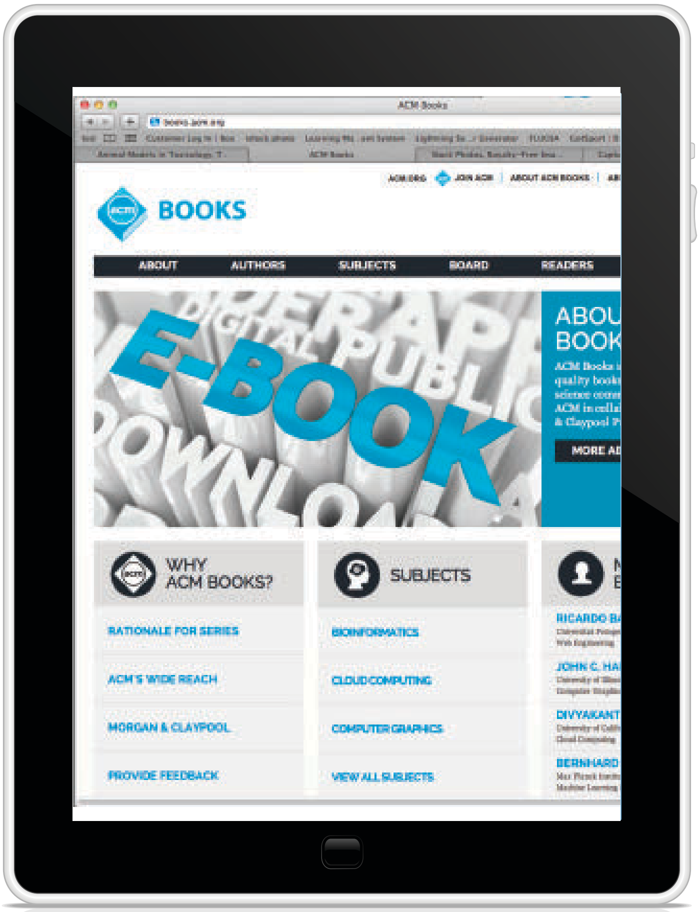
MORGAN & CLAYPOOL PUBLISHERS

Publish your next book in the ACM Digital Library

ACM Books is a new series of advanced level books for the computer science community, published by ACM in collaboration with Morgan & Claypool Publishers.

> *I'm pleased that ACM Books is directed by a volunteer organization headed by a dynamic, informed, energetic, visionary Editor-in-Chief (Tamer Özsu), working closely with a forward-looking publisher (Morgan and Claypool).*
>
> —Richard Snodgrass, University of Arizona

books.acm.org



ACM Books

- will include books from across the entire spectrum of computer science subject matter and will appeal to computing practitioners, researchers, educators, and students.
- will publish graduate level texts; research monographs/overviews of established and emerging fields; practitioner-level professional books; and books devoted to the history and social impact of computing.
- will be quickly and attractively published as ebooks and print volumes at affordable prices, and widely distributed in both print and digital formats through booksellers and to libraries and individual ACM members via the ACM Digital Library platform.
- is led by EIC M. Tamer Özsu, University of Waterloo, and a distinguished editorial board representing most areas of CS.

Proposals and inquiries welcome!

Contact: **M. Tamer Özsu,** Editor in Chief

booksubmissions@acm.org

Association for
Computing Machinery

Advancing Computing as a Science & Profession

DOI:10.1145/3129340

Tracing 20 years of progress in making machines hear our emotions based on speech signal properties.

BY BJÖRN W. SCHULLER

Speech Emotion Recognition

Two Decades in a Nutshell, Benchmarks, and Ongoing Trends

COMMUNICATION WITH COMPUTING machinery has become increasingly 'chatty' these days: Alexa, Cortana, Siri, and many more dialogue systems have hit the consumer market on a broader basis than ever, but do any of them truly notice our emotions and react to them like a human conversational partner would? In fact, the discipline of automatically recognizing human emotion and affective states from speech, usually referred to as *Speech Emotion Recognition* or SER for short, has by now surpassed the "age of majority," celebrating the 22nd anniversary after the seminal work of Daellert et al. in 1996[10]—arguably the first research paper on the topic. However, the idea has existed even longer, as the first patent dates back to the late 1970s.[41]

Previously, a series of studies rooted in psychology rather than in computer science investigated the role of acoustics of human emotion (see, for example, references[8,16,21,34]). Blanton,[4] for example, wrote that "*the effect of emotions upon the voice is recognized by all people. Even the most primitive can recognize the tones of love and fear and anger; and this knowledge is shared by the animals. The dog, the horse, and many other animals can understand the meaning of the human voice. The language of the tones is the oldest and most universal of all our means of communication.*" It appears the time has come for computing machinery to understand it as well.[28] This holds true for the entire field of *affective computing*—Picard's field-coining book by the same name appeared around the same time[29] as SER, describing the broader idea of lending machines emotional intelligence able to recognize human emotion and to synthesize emotion and emotional behavior.

Until now, the broader public has experienced surprisingly little *automatic recognition* of emotion in everyday life. In fact, only few related commercial products have found their way to the market, including the first-ever hardware product—the "Handy Truster"—which appeared around the turn of the millennium and claimed to be able to sense human stress-level and

» key insights

- **Automatic speech recognition helps enrich next-gen AI with emotional intelligence abilities by grasping the emotion from voice and words.**
- **After more than two decades of research, the field has matured to the point where it can be the "next big thing" in speech user interfaces, spoken language processing, and speech analysis for health, retrieval, robotics, security, and a plethora of further applications. This is also shown in the benchmarks of more than a dozen research competitions held in the field to date.**
- **While deep learning started in this field a decade ago, it recently pushed to end-to-end learning from raw speech data—just one of a couple of current breakthroughs.**

ILLUSTRATION BY VAULT49

deception contained in speech. Approximately 10 years later, the first broad-consumer market video game appeared. "Truth or Lies" (THQ) was equipped with a disc and a microphone for players to bring the popular "Spin the Bottle" game to the digital age. Unfortunately, the meta-review service metacritic.com reported only a score of 28 out of 100 based on only six reviews from professional critics. The tech side seemed premature: reviewers complained about "unstable tech" and "faulty software" that failed to achieve what it promised—detect lies from human speech. However, the first success stories can be observed at this time; including the European ASC-Inclusion project[a] that reports encouraging observations in open trials across three countries for a serious video game that teaches autistic children in a playful way how to best show emotions. Interestingly, a recent study shows that voice-only as modality seems best for humans' empathic accuracy as compared to video-only or audiovisual communication.[22]

Here, I aim to provide a snapshot of the state-of-the-art and remaining challenges in this field. Of course, over the years further overviews have been published that the reader may find of interest, such as references[2,6,15,20,38] or on the broader field of affective computing[17,43] where one finds an overview also on further modalities such as facial expression, body posture, or a range of bio-sensors and brain waves for the recognition of human emotion. These surveys cover progress up to 2013, but quite a bit has happened since then. Further, this short survey is the first to provide an overview on all open competitive challenges in this field to date. Finally, it distills a number of future tendencies discussed here for the first time.

The Traditional Approach

Let's start off by looking at the conventional way to build up an engine able to recognize emotion from speech.

Modeling. First things first: approaching the automatic recognition of emotion requires an appropriate emotion representation model. This raises two main questions: How to represent emotion per se, and how to optimally quantify the time axis.[32] Starting with representing emotion in an adequate way to ensure proper fit with the psychology literature while choosing a representation that can well be handled by a machine, two models are usually found in practice. The first model is discrete classes, such as the Ekman "big six" emotion categories, including anger, disgust, fear, happiness, and sadness—often added by a "neutral" rest-class as opposed to a *value* "continuous" dimension approach that appears to be the favored approach today.[17] In this second approach, the two axes *arousal* or activation (known to be well accessible in particular by acoustic features) and *valence* or positivity (known to be well accessible by linguistic features[17]) prevail alongside others such as power or expectation. One can translate between the categories and dimensions such as 'anger'→{*negative_valence*, *high_arousal*} in a coarse quantization. Other aspects of modeling include the temporal resolution[17] and the quality and masking of emotion, such as acted, elicited, naturalistic, pretended, and regulated.

Approaching the automatic recognition of emotion requires an appropriate emotion representation.

Annotation. Once a model is decided upon, the next crucial issue is usually the acquisition of labeled data for training and testing that suits the according emotion representation model.[13] A particularity of the field is the relatively high subjectivity and uncertainty in the target labels. Not surprising, even humans usually disagree to some degree as to what the emotion should be expressed in the speech of others—or any other modality accessible to humans.[13] Self-assessment could be an option, and is often used when no information to annotators is available or easily accessible, such as for physiological data. Suitable tools exist, such as the widely used PANAS, allowing for self-report assessment of positive and negative affect.[39] Yet, self-reported affect can be tricky as well, as no one has exact knowledge or memory of the emotion experienced at a moment in time. Further, observer rating can be a more appropriate label in the case of automatic emotion recognition that today largely targets assessment of the expressed emotion, rather than the felt emotion.

Likewise, external annotation may be more focused on the emotion ob-

[a] http://www.asc-inclusion.eu

served and being indeed observable. Likewise, usually five or more external raters' annotations—particularly in the case of crowdsourcing—form the basis of the construction of target labels, for example, by majority vote, or average in the case of a value continuous emotion representation.[17] Further, elimination of outliers or weighting of raters by their agreement/disagreement with the majority of raters can be applied, for example, by the *evaluator weighted estimator* (for example, Schuller and Batliner[32]). Such weighting becomes particularly relevant when crowdsourcing the labels, such as in a gamified way, for example, by the iHEARu-PLAY platform.[b] In the case of value continuous label and time representation, for example, for continuous arousal assessment, raters often move a joystick or slider in real time per emotion dimension while listening to the material to rate. This poses a challenge to time align different raters' annotations, as delays and speed variations in reaction time coin the annotation tracks. Such delays can be around four seconds,[37] and time warping enabled alignment algorithms should be preferred. In the case of discretized time, that is, judgment per larger segment of speech, pairwise comparisons leading to a ranking have recently emerged as an interesting alternative, as it may be easier for a rater to compare two or more stimuli rather than find an absolute value assignment for any stimulus.[17]

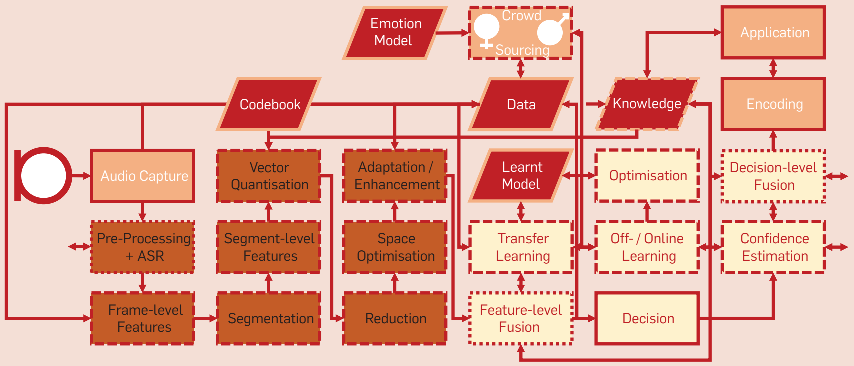
To avoid needs of annotation, past works often used acting (out an experience) or (targeted) elicitation of emotions. This comes at a disadvantage because the emotion may not be realistic or it may be questionable whether the right data collection protocol was followed such that the assumptions made on which emotion is finally collected would hold. In the present big data era, simply waiting for the emotion sought to become part of the collected data seems more feasible aiming at collection of emotion "from the wild" rather than from the lab.

Audio features. With labeled data at hand, one traditionally needs characteristic audio and textual features before feeding data into a suited machine-learning algorithm. This is an ongoing active subfield of research in the SER domain—the design of ideal features that best reflect the emotional content and should be robust against environmental noises, varying languages, or even cultural influences. Most of the established ones are rather low level, such as energy or spectral information, as these can be robustly determined. Yet, in the *synthesis* of emotion, there is a strong focus on prosodic features, that is, describing the intonation, intensity, and rhythm of the speech next to voice quality features. The automatic *analysis* of emotional speech often adds or even focuses entirely on spectral features, such as formants or selected band-energies, center of gravity, or roll-off points and cepstral features such as MFCC or mel-frequency bands as well as linear prediction coefficients.[2,15,38] Based on frame-by-frame extraction, one usually derives statistics by applying functionals that map a time series of frames with varying length onto a scalar value per segment of choice.[2,6] The length may vary with the *unit of analysis*, such as voiced or unvoiced sound, phoneme, syllable, or word. A second of audio material or shorter can be recommended considering the trade-off of having more information at hand versus higher parameter variability if the length of the analysis window is further increased. A high number of functionals is often used such as moments, extremes, segments, percentiles, or spectral functionals, for example, as offered by the openSMILE toolkit[c] that provides predefined feature sets that often serve as baseline reference in the research competitions in the field. The current trend is to increase the number of features up to some several thousands of brute-forced features that was often in stark contrast to the sparse amount of training material available in this field.[17,38,43]

Textual features. Going beyond how something is said, *textual features* as derived from the automatic speech recognition engine's output are mostly looking at individual words or sequences of these such as *n*-grams and their posterior probability to estimate a particular emotion class or value.[23] Alternatively, bag-of-word approaches are highly popular, where each textual entity in the vocabulary of all meaningful entities—from now on referred to as words—seen during vocabulary construction usually forms a textual feature.[18] Then, the frequency of occurrence of the words is used as actual feature value. It is possibly normalized to the number of occurrences in the training material, or to the current string of interest, length of the current string, or represented by logarithm, in binary format, and so on. Linguistically moti-

Figure 1. A current speech emotion recognition engine.

The chain of processing follows from the microphone (left) via the signal processing side of preprocessing and feature extraction (dark orange boxes) via the machine learning blocks (light orange) to encoding of information to feed into an application. Dashed boxes indicate optional steps. Five databases are shown in red. Crowdsourcing serves labeling efforts in the first place. ASR = Automatic Speech Recognition.



[b] https://ihearu-play.eu

[c] http://audeering.com/technology/opensmile

vated clustering of word variants may be applied, such as by stemming or representing morphological variants like different tenses. Also, "stopping," or the elimination of entities that do not occur sufficiently or frequently or seem irrelevant from a linguistic or expert's point of view, can be considered. However, in the recent years of increasingly big textual and further data resources to train from, the representation type of the word frequencies, as well as stemming and stopping, seem to have become increasingly irrelevant.[33] Rather, the retagging by word classes, such as part-of-speech tagging, for example, by groups such as noun, verb, or adjective, semantic word groups such as standard linguistic dimensions, psychological processes, personal concerns, and spoken categories as in the LIWC toolkit[d] or even the translation to affect categories or values by linguistic resources such as SenticNet[e] and others, or via relationships in ConceptNet,[f] General Inquirer,[g] WordNet,[h] and alike, can help to add further meaningful representations.

A promising recent trend is to use either soft clustering, that is, not assigning an observed word to a single word in the vocabulary or more general consideration of embedding words such as by word2vec approaches or convolutional neural networks. Alternatively, recurrent neural networks—possibly enhanced by long short-term memory[33]—and other forms of representation of longer contexts seem promising.

It should be noted that the traditional field of *sentiment analysis* is highly related to the recognition of emotion from text, albeit traditionally rather dealing with written and often longer passages of text.[33] This field offers a multiplicity of further approaches. A major difference is given by the uncertainty one has to deal with in spoken language—ideally, by incorporating confidence measures or *n*-best alternative hypotheses from the speech recognizer. Also, spoken language naturally differs from written text by lower emphasis on grammatical correctness, frequent use of word fragments, and so on. In particular, non-verbal vocalizations such as laughter, hesitations, consent, breathing, and sighing frequently occur, and should best be recognized as well, as they are often highly informative as to the emotional content. Once recognized, they can be embedded in a string.

Acoustic and linguistic feature information can be fused directly by concatenation into one single feature vector if both operate on the same time level, or by late fusion, that is, after coming to predictions per feature stream.[23] The latter also allows for representation of different acoustic or linguistic feature types on different time levels. As an example, one can combine bags-of-phonemes per fixed-length chunk of audio with turn-level word histograms in a late(r) fusion manner.

Peeking under the engine's hood. Now, let us look under the hood of an entire emotion recognition engine in Figure 1. There, one can see the features described here are the most characteristic part of a *speech* emotion recognizer —the rest of the processing chain is mainly a conventional pattern recognition system, and will thus not be further explored here. Some blocks in the figure will be mentioned in more detail later. Others, such as the learning part or, which classifier or regressor is popular in the field, will be illustrated by the practical examples from research competitions' results shown below.

En Vogue: The Ongoing Trends

Here, I outline a number of promising avenues that have recently seen increasing interest by the community. Obviously, this selection can only represent a subset, and many others exist.

Holistic speaker modeling. An important aspect of increased robustness is to consider other states and traits that temporarily impact on the voice production. In other words, one is not only emotional, but also potentially tired, having a cold, is alcohol intoxicated, or, sounds differently because being in a certain mood. Likewise, modern emotion recognition engines should see the larger picture of a speaker's states and traits beyond the emotion of interest to best recognize it independent of such co-influencing factors. As training a holistic model is difficult due to the almost entire absence of such richly annotated speech data resources that encompass a wide variety of states and traits, weakly supervised cross-task labeling offers an alternative to relabel databases of emotional speech in a richer way.

Efficient data collection. An ever-present if not main bottle neck since the beginning is the scarcity of speech data labeled by emotion. Not surprising, a major effort has been made over the last years to render data collection and annotation as efficient as possible.[17,38,43]

Weakly supervised learning. *Semi-supervised learning* approaches could prove successful in exploiting additional unlabeled data, once an initial engine was trained.[12,25] The idea is to have the machine itself label new previously unseen data—ideally only if a meaningful confidence measure is exceeded. However, it seems reasonable to keep human labeling in the loop to ensure a sufficient amount of quality labels. *Active learning* can help to reduce such human labeling requirements significantly. The machine preselects only those unlabeled instances for human labeling, which seem of particular interest. Such interest can be determined, for example, based upon whether a sample is likely to be from a class or interval on a continuous dimension that has previously been seen less than others. Further, the expected change in model parameters of the learned model can be the basis—if knowing the label would not change the model, there is no interest in spending human-labeling efforts. An extension can be to decide on how many and which humans to ask about a data point.

As mentioned earlier, emotion is often subjective and ambiguous. One usually must acquire several opinions. However, the machine can gradually learn "whom to trust when" and start with the most reliable labeler, for example, measured by the individual's average agreement with the average labeler population. If the label deviates from what the machine expects, a next opinion can be crowdsourced—ideally from the labeler who in such case would be most reliable. Putting these two ideas—semi-supervised and active learning —together, leads to the par-

[d] http://liwc.wpengine.com
[e] http://sentic.net
[f] http://conceptnet.io
[g] http://www.wjh.harvard.edu/ inquirer
[h] http://wordnet.princeton.edu

ticularly efficient *cooperative learning* of machines with human help.[24] In this approach, the machine decides based upon its confidence in its estimate whether it can label the data itself, such as in case of high confidence. If it is not sufficiently confident, it evaluates whether asking a human for aid is worth it. The overall process can be executed iteratively, that is, once newly labeled data either by the machine or a human is obtained, the model can be retrained, which will mostly increase its reliability and confidence. Then, the data that had not been labeled in a previous iteration might now be labeled by the machine or considered as worth labeling by a human. Monitoring improvements on test data is mandatory to avoid decreasing reliability.

If no initial data exists to start the iterative loop of weakly supervised learning, but similar related data is at hand, transfer learning may be an option.[11] To give an example, one may want to recognize the emotion of child speakers, but has only adult emotional speech data at hand. In such case, the features, the trained model, or even the representation, and further aspects can be transferred by learning from the data to the new domain. A broad number of transfer learning and domain adaptation algorithms exists and have been applied in this field, such as in Abdelwahab and Busso.[1] An interesting option of data enrichment can be to include other non-speech audio: as perception of certain emotional aspects such as arousal or valence seem to hold across audio types including music and general sound, one can seemingly train a speech emotion recognizer even on music or sound, as long as it is labeled accordingly.[40]

Obviously, transfer learning can help to make the types of signal more reusable to train emotion recognition engines across these audio types. Even image pretrained deep networks have recently been used to classify emotion in speech based on spectral representations at very impressive performance by the auDeep toolkit.[i] Should collecting and/or labeling of speech data not be an option, also *synthesized speech* can be considered for training of acoustic emotion models—either using synthesis of emotional speech, or simply to enrich the model of neutral speech by using non-emotional synthesized speech.[26] This can be beneficial, as one can generate arbitrary amounts of speech material at little extra cost varying the phonetic content, the speaker characteristics, and alike. Ideally, one could even ad-hoc render a phonetically matched speech sample in different emotions to find the closest match. A similar thought is followed by the recent use of generative adversarial network topologies, where a first neural network learns to synthesize training material, and another to recognize real from synthesized material and the task of interest.[5] Obviously, transfer learning can bridge the gap between artificial and real speech. In future efforts, a closer and immediate coupling between synthesis and analysis of emotional speech could help render this process more efficient.

An important aspect of increased robustness is to consider other states and traits that temporarily impact on voice production.

If no *annotated data* is available, and no emotional speech synthesizer is at hand, *unsupervised learning* could help if the knowledge of the emotion is not needed explicitly and in human-interpretable ways. An example is the integration of information on emotion in a spoken dialogue system: if features that bear information on the emotional content are used during unsupervised clustering of emotionally unlabeled speech material, one may expect the clusters to represent information related to emotion. The dialogue system could then learn—best reinforced—how to use the information on the current cluster in a dialogue situation to decide on its reaction based on observations of human-to-human dialogue. Likewise, at no point would someone know exactly what the clusters represent beyond designing the initial feature set for clustering to reflect, say, emotion; yet, the information could be used. Should no speech data be available, rule-based approaches could be used, which exploit the knowledge existing in the literature. A basis will usually be a speaker normalization. Then, one measures if the speech should, for example, be faster, higher pitched, or louder to assume a joyful state. Yet, given the oversimplification of a high-dimensional non-linear mapping problem, such an approach would, unfortunately, have limits.

[i] https://github.com/auDeep/auDeep

Data-learned features. As the quest for the optimal features has dominated the field similarly as the ever-lacking large and naturalistic databases, it is not surprising that with increased availability of the latter the first can be targeted in a whole new way, that is, *learn* features from data. This bears the charm that features should be optimally fitted to the data. Further, higher-level features could be learned. On the downside, one may wonder about potentially decreased generalization ability across databases. Below, two currently popular ways of learning feature representations are introduced.

The idea to cluster chunks of audio into words to then be able to treat these just like textual words during further feature extraction, for example, by histogram representation as "bag of audio words" was first used in sound recognition, but has found its way into recognition of emotion in speech.[31] Interestingly, these form some kind of modeling in between acoustic and linguistic representation depending on the low-level features that are used as basis.[31] As an example, one may use wavelet or cepstral coefficients and cluster these to obtain the audio words and the vocabulary built up by all found audio words. An even simpler, yet often similarly effective way is random sampling *k* vectors as audio words, that is, executing only the initialization of *k*-means. Then, the actual feature could be frequency of occurrence per audio word in a larger time window such as a second, a turn, or alike, for example, by the openXBOW tool.[j]

Split-vector quantization allows you to group the basis features to derive several histograms, for example, one for prosodic features and one for spectral features. The construction of this vocabulary is the actual data-injection step during feature learning, as speech data will be needed to reasonably build it up. There exists a huge potential of unexploited, more elaborate forms of audio words, such as variable length audio-words by clustering with dynamic time warping, soft-assignments of words during histogram calculation, audio-word embeddings, audio-word retagging or hierarchical clustering, such as the part-of-speech tagging in textual word handling, or speech component audio words by executing non-negative matrix factorization or alike, and creating audio words from components of audio.

The "neuro"-naissance or renaissance of neural networks has not stopped at revolutionizing automatic speech recognition.

The "neuro"-naissance or renaissance of neural networks has not stopped at revolutionizing automatic speech recognition. Since the first publications on deep learning for speech emotion recognition (in Wöllmer et al.,[42] a long-short term memory recurrent neural network (LSTM RNN) is used, and in Stuhlsatz et al.[35] a restricted Boltzman machines-based feed-forward deep net learns features), several authors followed this idea to learn the feature representation with a deep neural network, for example, Cibau[7] and Kim et al.[19] Convolutional neural networks (CNN) were also successfully employed to learn emotional feature representations.[27] The first end-to-end learning system for speech emotion recognition was recently presented by a sequence of two CNN layers operating at different time resolutions: 5ms first, then 500ms followed by a LSTM RNN at highly impressive performance.[37,k] In future topologies, one may consider stacking neural layers with different purposes such as speech denoising, feature extraction, feature enhancement, feature bundling, for example, by use of a bottleneck layer, and classification/regression with memory.[33]

Confidence measures. Given the higher degree of subjectivity of the task and imperfect recognition results, the provision of *confidence measures* of an emotion estimate seems mandatory in any application context.[17] However, the estimation of meaningful independent confidence measures beyond direct measures coming from the machine-learning algorithm, for example, distance to the hyperplane in kernel machines, softmax functions at the output layer of neural networks, or alike, has hardly been researched in SER.[17] Four main directions seem promising: 1) Automatic estimation of human labelers' agreement on unseen data: instead of training the emotion as a target, one can train a classifier on the number of raters that agreed on

[j] https://github.com/openXBOW/openXBOW

[k] A recent toolkit is found at https://github.com/end2you/end2you.

the label or, the standard deviation or alike in case of a regression task. Then, by automatically estimating human agreement on *novel* data, one obtains an impression on the difficulty of the current emotion prediction. In other words, one learns to estimate for new data if humans would agree or likely disagree on its emotion. Ideally, this can be targeted as a multitask problem learning the emotion and human agreement in parallel. 2) One can train a second learning algorithm to predict errors of the emotion recognition engine. To this end, one needs to run the trained emotion recognizer versus the development data to then train the confidence estimator on the errors or non-errors of the SER engine observed on that data. In case of a regression task, the linear error or other suited measures can be used as target. 3) Estimating the similarity of the data to the training data can be another option. A possible solution is training a compression autoencoder (a neural network that maps the feature space input onto itself to learn for example a compact representation of the data) on the data the emotion recognition was trained upon. Then, the new data to be handled can be run through the autoencoder. If the deviation between input and output of the autoencoder is high, for example, measured by Euclidean distance, one can assume low confidence in the emotion recognition results as the data is likely to be highly dissimilar. 4) Estimating acoustic degradation or word error rate. On a final note, reliable confidence measures are also the heart-piece of efficient weakly supervised learning.

Coming Clean: The Benchmarks

But how reliable are SER engines? This can partially be answered looking at the research challenges held in the field up to now. While the first official competition event with properly defined train and test sets and labels unknown to the participants—the Interspeech 2009 Emotion Challenge[l]—dates back nine

[l] http://compare.openaudio.eu.

Benchmark results of the SER challenge events.

Databases = the basis of data used in the competitions. Note that sometimes only subsets have been used. Only challenges are listed that provided audio only results (thus excluding, for example, AVEC 2014 and EmotiW since 2015). Some abbreviations here are obvious, others include lng=language (by country code ISO 3166 ALPHA-2 where "–" indicates an artificial language). hrs/spks/# = hours/speakers/number of data points. Task gives the number of classes or the dimensions =(A)rousal, (V)alence, (P)ower, (E)xpectation. "·2"= a binary classification per dimension. oS = openSMILE (feature extractor with standardized feature sets). EC = Interspeech Emotion Challenge. CRNN = CNN followed by a recurrent neural network with LSTM. RF = Random Forests. SVM/R = Support Vector Machines/Regression. BoAW = Bag-of-Audio-Words. UA = Unweighted Accuracy. WA = Weighted Accuracy. MAP = Macro Average Precision. PCC = Pearson's Correlation Coefficient. CCC = Concordance Correlation Coefficient. Baseline results follow the order under each "task."

Challenge	Database	lng	Quality	hrs/spks/#	task	unit	# feat	model	baseline
EC 09	FAU AEC	DE	lab	9.1/51/18216	2/5	chunk	384 oS	SVM	.677/.382 UA
ComParE 13	GEMEP	–	lab	~.6/10/1260	AV·2/12	turn	6373 oS	SVM	.750/.616/.409 UA
AVEC 11	SEMAINE	UK	lab	3.7/24/50350	AVPE·2	word	1941 oS	SVM	.412/.558/.527/.592 WA
AVEC 12	SEMAINE	UK	lab	3.7/24/50350	AVPE	word	1841 oS	SVR	.014/.040/.016/.038 PCC
AVEC 13	AViD	DE	lab	240/292/864k*	AV	sgmt	2268 oS	SVR	.090/.089 PCC
AVEC 15	RECOLA	FR	VoIP	2.3/27/202527	AV	sgmt	102 oS	SVR	.228/.068 CCC
AVEC 16	RECOLA	FR	VoIP	2.3/27/202527	AV	sgmt	88 oS	SVR	.648/.375 CCC
							–	CRNN	.686/.261 CCC
							BoAW	SVR	.753/.430 CCC
AVEC 17	SEWA	DE	VoIP	3/64/106896	AV	sgmt	BoAW	SVR	.225/.244 CCC
EmotiW 13	AFEW 3.0	US	film	~.8/315/1088	7	clip	1582 oS	SVM	.2244 WA
EmotiW 14	AFEW 4.0	US	film	~1.0/428/1368	7	clip	1582 oS	SVM	.2678 WA
MEC 16	CHEAVD	CN	film/TV	2.3/238/2852	8	clip	88 oS	RF	.2402 MAP/.2436 WA
MEC 17	CHEAVD 2.0	CN	film/TV	7.9/527/7030	8	clip	88 oS	SVM	.392 MAP/.405 WA

years by now, several further followed. In 2011, the first AudioVisual Emotion Challenge (AVEC 2011) took place, which also featured a speech-only track. By now, seven annual AVEC challenges took place[m]—in 2015 physiological signal information was added for the first time. The Interspeech Computational Paralinguistics challengE (Interspeech ComParE) series revisited SER as task in 2013. Meanwhile, challenges considering media-material such as clips of films appeared, namely the annual (since 2013) Emotion in the Wild Challenge (EmotiW[14]) run, and the new Multimodal Emotion Challenge (MEC 2016 and MEC 2017[n]). A loser relation to emotion in speech is given in further challenges such as MediaEval[o] ("affective (2015) /emotional (2016) impact of movies" task).

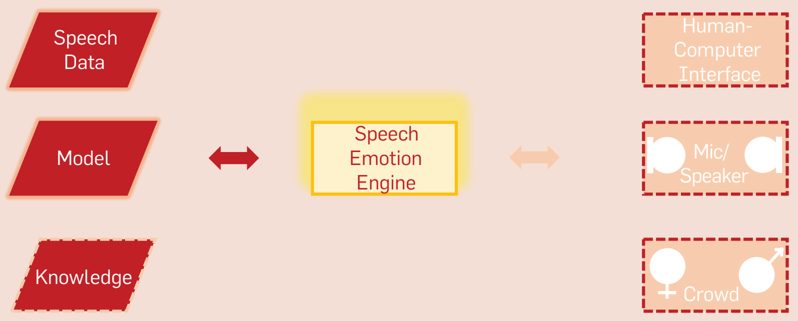
The accompanying table presents an overview on the challenges and their results to date that focused on SER. Interestingly, all challenges used the same feature extractor for the baselines. For comparison, the AVEC 2016 results for end-to-end learning[37] and Bags-of-Audio-Words[31] are further given, which are no official baselines. At press time, the series MEC is rerun, and the series ComParE is calling for participation for their 2018 reinstantiations offering novel affect tasks on atypical and self-assessed affect.

One would wish to compare these challenges in terms of technical or chronological improvements over the years. However, as the table indicates, the same database was used only once in two challenges with the same task definition (AVEC 15/16). There, one notices a striking improvement in the baseline of this challenge in the more recent edition. It seems desirable to rerun former tasks more often for a better comparability across years rather than having a mere provision of snapshots. However, the table shows that the task attempted was becoming increasingly challenging, going from lab to voice over IP to material from films with potential audio overlay.

Further, one would want to see the results of these events set into relation with human emotion perception benchmarks. Again, this is not straightforward for the following reasons: the ground truth does not exist in a reliable way—the data was labeled by a small number of humans in the first place. Comparing it to the perception of other humans on the test data would thus not be entirely fair, as they would likely have a different perception from those who labeled the training and test data. Further, there simply is no perception study available on these sets, indicating another white spot in the tradition of challenge culture in the field. Perhaps the more important question would be how these results relate to acceptable rates for human-machine applications. Such numbers are unfortunately also largely missing and would need to be provided by application developers stemming from according usability studies. To provide a statement non-the-less, the technology can already be used in a range of applications as outlined above, and seems to improve over time, reaching closer to human performance.

Figure 2. A modern speech emotion recognition engine.

Ideally, this engine experiences life-long learning 24/7 to analyze and synthesize emotion across languages and cultures. It gains feedback from the crowd in cooperative, gamified, and reinforced ways learning end-to-end and transferring gained knowledge. For a holistic understanding, it integrates contextual knowledge such as other speaker states and traits.



Moonshot Challenges?

Seeing the ongoing trends in the field, one may wonder what is left as high hanging fruit, grand challenge, or even a moonshot challenge. Certainly, several further steps must be taken before SER can be considered ready for broad consumer usage "in the wild." These include robustness across cultures and languages as one of the major white spots in the literature. A number of studies show the downgrades one may expect when going cross-language in terms of acoustic emotion recognition.[3] As to cross-cultural studies, these are still particularly sparse, and there exists practically no engine that is adaptive to cultural differences at the time. Beyond cross-cultural robustness, such against atypicality must be further investigated. For example, a few studies deal with emotion portrayal of individuals on the autism spectrum.[30] Further, the assessment of emotion of speaker groups has hardly been targeted. In a first step, this requires dealing with far-field acoustics, but it also must ideally isolate speakers' voices to analyze overlapping speech in search of emotional cues to then come to a conclusion regarding a groups' emotion. A potentially more challenging task may then be the recognition of irony or sarcasm as well as regulation of emotion. Differences between the acoustic and the linguistic channels may be indicative, but the research up to this point is limited. Next, there is little work to be found on speaker long-term adaptation, albeit being highly promising.

A genuine moonshot challenge, however, may be to target the *actual* emotion of an individual sensed by speech analysis. Up to this point, the gold standard is to use other human raters' assessment, that is, ratings or

[m] http://sspnet.eu/avec201x, with $x \in [1–7]$
[n] http://www.chineseldc.org/htdocsEn/emotion.html
[o] http://multimediaeval.org

annotations, as an "outer emotion", as perceived by others, as learning target. Obviously, this can be highly different from the "inner emotion" of an individual. To assess it, one will first need a ground truth measurement method, for example, by deeper insight into the cognitive processes as measured by EEG or other suited means. Then, one will also have to develop models that are robust against differences between expressed emotion and the experienced one—potentially by deriving further information from the voice which is usually not accessible to humans such as the heart rate, skin conductance, current facial expression, body posture, or eye contact,[32] and many further bio-signals.

Obviously, one can think of many further interesting challenges such as emotion recognition "from a chips bag" by high-speed camera capture of the vibrations induced by the acoustic waves,[9] in space, under water, and, of course, in animal vocalizations.

Conclusion

In this article, I elaborated on making machines hear our emotions from end to end—from the early studies on acoustic correlates of emotion[8,16,21,34] to the first patent[41] in 1978, the first seminal paper in the field,[10] to the first end-to-end learning system.[37] We are still learning. Based on this evolution, an abstracted summary is shown in Figure 2 presenting the main features of a modern engine. Hopefully, current dead-ends, such as the lack of rich amounts of spontaneous data that allow for coping with speaker variation, can be overcome. After more than 20 years into automatic recognition of emotion in the speech signal, we are currently witnessing exciting times of change: data learned features, synthesized training material, holistic architectures, and learning in an increasingly autonomous way—all of which can be expected to soon lead to the rise of broad day-to-day usage in many health, retrieval, security, and further beneficial use-cases alongside—after years of waiting[36]—the advent of emotionally intelligent speech interfaces.

Acknowledgments

The research leading to these results has received funding from the European Union's HORIZON 2020 Framework Programme under the Grant Agreement No. 645378.

References

1. Abdelwahab, M. and Busso, C. Supervised domain adaptation for emotion recognition from speech. In *Proceedings of ICASSP*. (Brisbane, Australia, 2015). IEEE, 5058–5062.
2. Anagnostopoulos, C.-N., Iliou, T. and Giannoukos, I. Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. *Artificial Intelligence Review 43*, 2 (2015), 155–177.
3. Bhaykar, M., Yadav, J. and Rao, K.S. Speaker dependent, speaker independent and cross language emotion recognition from speech using GMM and HMM. In *Proceedings of the National Conference on Communications*. (Delhi, India, 2013). IEEE, 1–5.
4. Blanton, S. The voice and the emotions. *Q. Journal of Speech 1*, 2 (1915), 154–172.
5. Chang, J. and Scherer, S. Learning Representations of Emotional Speech with Deep Convolutional Generative Adversarial Networks. *arxiv.org*, (arXiv:1705.02394), 2017.
6. Chen, L., Mao, X., Xue, Y. and Cheng, L.L. Speech emotion recognition: Features and classification models. *Digital Signal Processing 22*, 6 (2012), 1154–1160.
7. Cibau, N.E., Albornoz. E.M., and Rufiner, H.L. Speech emotion recognition using a deep autoencoder. San Carlos de Bariloche, Argentina, 2013, 934–939.
8. Darwin, C. *The Expression of Emotion in Man and Animals*. Watts, 1948.
9. Davis, A., Rubinstein, M., Wadhwa, N., Mysore, G. J., Durand, F. and Freeman, W.T. The visual microphone: Passive recovery of sound from video. *ACM Trans. Graphics 33*, 4 (2014), 1–10.
10. Dellaert, F., Polzin, T. and Waibel, A. Recognizing emotion in speech. In *Proceedings of ICSLP 3*, (Philadelphia, PA, 1996). IEEE, 1970–1973.
11. Deng, J. Feature Transfer Learning for Speech Emotion Recognition. PhD thesis, Dissertation, Technische Universität München, Germany, 2016.
12. Deng, J., Xu, X., Zhang, Z., Frühholz, S., and Schuller, B. Semisupervised Autoencoders for Speech Emotion Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 26*, 1 (2018), 31–43.
13. Devillers, L., Vidrascu, L. and Lamel, L. Challenges in real-life emotion annotation and machine learning based detection. *Neural Networks 18*, 4 (2005), 407–422.
14. Dhall, A., Goecke, R., Joshi, J., Sikka, K. and Gedeon, T. Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *Proceedings of ICMI* (Istanbul, Turkey, 2014). ACM, 461–466.
15. El Ayadi, M., Kamel, M.S., and Karray, F. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition 44*, 3 (2011), 572–587.
16. Fairbanks, G. and Pronovost, W. Vocal pitch during simulated emotion. *Science 88*, 2286 (1938), 382–383.
17. Gunes, H. and Schuller, B. Categorical and dimensional affect analysis in continuous input: Current trends and future directions. *Image and Vision Computing 31*, 2 (2013), 120–136.
18. Joachims, T. *Learning to classify text using support vector machines: Methods, theory and algorithms.* Kluwer Academic Publishers, 2002.
19. Kim, Y., Lee, H. and Provost, E.M. Deep learning for robust feature generation in audiovisual emotion recognition. In *Proceedings of ICASSP*, (Vancouver, Canada, 2013). IEEE, 3687–3691.
20. Koolagudi, S.G. and Rao, K.S. Emotion recognition from speech: A review. *Intern. J. of Speech Technology 15*, 2 (2012), 99–117.
21. Kramer, E. Elimination of verbal cues in judgments of emotion from voice. *The J. Abnormal and Social Psychology 68*, 4 (1964), 390.
22. Kraus, M.W. Voice-only communication enhances empathic accuracy. *American Psychologist 72*, 7 (2017), 644.
23. Lee, C.M., Narayanan, S.S., and Pieraccini, R. Combining acoustic and language information for emotion recognition. In *Proceedings of INTERSPEECH*, (Denver, CO, 2002). ISCA, 873–876.
24. Leng, Y., Xu, X., and Qi, G. Combining active learning and semi-supervised learning to construct SVM classifier. *Knowledge-Based Systems 44* (2013), 121–131.
25. Liu, J., Chen, C., Bu, J., You, M. and Tao, J. Speech emotion recognition using an enhanced co-training algorithm. In *Proceedings ICME*. (Beijing, P.R. China, 2007). IEEE, 999–1002.
26. Lotfian, R. and Busso, C. Emotion recognition using synthetic speech as neutral reference. In *Proceedings of ICASSP*. (Brisbane, Australia, 2015). IEEE, 4759–4763.
27. Mao, Q., Dong, M., Huang, Z. and Zhan, Y. Learning salient features for speech emotion recognition using convolutional neural networks. *IEEE Trans. Multimedia 16*, 8 (2014), 2203–2213.
28. Marsella, S. and Gratch, J. Computationally modeling human emotion. *Commun. ACM 57*, 12 (Dec. 2014), 56–67.
29. Picard, R.W. and Picard, R. *Affective Computing*, vol. 252. MIT Press Cambridge, MA, 1997.
30. Ram, C.S. and Ponnusamy, R. Assessment on speech emotion recognition for autism spectrum disorder children using support vector machine. *World Applied Sciences J. 34*, 1 (2016), 94–102.
31. Schmitt, M., Ringeval, F. and Schuller, B. At the border of acoustics and linguistics: Bag-of-audio-words for the recognition of emotions in speech. In *Proceedings of INTERSPEECH*. (San Francisco, CA, 2016). ISCA, 495–499.
32. Schuller, B. and Batliner, A. *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing.* Wiley, 2013.
33. Schuller, B, Mousa, A. E.-D., and Vasileios, V. Sentiment analysis and opinion mining: On optimal parameters and performances. *WIREs Data Mining and Knowledge Discovery* (2015), 5:255–5:263.
34. Soskin, W.F. and Kauffman, P.E. Judgment of emotion in word-free voice samples. *J. of Commun. 11*, 2 (1961), 73–80.
35. Stuhlsatz, A., Meyer, C., Eyben, F., Zielke, T., Meier, G. and Schuller, B. Deep neural networks for acoustic emotion recognition: Raising the benchmarks. In *Proceedings of ICASSP*. (Prague, Czech Republic, 2011). IEEE,5688–5691.
36. Tosa, N. and Nakatsu, R. Life-like communication agent-emotion sensing character 'MIC' and feeling session character 'MUSE.' In *Proceedings of the 3rd International Conference on Multimedia Computing and Systems*. (Hiroshima, Japan, 1996). IEEE, 12–19.
37. Trigeorgis, G., Ringeval, F., Brückner, R., Marchi, E., Nicolaou, M., Schuller, B. and Zafeiriou, S. Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. In *Proceedings of ICASSP*. (Shanghai, P.R. China, 2016). IEEE, 5200–5204.
38. Ververidis, D. and Kotropoulos, C. Emotional speech recognition: Resources, features, and methods. *Speech Commun. 48*, 9 (2006), 1162–1181.
39. Watson, D., Clark, L.A., and Tellegen, A. Development and validation of brief measures of positive and negative affect: the PANAS scales. *J. of Personality and Social Psychology 54*, 6 (1988), 1063.
40. Weninger, F., Eyben, F., Schuller, B.W., Mortillaro, M., and Scherer, K.R. On the acoustics of emotion in audio: What speech, music and sound have in common. *Frontiers in Psychology 4*, Article ID 292 (2013), 1–12.
41. Williamson, J. Speech analyzer for analyzing pitch or frequency perturbations in individual speech pattern to determine the emotional state of the person. U.S. Patent 4,093,821, 1978.
42. Wöllmer, M., Eyben, F., Reiter, S., Schuller, B., Cox, C., Douglas-Cowie, E. and Cowie, R. Abandoning emotion classes—Towards continuous emotion recognition with modeling of long-range dependencies. In *Proceedings of INTERSPEECH*. (Brisbane, Australia, 2008). ISCA, 597–600.
43. Zeng, Z., Pantic, M., Roisman, G.I., and Huang, T.S. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Trans. Pattern Analysis and Machine Intelligence 31*, 1 (2009), 39–58.

Björn W. Schuller (schuller@tum.de) is a professor and head of the ZD.B Chair of Embedded Intelligence for Health Care and Wellbeing at the the University of Augsburg, Germany.

Watch the author discuss his work in this exclusive *Communications* video. https://cacm.acm.org/videos/speech-emotion-recognition

P. 102
Technical Perspective
Breaking the Mold of Machine Learning
By Oren Etzioni

P. 103
Never-Ending Learning
By T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling

DOI:10.1145/3191511

To view the accompanying paper, visit doi.acm.org/10.1145/3191513 rh

Technical Perspective Breaking the Mold of Machine Learning

By Oren Etzioni

THE FIELD OF ARTIFICIAL INTELLIGENCE (AI) is rife with misnomers and machine learning (ML) is a big one. ML is a vibrant and successful subfield, but the bulk of it is simply "function approximation based on a sample." For example, the learning portion of AlphaGo—which defeated the human world champion in the game of GO—is in essence a method for approximating a non-linear function from board position to move choice, based on tens of millions of board positions labeled by the appropriate move in that position.[a] As pointed out in my *Wired* article,[4] function approximation is only a small component of a capability that would rival human learning, and might be rightfully called machine learning.

Tom Mitchell and his collaborators have been investigating how to broaden the ML field for over 20 years under headings such as multitask learning,[2] life-long learning,[7] and more. The following paper, "Never-ending Learning," is the latest and one of the most compelling incarnations of this research agenda. The paper describes the NELL system, which aims to learn to identify instances of concepts (for example, city or sports team) in Web text. It takes as input more than 500M sentences drawn from Web pages, an initial hierarchy of interrelated concepts, and small number of examples of each concept. Based on this information, and the relationships between the concepts, it is able to learn to identify millions of concept instances with high accuracy. Over time, NELL has also begun to identify relationships between concept classes, and extend its input concept set.

The NELL project is important and unique for a number of additional reasons:

The following paper describes the NELL system, which aims to learn to identify instances of concepts in Web text.

1. The system has been running at CMU for over five years, and its knowledge base is available online for inspection and download here: http://rtw.ml.cmu.edu/rtw/

2. The work is also an instance of 'Reading the Web,' a paradigm that was inspired by Mitchell's WebKB project.[3] The paradigm led to the KnowItAll system,[5] Open Information Extraction,[1] and much more.

3. The paper both places the work in context ("Learning in NELL as an approximation to EM") and identifies key lessons from the effort ("To achieve successful semi-supervised learning, couple the training of many different learning tasks.").

As is often the case with outstanding research, the work raises many open questions including:

1. Could one, with the benefit of hindsight, reimplement NELL in a radically more efficient fashion where iterations of the learning process take mere seconds?

2. What is the end-state of NELL's learning process?

3. While NELL taught us a lot about continuously running semi-supervised learning systems, it is still unable to perform increasingly challenging learning tasks over time. What are the next steps in the life-long learning paradigm?

4. More broadly, what is NELL unable to learn, and what AI architecture is necessary to go beyond these limitations?

The paper articulates both the key abstractions underlying NELL and its limitations, which suggest avenues for future work in its concluding discussion section.

In a world that has become obsessed with the latest deep neural network mechanism, and its performance on one benchmark or another, NELL is an important reminder of the power another style of research: exploratory research that seeks create new paradigms and substantially broaden the capabilities and the sophistication of machine learning systems. C

[a] Of course, this is an oversimplification but it suffices for our purposes here. See AlphaGo in *Nature*[6] for an in-depth presentation.

References

1. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M. and Etzioni, O. Open information extraction from the Web. *IJCAI*, 2007.
2. Caruana, R. Multitask learning: A knowledge-based of source of inductive bias. In *Proceedings of the 10th International Conference on Machine Learning.* (San Mateo, CA, USA, 1993). Morgan Kaufmann, 41–48.
3. Craven, M. DePasquo, D., Freitag, D., McCallum, A., Mitchell, T.M., Nigam, K. and Slattery, S. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the AAAI/IAAI*, 1998.
4. Etzioni, O. Deep learning isn't a dangerous magic genie. It's just math. *Wired* (June 15, 2016); https://www.wired.com/2016/06/deep-learning-isnt-dangerous-magic-genie-just-math/.
5. Etzioni, O., Cafarella, M.J., Downey, D. Popescu, A-M, Shaked, T. Soderland, S., Weld, D.S. and Yates, A. Unsupervised named-entity extraction from the Web: An experimental study (2005); http://bit.ly/2F5MZlf
6. Gibley, E. Google AI algorithm masters ancient game of Go. *Nature* (Jan. 27, 2016); http://www.nature.com/news/google-ai-algorithm-masters-ancient-game-of-go-1.19234.
7. Thrun, S. and Mitchell, T.M. Learning one more thing. *IJCAI*, 1995.

Oren Etzioni is Chief Executive Officer of the Allen Institute for Artificial Intelligence, Seattle, and professor of computer science at the University of Washington, Seattle, WA, USA.

DOI:10.1145/3191513

Never-Ending Learning

By T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling

Abstract

Whereas people learn many different types of knowledge from diverse experiences over many years, and become better learners over time, most current machine learning systems are much more narrow, learning just a single function or data model based on statistical analysis of a single data set. We suggest that people learn better than computers precisely because of this difference, and we suggest a key direction for machine learning research is to develop software architectures that enable intelligent agents to also learn many types of knowledge, continuously over many years, and to become better learners over time. In this paper we define more precisely this *never-ending learning* paradigm for machine learning, and we present one case study: the Never-Ending Language Learner (NELL), which achieves a number of the desired properties of a never-ending learner. NELL has been learning to read the Web 24hrs/day since January 2010, and so far has acquired a knowledge base with 120mn diverse, confidence-weighted beliefs (e.g., *servedWith(tea,biscuits)*), while learning thousands of interrelated functions that continually improve its reading competence over time. NELL has also learned to reason over its knowledge base to infer new beliefs it has not yet read from those it has, and NELL is inventing new relational predicates to extend the ontology it uses to represent beliefs. We describe the design of NELL, experimental results illustrating its behavior, and discuss both its successes and shortcomings as a case study in never-ending learning. NELL can be tracked online at http://rtw.ml.cmu.edu, and followed on Twitter at @CMUNELL.

1. INTRODUCTION

Machine learning is a highly successful branch of artificial intelligence (AI), and is now widely used for tasks from spam filtering, to speech recognition, to credit card fraud detection, to face recognition. Despite these successes, the ways in which computers learn today remain surprisingly narrow when compared to human learning. This paper explores an alternative paradigm for machine learning that more closely models the diversity, competence and cumulative nature of human learning. We call this alternative paradigm *never-ending learning*.

To illustrate, note that in each of the above machine learning applications, the computer learns only a single function to perform a single task in isolation, usually from human labeled training examples of inputs and outputs of that function. In spam filtering, for instance, training examples consist of specific emails and spam or not-spam labels for each. This style of learning is often called *supervised function approximation*, because the abstract learning problem is to approximate some unknown function $f: X \to Y$ (e.g., the spam filter) given a training set of input/output pairs $\{\langle x_i, y_i\rangle\}$ of that function. Other machine learning paradigms exist as well (e.g., unsupervised clustering, topic modeling, reinforcement learning) but these paradigms also typically acquire only a single function or data model from a single dataset.

In contrast to these paradigms for learning single functions from well organized data sets over short time-frames, humans learn many different functions (i.e., different types of knowledge) over years of accumulated diverse experience, using extensive background knowledge learned from earlier experiences to guide subsequent learning. For example, humans first learn to crawl, then to walk, run, and perhaps ride a bike. They also learn to recognize objects, to predict their motions in different circumstances, and to control those motions. Importantly, they learn *cumulatively*: as they learn one thing this new knowledge helps them to more effectively learn the next, and if they revise their beliefs about the first then this change refines the second.

The thesis of our research is that *we will never truly understand machine or human learning until we can build computer programs that, like people,*

- learn many different types of knowledge or functions,
- from years of diverse, mostly self-supervised experience,
- in a staged curricular fashion, where previously learned knowledge enables learning further types of knowledge,
- where self-reflection and the ability to formulate new representations and new learning tasks enable the learner to avoid stagnation and performance plateaus.

We refer to this learning paradigm as "never-ending learning." The contributions of this paper are to (1) define more precisely the never-ending learning paradigm, (2) present as a case study a computer program called the NELL which implements several of these capabilities, and which has been learning to read the Web 24hrs/day since January 2010, and (3) identify from NELL's strengths and weaknesses a number of key design features important to any never-ending learning system. This paper is an elaboration and extension to an earlier overview of the NELL system.[27]

2. RELATED WORK

Previous research has considered the problem of designing machine learning agents that persist over long periods

The original version of this paper appeared in the *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (Austin, TX, Jan. 25–30, 2015), 2302–2310.

of time (e.g., life long learning[38]), and that learn to learn[39] by various methods, including using previously learned knowledge from earlier tasks to improve learning of subsequent tasks.[11] Still, there remain few if any working systems that demonstrate this style of learning in practice. General architectures for problem solving and learning (e.g., SOAR Laird et al.[21], ICARUS Langley et al.[22], PRODIGY Donmez and Carbonell[15], and THEO Mitchell et al.[26]) have been applied to problems from many domains, but again none of these programs has been allowed to learn continuously for any sustained period of time. Lenat's work on Automated Mathematician (AM) and Eurisko[24] represents an attempt to build a system that invents concepts, then uses these as primitives for inventing more complex concepts, but again this system was never allowed to run for a sustained period, because the author determined it would quickly reach a plateau in its performance.

Beyond such work on integrated agent architectures, there has also been much research on individual subproblems crucial to never-ending learning. For example, work on multitask transfer learning[10] suggests mechanisms by which learning of one type of knowledge can guide learning of another type. Work on active and proactive learning[15, 40] and on exploitation/exploration trade-offs[5] presents strategies by which learning agents can collect optimal training data from their environment. Work on learning of latent representations[2, 29] provides methods that might enable never-ending learners to expand their internal knowledge representations over time, thereby avoiding plateaus in performance due to lack of adequate representations. Work on curriculum learning[3] explores potential synergies across sets or sequences of learning tasks. Theoretical characterizations of cotraining[4] and other multitask learning methods[1, 32] have provided insights into when and how the sample complexity of learning problems can be improved via multitask learning.

There is also related work on constructing large knowledge bases on the Web—the application that drives our NELL case study. The WebKB project,[13] Etzioni's early[17] and more recent[18] work on machine reading the Web, and the YAGO[37] project all represent attempts to construct a knowledge base using Web resources, as do commercial knowledge graph projects at Google, Yahoo!, Microsoft, Bloomberg, and other companies. However, unlike NELL, none of these efforts has attempted a sustained never ending learning approach to this problem.

Despite this relevant previous research, we remain in the very early stages in studying never-ending learning methods. We have almost no working systems to point to, and little understanding of how to architect a computer system that successfully learns over a prolonged period of time, while avoiding plateaus in learning due to saturation of learned knowledge. The key contributions of this paper are first, to present a working case study system, an extended version of an early prototype reported in Carlson et al.[8], which successfully integrates a number of key competencies; second, an empirical evaluation of the prototype's performance over time; and third, an analysis of the prototype's key design features and shortcomings, relative to the goal of understanding never-ending learning.

3. NEVER-ENDING LEARNING

Informally, we define a never-ending learning agent to be a system that, like humans, learns many types of knowledge, from years of diverse and primarily self-supervised experience, using previously learned knowledge to improve subsequent learning, with sufficient self-reflection to avoid plateaus in performance as it learns. The never-ending learning *problem* faced by the agent consists of a collection of learning tasks, and constraints that couple their solutions.

To be precise, we define a *never-ending learning problem* $\mathscr{L}$ to be an ordered pair consisting of: (1) a set $L = \{L_i\}$ of learning tasks, where the *i*th *learning task* $L_i = \langle T_i, P_i, E_i\rangle$ is to improve the agent's performance, as measured by *performance metric* P_i, on a given *performance task* T_i, through a given type of *experience* E_i; and (2) a set of coupling constraints $C = \{\langle\phi_k, V_k\rangle\}$ among the solutions to these learning tasks, where ϕ_k is a real-valued function over two or more learning tasks, specifying the degree of satisfaction of the constraint, and V_k is a vector of indices over learning tasks, specifying the arguments to ϕ_k.

$$\begin{aligned} \mathscr{L} &= (L, C) \qquad (1)\\ where, L &= \{\langle T_i, P_i, E_i\rangle\}\\ C &= \{\langle\phi_k, V_k\rangle\} \end{aligned}$$

Above, each performance task T_i is a pair $T_i \equiv \langle X_i, Y_i\rangle$ defining the domain and range of a function to be learned $f_i^* : X_i \rightarrow Y_i$. The performance metric $P_i : f \rightarrow \mathbb{R}$ defines the optimal learned function f_i^* for the *i*th learning task:

$$f_i^* \equiv \arg\max_{f \in F_i} P_i(f)$$

where F_i is the set of all possible functions from X_i to Y_i.

Given such a learning *problem* containing *n* learning tasks, a never-ending learning *agent* $\mathscr{A}$ outputs a sequence of solutions to these learning tasks. As time passes, the quality of these *n* learned functions should improve, as measured by the individual performance metrics $P_1 \ldots P_n$ and the degree to which the coupling constraints *C* are satisfied.

To illustrate, consider a mobile robot with sensor inputs *S* and available actions *A*. One performance task, $\langle S, A\rangle$, might be for the robot to choose actions to perform from any given state, and the corresponding learning task $\langle\langle S, A\rangle, P_1, E_1\rangle$ might be to learn the specific function $f_1 : S \rightarrow A$ that leads most quickly to a goal state defined by performance metric P_1, from training experience E_1 obtained via human teleoperation. A second performance task for the same robot may be to predict the outcome of any given action in any given state: $\langle S \times A, S\rangle$. Here, the learning task $\langle\langle S \times A, S\rangle, P_2, E_2\rangle$ might be to learn this prediction function $f_2 : S \times A \rightarrow S$ with *high accuracy* as specified by performance metric P_2, from experience E_2 consisting of the robot wandering autonomously through its environment.

Note these two robot learning tasks can be coupled by enforcing the constraint that the learned function f_1 must choose actions that do indeed lead optimally to the goal state according to the predictions of learned function f_2. By defining this coupling constraint $\phi(L_1, L_2)$ between the solutions to these two learning tasks, we give the learning agent a chance to improve its ability to learn one function by success in learning the other.

We are interested in never-ending Learning agents that address such never-ending learning problems $\mathscr{L} = (L, C)$, especially in which the learning agent

- *learns many different types of inter-related knowledge*; that is, L contains many learning tasks, coupled by many cross-task constraints,
- *from years of diverse, primarily self-supervised experience*; that is, the experiences $\{E_i\}$ on which learning is based are realistically diverse, and largely provided by the system itself,
- *in a staged, curricular fashion where previously learned knowledge supports learning subsequent knowledge*; that is, the different learning tasks $\{L_i\}$ need not be solved simultaneously—solving one helps solve the next, and
- *where self-reflection and the ability to formulate new representations, new learning tasks, and new coupling constraints enables the learner to avoid becoming stuck in performance plateaus*; that is, where the learner may itself add new learning tasks and new coupling constraints that help it address the given learning problem $\mathscr{L}$.

4. CASE STUDY: NEVER-ENDING LANGUAGE LEARNER

The Never-Ending Language Learner (NELL), an early prototype of which was reported in Carlson et al.[8], is a learning agent whose task is to learn to read the Web. The input-output specification of NELL's never-ending learning problem is:

Given:

- an initial ontology defining hundreds of categories (e.g., Sport, Athlete) and binary relations that hold between members of these categories (e.g., AthletePlaysSport(x,y)),
- approximately a dozen labeled training examples for each category and each relation (e.g., examples of Sport might include the noun phrases "baseball" and "soccer"),
- the Web : an initial 500mn Web pages from the ClueWeb 2009 collection,[7] augmented in 2017 by the addition of the ClueWeb 2012 collection[6] to form a collection of 1.233bn Web pages. In addition, Google has granted NELL access to 100,000 Google Application Program Interface (API) search queries each day.
- occasional interaction with humans (e.g., through NELL's public Website http://rtw.ml.cmu.edu);

Do: Run 24hrs/day, forever, and each day:

- read (extract) more beliefs from the Web, and remove old incorrect beliefs, to populate a growing knowledge base containing a confidence and provenance for each belief,
- learn to read better than the previous day.

NELL has been running non-stop since January 2010, each day extracting more beliefs from the Web, then retraining itself to improve its competence. The result so far is a Knowledge Base (KB) with approximately 120mn interconnected beliefs (Figure 1), along with millions of learned phrasings, morphological features, and Web page structures NELL now uses to extract beliefs from the Web. NELL is also now learning to reason over its extracted knowledge to infer new beliefs it has not yet read, and it is now able to propose extensions to its initial manually-provided ontology.

5. NELL'S NEVER-ENDING LEARNING PROBLEM

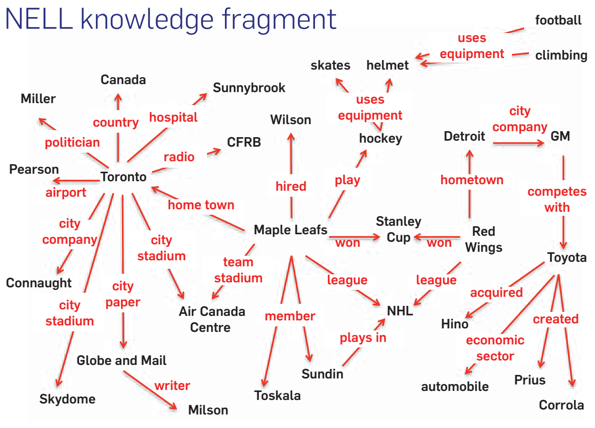
Above we described the input-output specification of the NELL system. Here we describe NELL's never-ending learning problem $\langle L, C\rangle$ in terms of the general formalism introduced in Section 2, first describing NELL's learning tasks L, then its coupling constraints C. The subsequent section describes NELL's approach to this never-ending learning problem, including NELL's mechanisms for adding its own new learning tasks and coupling constraints.

5.1. NELL's Learning Tasks

Following the notation in Equation (1), each of NELL's learning tasks consists of a performance task, performance metric, and type of experience $\langle T_i, P_i, E_i\rangle$. NELL faces over 4100 distinct learning tasks, corresponding to distinct functions $f_i : X_i \rightarrow Y_i$ it is trying to learn for its distinct performance tasks $T_i = \langle X_i, Y_i\rangle$. These tasks fall into several broad groups:

Category Classification: Functions that classify noun phrases by semantic category (e.g., a Boolean valued function that classifies whether any given noun phrase refers to a food). NELL learns different Boolean functions for each of the 293 categories in its ontology, allowing noun phrases to refer to entities in multiple semantic categories (e.g., "apple" can refer to a "Food" as well as a "Company"). For each category Y_i NELL learns at least five, and in some cases six distinct functions that predict Y_i, based on five different views of the noun phrase (different X_i's), which are:

Figure 1. Fragment of the 120mn beliefs NELL has read from the Web. Each edge represents a belief triple (e.g., play(MapleLeafs, hockey), with an associated confidence and provenance not shown here. This figure contains only correct beliefs from NELL's KB—it has many incorrect beliefs as well since NELL is still learning.

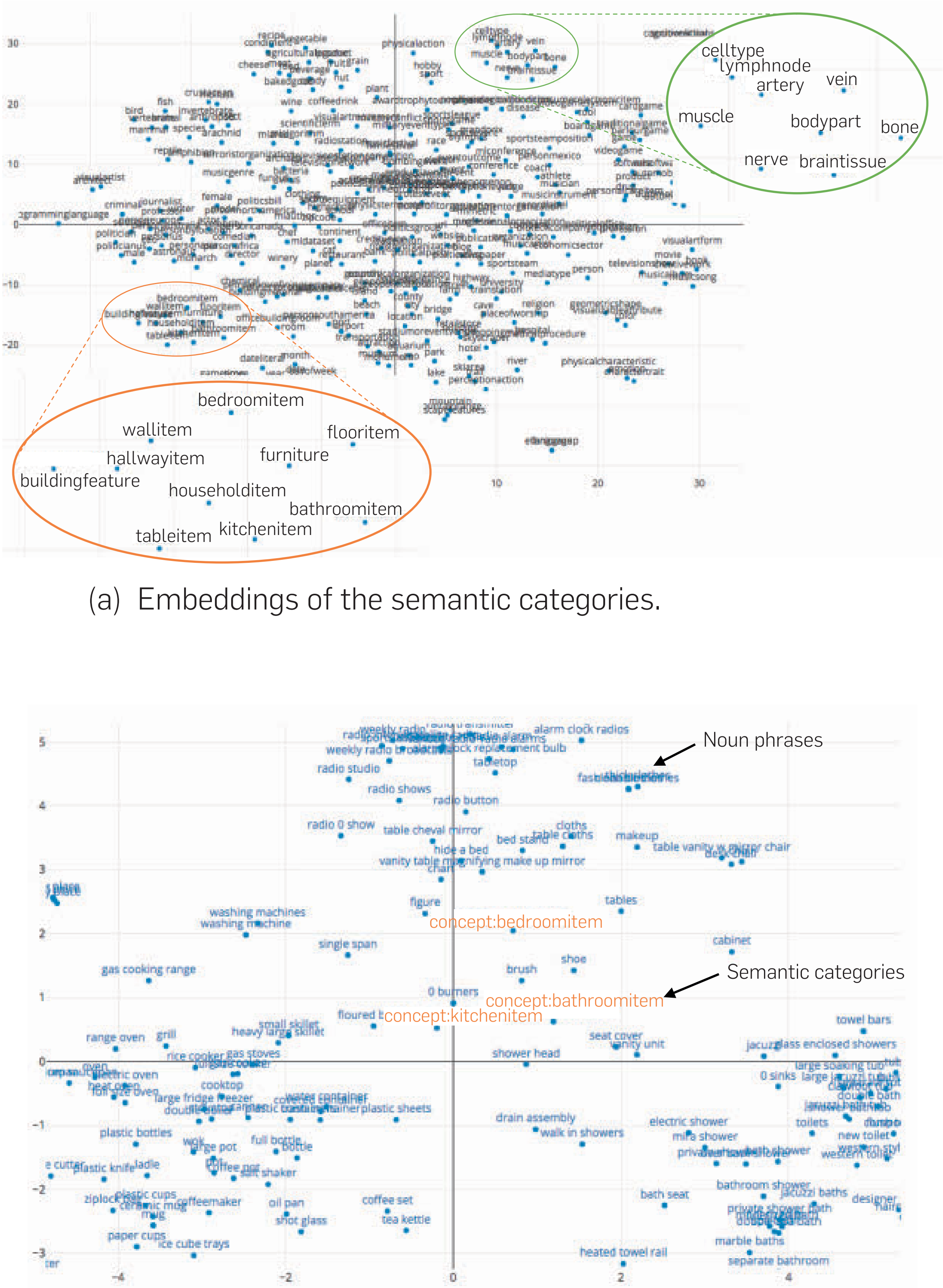


- *Character string features* of the noun phrase (e.g., whether the noun phrase ends with the character string "…burgh"). This is performed by the Coupled Morphological Classifier (CMC) system,[8] which represents the noun phrase by a vector with thousands of string features.
- *The distribution of text contexts found around this noun phrase in 1.233bn English Web pages* from the ClueWeb2009 and ClueWeb2012 text corpus (e.g., how frequently the noun phrase N occurs in the context "mayor of N"). This is performed by the Coupled Pattern Learner (CPL) system.[9]
- *The distribution of text contexts found around this noun phrase through active Web search*. This is performed by the OpenEval system,[36] which uses somewhat different context features from the above CPL system, and uses real time Web search to collect this information.
- *Hypertext Markup Language (HTML) structure of Web pages that mention the noun phrase* (e.g., whether the noun phrase is mentioned inside an HTML list, alongside other known cities). This is performed by the Set Expander for Any Language (SEAL) system.[41]
- *Visual images* associated with this noun phrase, when the noun phrase is given to an image search engine. This is performed by the Never Ending Image Learner (NEIL) system,[12] and applies only to a subset of NELL's ontology categories (e.g., not to non-visual categories such as MusicGenre).
- *Learned vector embeddings* of the noun phrases. This is performed by the Learned Embeddings (LE) module which has not been previously described, so we summarize the approach in some detail here. LE[44] learns a vector embedding for each noun phrase associated with each NELL entity, a vector embedding for each of the 293 categories in NELL's ontology, and a matrix embedding to represent the Generalizations relation that relates each NELL entity to the general categories to which it belongs. We employ a neural network architecture to learn these vector and matrix embeddings, training them on each NELL iteration to maximize their fit to the beliefs in NELL's current knowledge base. Note this knowledge base is updated on each NELL iteration in response to the combined results of all of NELL's reading and inference modules. Specifically, LE quantifies its confidence in the assertion that Generalization(X, Y) using the scoring function: $S(X_i, Y_i) = \mathbf{v}_{X_i}^T \mathbf{M} \mathbf{v}_{Y_i}$, where $\mathbf{v}_{X_i}$ and $\mathbf{v}_{Y_i}$ are d-dimensional vectors representing noun phrase X_i and NELL category Y_i respectively, and where M is a $d \times d$ matrix representing the Generalization relation. The vector embedding $\mathbf{v}_{X_i}$ is constructed by first averaging the vectors of the words in the noun phrase, then concatenating to this vector the word vector of its head noun. These vectors are initialized with pre-trained vectors for each word, obtained from Wieting et al.[42] These word vectors are then fine tuned during training, and used to produce the noun phrase vectors as described above. During training, LE minimizes a ranking loss $\max\{0, 1 - S(X_i, Y_i) + S(X_i, Y_i')\}$ for each positive training example $\langle X_i, Y_i \rangle$, paired with a negative training example $\langle X_i, Y_i' \rangle$. Positive examples are high confidence beliefs in NELL's knowledge base, and negative examples are constructed by changing the value of Y to form a belief triple which is not in NELL's knowledge base.

  Learned Embeddings obtains an top-1 accuracy of 0.88 when classifying new noun phrases into NELL's 293 categories. Figure 2 shows a visualization of the learned embeddings using t-SNE.[25] In general, the learned embeddings nicely reflect the semantics of the noun phrases and categories. Figure 2a displays the embeddings of 280 categories in NELL. We can see that semantically similar categories tend to be close to each other. For example, there is a cluster about body parts (on the top) and a cluster about room items (in the bottom). Figure 2b further shows three specific room-item categories and the noun phrases surrounding them. We can see that items belonging to kitchens, bedrooms, and bathrooms are generally well separated. We also find that items that can belong to multiple categories tend to locate on the boundaries. For example, "brush" and "shoe" could be both a bedroom item and a bathroom item.

- *Relation Classification*: These functions classify pairs of noun phrases by whether or not they satisfy a given relation (e.g., classifying whether the pair ⟨"Pittsburgh," "U.S."⟩ satisfies the relation "CityLocatedInCountry(x,y)"). NELL learns distinct boolean-valued classification functions for each of the 461 relations in its ontology. For each relation, NELL learns four distinct classification functions based on different feature views of the input noun phrase pair. Specifically, it uses the two classification methods CPL and OpenEval based on the distribution of text contexts found between the two noun phrases on Web pages, and it uses the SEAL classification method based on HTML structure of Web pages. These methods are described above in the list of Category Classification methods. NELL furthermore uses the LE module described above to learn to predict relation instances from learned vector embeddings of noun phrases and learned matrix embeddings for relations. Above we described the algorithm used by LE to learn vector embeddings for NELL entities and for NELL categories, and to learn a matrix embedding for the $\langle x_1$, Generalization, $x_2 \rangle$ relation. The same algorithm is used to learn a matrix embedding M_r for each NELL relation r. As in the case of the Generalization relation, LE then assigns a confidence score $S(\langle x_1, r, x_2 \rangle)$ to each possible relation triple according to the formula $S(\langle x_1, r, x_2 \rangle) = \mathbf{v}_{x_1}^T \mathbf{M}_r \mathbf{v}_{x_2}$, where $\mathbf{v}_{x_1}$ and $\mathbf{v}_{x_2}$ represent the learned vector embeddings for x_1 and x_2, and where $\mathbf{M_r}$ is the learned matrix embedding for relation r. In general we find LE's inferences about relations other than Generalization are less accurate than for the Generalization relation.

Figure 2. t-SNE visualization of the embeddings learned by LE.



(a) Embeddings of the semantic categories.

(b) Embeddings of the noun phrases and semantic categories.

This may be due in part to the smaller number of training examples for these relations, and may in part be due to the greater suitability for our approach to semantic category assignment (i.e., predicting the Generalization) compared to predicting other relations such as PersonFoundedCompany().

- *Entity Resolution*: Functions that classify whether pairs of noun phrases are synonyms. NELL's knowledge base represents noun phrases as distinct from the entities to which they can refer. This is essential because polysemous words can refer to multiple types of entities (e.g., the word "coach" can refer to a type of "person," or a type of "vehicle"), and because synonymous words can refer to the same entity (e.g., "NYC," "New York City" and "Big Apple" are synonyms for the same entity). In order to deal with polysemy, NELL simply allows a noun phrase to be classified into multiple categories if there is strong evidence according to its reading methods. To deal with synonymy, NELL learns explicit functions that classify noun phrase pairs by whether or not they are synonyms (e.g., whether "NYC" and "Big Apple" can refer to the same entity). This classification method is described in Krishnamurthy and Mitchell.[20] For each of NELL's 293 categories, it co-trains two synonym classifiers. One classifier is based on string similarity between the two noun phrases (e.g., "NYC" and "New York City" have similar string features). The second is based on similarities in the beliefs NELL has extracted (e.g., if NELL's KB believes that "NYC" and "New York City" have the same mayor, this is evidence that the assumed two city names may be synonyms, though belonging to the same country might not be evidence that two city names are synonyms). NELL learns for each of its categories (e.g., "city"), what are the category specific types of knowledge that are evidence of synonymy, and which types of string features indicate synonymy.
- *Inference Rules among belief triples*: Functions that map from NELL's current KB, to new beliefs it should add to its KB. For each relation in NELL's ontology, the corresponding function is represented by a collection of restricted Horn Clause rules learned by the Path Ranking Algorithm (PRA) system.[19, 23]

Each of the above functions $f : X \rightarrow Y$ represents a performance task $T_i = \langle X, Y\rangle$ for NELL, and each maps to the learning task of acquiring that function, given some type of experience E_i and a performance metric P_i to be optimized during learning. In NELL, the performance metric P_i to optimize is simply the *accuracy* of the learned function. In all cases except one, the experience E_i is a combination of *human-labeled training examples* (the dozen or so labeled examples provided for each category and relation in NELL's ontology, plus labeled examples contributed over time through NELL's Website), a set of *NELL self-labeled training examples* corresponding to NELL's current knowledge base, and a huge volume of unlabeled Web text. The one exception is learning over visual images, which is handled by the NEIL system with its own training procedures.

5.2. NELL's Coupling Constraints

The second component of NELL's never-ending learning task is the set of *coupling constraints* which link its learning tasks. NELL's coupling constraints fall into five groups. We describe them below as hard logical constraints. However, NELL uses these primarily as soft constraints that can be violated at some penalty cost.

- *Multi-view co-training coupling*. NELL's multiple methods for classifying noun phrases into categories (and noun phrase pairs into relations) provide a natural co-training setting,[4] in which alternative classifiers for the same category should agree on the predicted label whenever they are given the same input, even though their predictions are based on different noun phrase features. To be precise, let $v_k(z)$ be the feature vector used by the kth function, when considering input noun phrase z. For any pair of functions $f_i : v_i(Z) \rightarrow Y$ and $f_j : v_j(Z) \rightarrow Y$ that predict the same Y from the same Z using the two different feature views v_i and v_j, NELL uses the coupling constraint $(\forall z) f_i(z) = f_j(z)$. This couples the tasks of learning f_i and f_j.
- *Subset/superset coupling*. When a new category is added to NELL's ontology, the categories which are its immediate parents (supersets) are specified (e.g., "Beverage" is declared to be a subset of "Food."). When category $C1$ is added as a subset of category $C2$, NELL uses the coupling constraint that $(\forall x) C1(x) \rightarrow C2(x)$. This couples learning tasks that learn to predict $C1$ to those that learn to predict $C2$.
- *Multi-label mutual exclusion coupling*. When a category C is added to NELL's ontology, the categories that are known to be disjoint from (mutually exclusive with) C are specified (e.g., "Beverage" is declared to be mutually exclusive with "Emotion," "City," etc.). These mutual exclusion constraints are typically inherited from more general classes, but can be overridden by explicit assertions. When category $C1$ is declared to be mutually exclusive with $C2$, NELL adopts the constraint that $(\forall x) C1(x) \rightarrow \neg C2(x)$.
- *Coupling relations to their argument types*. When a relation is added to NELL's ontology, the types of its arguments must be defined in terms of NELL categories (e.g., "zooInCity(x,y)" requires arguments of types "Zoo" and "City" respectively). NELL uses these argument type declarations as coupling constraints between its category and relation classifiers.
- *Horn clause coupling*. Whenever NELL learns a Horn clause rule to infer new KB beliefs from existing beliefs, that rule serves as a coupling constraint to augment NELL's never ending learning problem $\langle L, C\rangle$. For example, when NELL learns a rule of the form $(\forall x, y, z) R_1(x, y) \wedge R_2(y, z) \rightarrow R_3(x, z)$ with probability p, this rule serves as a new probabilistic coupling constraint over the functions that learn relations R_1, R_2, and R_3. Each learned Horn clause requires that

learned functions mapping from noun phrase pairs to relations labels for R_1, R_2, and R_3 are consistent with this Horn clause; hence, they are analogous to NELL's subset/superset coupling constraints, which require that functions mapping from noun phrases to category labels should be consistent with the subset/superset constraint.

NELL's never ending learning problem thus contains over 4100 learning tasks, inter-related by over a million coupling constraints. In fact, NELL's never ending learning problem $\langle L, C\rangle$ is open ended, in that NELL has the ability to add both new consistency constraints in the form of learned Horn clauses (as discussed above) and new learning tasks, by inventing new predicates for its ontology (as discussed below).

6. NELL'S LEARNING METHODS AND ARCHITECTURE

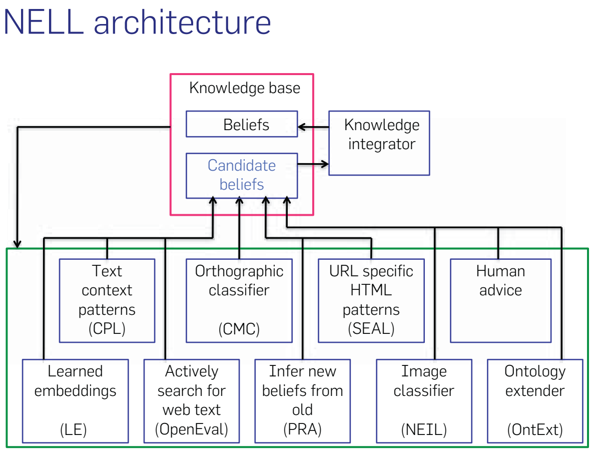
The software architecture for NELL, depicted in Figure 3, includes a KB which acts as a blackboard through which NELL's various learning and inference modules communicate.[a] As shown in Figure 3, these software modules map closely to the learning methods (CPL, CMC, SEAL, OpenEval, PRA, and NEIL) for the different types of functions mentioned in the previous section, so that NELL's various learning tasks are partitioned across these modules.

6.1. Learning in NELL as an Approximation to EM

NELL is in an infinite loop analogous to an Expectation-Maximization (EM) algorithm[14, 30] for semi-supervised learning, performing an E-like step and an M-like step on each iteration through the loop. During the E-like step, the set of beliefs that form the knowledge base is re-estimated; that is, each reading and inference module in NELL proposes updates to the KB (additions and deletions of specific beliefs, with specific confidences and provenance information). The Knowledge Integrator (KI) both records these individual recommendations and makes a final decision about the confidence assigned to each potential belief in the KB. Then, during the M-like step, this refined KB is used to retrain each of these reading and inference modules, employing module-specific learning algorithms for each. The result is a large-scale coupled training system in which thousands of learning tasks are guided by one another's results, through the shared KB and coupling constraints.

Notice that a full EM algorithm is impractical in NELL's case; NELL routinely considers tens of millions of noun phrases, yielding 10^{17} potential relational assertions among noun phrase pairs. It is impractical to estimate the probability of each of these potential latent assertions on each E-like step. Instead, NELL constructs and considers only the beliefs in which it has highest confidence, limiting each software module to suggest only a bounded number of new candidate beliefs for any given predicate on any given iteration. This enables NELL to operate tractably, while retaining the ability to add millions of new beliefs over many iterations, and to delete beliefs in which it subsequently loses confidence. In addition, NELL enforces its consistency constraints in a limited-radius fashion on each iteration (i.e., if one belief changes, the only other beliefs influenced are those coupled directly by some constraint; compositions of constraints are not considered). However, across multiple iterations of its EM-like algorithm, the influence of any given belief update can propagate throughout the knowledge graph as neighboring beliefs are themselves revised.

Figure 3. NELL's software architecture. NELL's growing knowledge base (red box) serves as a shared blackboard through which its various reading and inference modules (green box) interact. On each NELL iteration the knowledge base is first updated by integrating proposals from the various reading and inference modules. The revised knowledge base is then used to retrain each of these modules.


NELL architecture
Knowledge base
Beliefs
Candidate beliefs
Knowledge integrator
Text context patterns (CPL)
Orthographic classifier (CMC)
URL specific HTML patterns (SEAL)
Human advice
Learned embeddings (LE)
Actively search for web text (OpenEval)
Infer new beliefs from old (PRA)
Image classifier (NEIL)
Ontology extender (OntExt)

6.2. Knowledge Integrator in NELL

The KI integrates the incoming proposals for KB updates. For efficiency, the KI considers only moderate-confidence candidate beliefs, and re-assesses confidence using a limited radius subgraph of the full graph of consistency constraints and beliefs. As an example, for each new relational triple that the KI asserts, it checks that the entities in the relational triple have a category type consistent with the relation, but does not consider using new triples as a trigger to update beliefs about these argument types during the same iteration. Over multiple iterations, the effects of constraints propagate more widely through this graph of beliefs and constraints. In Pujara et al.[35] a more effective algorithm is proposed for the joint inference problem faced by NELL's KI; we believe it will be helpful to upgrade NELL's KI in the future to use this approach.

6.3. Adding Learning Tasks and Ontology Extension in NELL

NELL has the ability to extend its ontology by inventing new relational predicates using the OntExt system.[28]

[a] The KB is implemented as a frame-based knowledge representation which represents language tokens (e.g., NounPhrase:bank) distinct from non-linguistic entities to which they can refer (e.g., Company:bank, LandscapeFeature:bank), and relates the two by separate CanReferTo(noun phrase, entity) assertions.

OntExt searches for new relations by considering every pair of categories in NELL's current ontology, to search for evidence of a new, frequently discussed relation between members of that category pair. It performs this search in a three step process: (1) Extract sentences mentioning known instances of both categories (e.g., for the category pair ⟨*drug,disease*⟩ the sentence *Prozac may cause migraines* might be extracted if *prozac* and *migraines* were already present in NELL's KB). (2) From the extracted sentences, build a context by context co-occurrence matrix, then cluster the related contexts together. Each cluster corresponds to a possible new relation between the two input category instances. (3) Employ a trained classifier, and a final stage of manual filtering, before allowing the new relation (e.g., DrugHasSideEffect(x,y)) to be added to NELL's ontology. OntExt has added 62 new relations to NELL's ontology; a sample is shown in Figure 4. Note the invention and introduction of each new relation into NELL's ontology spawns a number of new tasks. These include new "learning to read" tasks, to classify which noun phrase pairs satisfy the relation, based on different views of the noun phrase pair. Each new relation also spawns a new task of learning Horn clause rules to infer this new relation from others, and of course the new relation also becomes available for representing new rules that infer instances of other NELL relations.

In addition to the OntExt algorithm for proposing new relations, we have more recently developed the verb knowledge base (VerbKB[b]) which proposes new relations on a much larger scale.[43] Verbs and verb phrases naturally express relations between noun phrases, and can provide the high coverage vocabulary of relation predicates required to represent beliefs in arbitrary text. VerbKB groups semantically similar verb patterns by analyzing the statistics of all subject, verb lexeme (plus preposition where available), and object triples (e.g., <*horse, eat, hay*>, <*john, eat with, fork*>) found by parsing the 500 mn English Web pages in NELL's initial cache of Web pages from ClueWeb2009.

VerbKB is guided by NELL's knowledge about the semantic categories to which the subject and object belong. The groups of <*subjectCategory verbCluster, objectCategory*> patterns discovered by VerbKB are proposed as new typed relations for NELL. For example, VerbKB has proposed that the group of verbs {have, experience, suffer, survive, sustain, bear, endure, tolerate} represents a potential new NELL relation PersonHaveDisease(person,disease), when these verbs occur with a subject belonging to the NELL category "Person" and an object of NELL category "Disease." VerbKB has clustered 65,000 verb lexemes (+prepositions) which cover 98% of all verb mentions in ClueWeb2010 and has proposed a collection of 86,000 verb clusters (58,000 of which are non-singleton clusters) as new relations to NELL. Because this very large number of proposed relations raises scaling issues for NELL's current hardware and software, we are currently exploring ways to scale up NELL, and ways to select among these proposed relations by relying on NELL's Twitter interface[c] followers to decide (as described in Pedro and Hruschka[31]) which among these relations will be most interesting for NELL to learn. Although we are still working to incorporate this into routine use by NELL's never-ending execution run, we are optimistic that this will provide a significant increase in the coverage and capability of NELL's learned knowledge.

Figure 4. Sample of relations automatically discovered by NELL's OntExt algorithm. When NELL adds a newly discovered relation to its ontology, its learning algorithms are automatically triggered to seek new instances. For example, since adding these new relations NELL has added hundreds of instances of buildingFeatureMadeFromMaterial including (tiles, porcelain) and (garage doors, steel), and thousands of instances of clothingGoesWithClothing including (tee shirt, jeans), (tuxedo jacket, tie) and (gloves, warm coat).

Sample of self-discovered NELL relations

- athleteWonAward
- animalEatsFood
- languageTaughtInCity
- clothingMadeFromPlant
- beverageServedWithFood
- fishServedWithFood
- athleteBeatAthlete
- athleteInjuredBodyPart
- arthropodFeedsOnInsect
- animalEatsVegetable
- plantRepresentsEmotion
- foodDecreasesRiskOfDisease
- clothingGoesWithClothing
- bacteriaCausesPhysCondition
- buildingFeatureMadeFromMaterial
- emotionAssociatedWithDisease
- foodCanCauseDisease
- agriculturalProductAttractsInsect
- arteryArisesFromArtery
- countryHasSportsFans
- bakedGoodServedWithBeverage
- beverageContainsProtein
- animalCanDevelopDisease
- beverageMadeFromBeverage

6.4. Self-Reflection and Self-Evaluation

One important capability we wish to add to NELL is the ability to self-reflect on, and self-evaluate its own performance, to enable it to focus its learning efforts where it most needs improvement. Although NELL's architecture does not yet have such a self-reflection component, we have recently developed and tested the key algorithms that will enable it to estimate the accuracies of thousands of functions it is learning, based solely on the unlabeled data it has access to. The key theoretical question here is "under what conditions can unlabeled data be used to estimate accuracy of learned functions?" Surprisingly, we have found that there are conditions under which the observed *consistency* among different learned functions applied to unlabeled data can be used to derive highly precise estimates of *accuracies* of these functions, and that these methods work well for accuracy estimation in NELL.

For example, in Platanios et al.[32] we show that if one has three or more approximations to the same function (e.g., NELL's different learned classifiers that predict whether a noun phase refers to a city, based on different views of the noun phrase), if these functions are more accurate than chance, and if their errors are independent, then the rates at which these functions agree on the classification of unlabeled examples can be used to solve exactly for their accuracies. While NELL comes close to meeting

[b] http://gourierverb.azurewebsites.net.

[c] https://twitter.com/cmunell.

these conditions, in general it does not satisfy the assumption that its different functions make completely independent errors. However, we found it possible to weaken the assumption of independent errors, and effectively replace it by a prior stating that more independent errors are more probable. Experimental results show that these algorithms, run on NELL's learned functions for 15 representative categories, yield accuracy estimates that deviate on average less than 0.01 from the true accuracies. In Platanios et al.[33] we introduce a related Bayesian approach which also leverages the fact that NELL is learning to predict many different functions for each input noun phrase, hence leveraging the full mulit-view, multi-task nature of NELL's learning problem. Finally, in Platanios et al.[34] we also propose a probabilistic logic approach that further leverages the information provided by logical constraints between the outputs of the functions that NELL is learning to predict (e.g., a noun phrase that refers to a city has to also refer to a location).

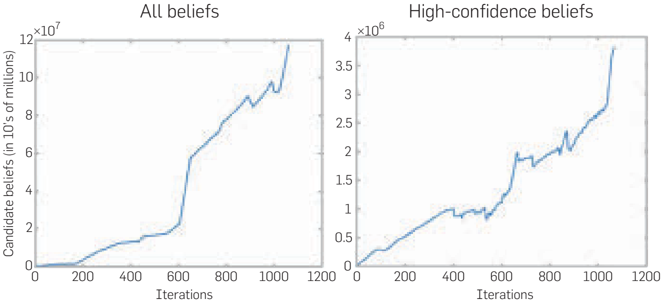
7. EMPIRICAL EVALUATION

Our primary goal in experimentally evaluating NELL is to understand the degree to which NELL improves over time through learning, both in its reading competence, and in the size and quality of its KB.

First, consider the growth of NELL's KB over time, from its inception in January 2010 through July 10, 2017, during which NELL completed 1064 iterations. The left panel of Figure 5 shows the number of beliefs in NELL's KB over time, and the right panel of Figure 5 shows the number of beliefs for which NELL holds high confidence. Note that as of July 2017, NELL's KB contains approximately 117mn beliefs with varying levels of confidence, including 3.81mn that it holds in high confidence. Here, "high confidence" indicates either that one of NELL's modules assigns a confidence of at least 0.9 to the belief, or that multiple modules independently propose the belief.

As Figure 5 illustrates, NELL's KB is clearly growing, though its high confidence beliefs constitute only about 3% of the total set of beliefs it is considering. Although NELL has now saturated some of the categories and relations in its ontology (e.g., for the category "Country" it extracted most actual country names during the first few hundred iterations), the knowledge base nevertheless continues to grow overall. This ongoing growth is due in part to the fact that NELL's ontology extension module is adding new predicates to the ontology over time (e.g., *athleteInjuredBodyPart(athlete, bodyPart)*), creating the opportunity for NELL to acquire new beliefs that it could not even represent in its original ontology.

Figure 5. NELL KB size over time. Total number of beliefs (left) and number of high confidence beliefs (right) versus iterations. Left plot vertical axis is tens of millions, right plot vertical axis is in millions. The horizontal axis covers NELL iterations from January 2010 until July 2017.



Beyond the volume of beliefs, consider the accuracy of NELL's reading competence over time. To evaluate this, we applied different versions of NELL obtained at different iterations in its history, to extract beliefs from its cache of English Web pages, plus the world wide Web as accessed through NELL's reading modules. We then manually evaluated the accuracy of the beliefs extracted by these different historical versions of NELL, to measure NELL's evolving reading competence. To obtain different versions of NELL over time, we relied on the fact that NELL's state at any given time is fully determined by its KB. In particular, given NELL's KB at iteration *i* we first had NELL train itself on that KB plus unlabeled text from the Web, then had it apply its trained methods to a test set of unlabeled Web text to propose a rank-ordered set of confidence-weighted beliefs. We evaluated the accuracy of these beliefs to measure NELL's evolving competence at different points in time.[d]

In greater detail, we first selected 12 different points in time to test NELL's reading competence: iterations 166, 261, 337, 447, 490, 561, 641, 731, 791, 886, 960, and 1026. These iterations span from the inception of NELL in January 2010 through November 2016. For each of those iterations, we trained NELL using the KB from that iteration, then evaluated its reading competence over a representative sample of 18 categories and 13 relations (31 predicates in total) from NELL's initial ontology. Each iteration-specific trained version of NELL was then applied to produce a ranked list of the top 1000 *novel* predictions, omitting any prediction corresponding to a noun phrase or relation instance for which NELL had received human feedback at any point in its history. To estimate NELL's reading competence at each point we first created a pool of test instances to manually annotate. For each iteration to be evaluated, this pool included the top 10 ranked predictions for each predicate, 20 more predictions sampled uniformly at random from ranks 11 to 100, and an additional 20 from ranks 101 to 1000. This provided 50 (potentially overlapping) instances per predicate from each iteration, averaging about 350 instances per predicate over all iterations. We manually annotated each of these instances as correct or incorrect, yielding approximately 11,000 total annotated beliefs regarding 31 predicates, which we used to evaluate NELL's learned reading competence at each iteration.

The results of this evaluation are summarized in Figure 6, which shows the improvement in NELL's reading competence over time, as measured by NELL's estimated Mean

[d] To be precise, for NELL iterations prior to and including November 2014, we used test data from the Web as of November 2014. For evaluations of NELL's competence in November 2015 and November 2016, we instead used the Web as it existed on those dates.

Average Precision (MAP) over this sample of 1000 most confident predictions for each of these 31 predicates. Taken together, the results in the Figure 6 and the results of Figure 5 show that over several years NELL's reading accuracy, and the accuracy of its most confident beliefs have grown at the same time that the volume of beliefs in the knowledge base has also grown by millions.

Next, we summarize feedback from humans to NELL. This feedback is nearly all negative feedback identifying NELL's incorrect beliefs. Figure 7 shows the distribution of this negative feedback from humans to NELL over its first 802 iterations, which is very similar to the distribution of feedback in more recent iterations. During this period, NELL received on average 2.4 negative feedback labels per predicate, per month, for a total of 85,088 items of negative feedback (an average of 1,467 per month). Note the large burst of feedback from iteration 100 to 177. During the first two years, the bulk of feedback was provided by members of the NELL research project, though in more recent years most of the feedback is now crowdsourced, that is, provided by external visitors to the NELL Website, or by followers of @CMUNELL on Twitter.

Figure 6. Evolution of NELL reading accuracy over time. The vertical axis shows the estimated Mean Average Precision over the 1000 most confident predictions for a representative sample of 18 categories and 13 relations in NELL's ontology. The horizontal axis represents NELL iterations from January 2010 through November 2016.

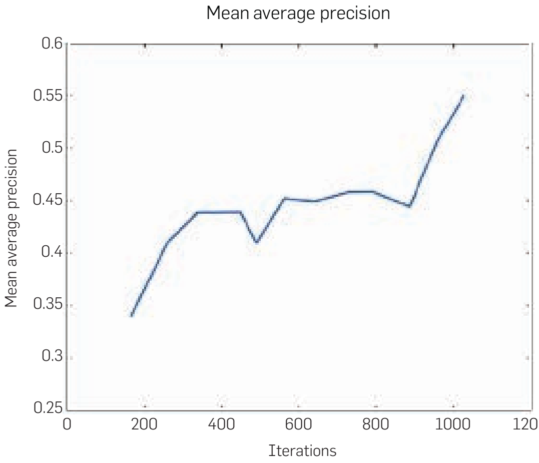
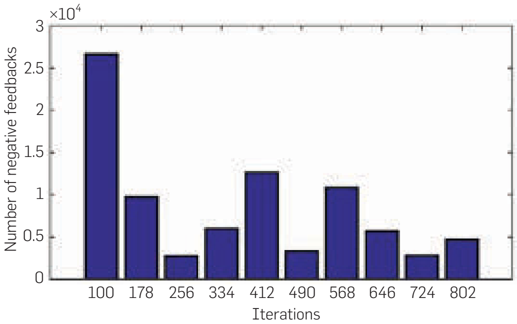


Figure 7. Human feedback to NELL over time. Each bar in this histogram shows the number of NELL beliefs for which humans provided negative feedback, during a 78 iteration interval. This averages out to 2.4 items of feedback per month, per predicate in NELL's ontology. Human input to NELL has been dropping over time, even as its reading accuracy has been increasing.



In addition to the above aggregate measures of NELL's behavior, it is interesting to consider its detailed behavior for specific predicates. Here we find that NELL's performance varies dramatically across predicates: the precision over NELL's 1000 highest confidence predictions for categories such as "river," "body part," and "physiological condition" is well above 0.95, whereas for "machine learning author" and "city capital of country(x,y)" accuracies are well below 0.5. One factor influencing NELL's ability to learn well is whether it has other mutually exclusive categories to learn—this mutually exclusive relationship provides a coupling constraint that typically yields valuable negative examples. For instance, many of NELL's errors for the category "machine learning author" are computer science researchers (e.g., "Robert Kraut") who do not happen to work in the area of machine learning—NELL would presumably learn this category better if we added to its ontology other categories such as "HCI author" to provide examples that are usually mutually exclusive. Another factor is the number of actual members of the category: for example, the category "planet" has only a small number of actual members, but NELL is searching for more, so it proposes members such as "counter earth" and "asteroid ida." In some cases, NELL performs poorly for a predicate due to a particular error which propagates due to its bootstrap-style learning from unlabeled or self-labeled data. For example, for the category "sports team position" NELL has numerous correct members such as "quarterback" and "first base," but it has acquired a systematic error in having a strong belief that phrases ending with "layer" (e.g., "defence layer" and "cloud layer") refer to sports positions. While some do, most do not, yet NELL has no easy way to determine this.

It is important to realize that as NELL progresses, the task of adding the next new belief to the knowledge base naturally becomes more difficult. NELL's redundancy-based reading methods tend to extract the most frequently-mentioned beliefs earlier (e.g., for the category "emotions" NELL first extracted frequently mentioned emotions such as "gladness" and "loneliness"). But once it has extracted the frequently mentioned instances which are easiest for its statistically-based methods, it later can only grow the KB by extracting less frequently mentioned beliefs (e.g., later the emotions it was able to add were more obscure instances such as "incredible lightness," "cavilingness," and "nonop-probriousness," as well as some non-emotion phrases).

This increasing difficulty over time seems to be inherent to the task of never-ending learning. Meeting this challenge in NELL suggests several opportunities for future research: (1) add a self-reflection capability to NELL to enable it to detect where it is doing well, where it is doing poorly, when it has sufficiently populated any given category or relation, enabling it to allocate its efforts in a more intelligently targeted fashion, (2) broaden the scope of data NELL uses to extract beliefs, for example by including languages beyond English,[16] image data as well as text, and new continuous streams of data such as Twitter, (3) expand NELL's ontology dramatically, both by relying more heavily on automated algorithms for inventing new relations and categories, and by merging other open-source ontologies such as DBpedia into NELL's ontology, and (4) add a new generation of "micro-reading" methods to NELL—methods that perform deep semantic analysis of individual sentences and text passages, and which therefore do not need to rely on redundancy across the Web to achieve accurate reading. We are currently actively exploring each of these directions.

8. DISCUSSION

Based on the above empirical analysis, it is clear that NELL is successfully learning to improve its reading competence over time, and is using this increasing competence to build an ever larger KB of beliefs about the world. In this paper, we present NELL as an early case study of a never-ending learning system. What are the lessons to be learned from this case study? Our experience with NELL suggests four useful design features that have led to the successes it has had—design features we recommend for any never-ending learning system:

To achieve successful semi-supervised learning, couple the training of many different learning tasks. The primary reason NELL has succeeded in learning thousands of functions from only a small amount of supervision is that it has been designed to simultaneously learn thousands of different functions that are densely connected by a large number of coupling constraints. As progress begins to be made on one of these learning tasks, the coupling constraints allow the learned information to constrain subsequent learning for other tasks.

Allow the agent to learn additional coupling constraints. Given the critical importance of coupling the training of many functions, great gains can be had by automatically learning additional coupling constraints. In NELL, this is accomplished by learning restricted-form probabilistic Horn clauses by data-mining NELL's KB. NELL has learned hundreds of thousands of probabilistic Horn clauses and related probabilistic inference rules which it uses to infer new KB beliefs it has not yet read. As a side effect of creating new beliefs which are subsequently used to retrain NELL's reading functions, these Horn clauses also act as coupling constraints to further constrain and guide subsequent joint learning of NELL's reading functions for relations mentioned by the Horn clause.

Learn new representations that cover relevant phenomena beyond the initial representation. To continuously improve, and to avoid reaching a plateau in performance, a never-ending learning system may need to extend its representation beyond what is initially provided. NELL has a primitive but already-useful ability to extend its representation by suggesting new relational predicates (e.g., RiverFlowsThroughCity(x,y)) between existing categories (e.g., river, city). Each new relation NELL introduces leads to new learning tasks such as learning to extract the relation from text, and learning to infer instances of the relation from other beliefs.

Organize the set of learning tasks into an easy-to-increasingly-difficult curriculum. Given a complex set of learning tasks, it will often be the case that some learning tasks are easier, and some produce prerequisite knowledge for others. In NELL, we have evolved the system by manually introducing new types of learning tasks over time. During NELL's first six months, its only tasks were to classify noun phrases into categories, and noun phrase pairs into relations. Later, once it achieved some level of competence at these, and grew its KB accordingly, it became feasible for it to confront more challenging tasks. At that point, we introduced the task of datamining the KB to discover useful Horn clause rules, as well as the task of discovering new relational predicates based on NELL's knowledge of category instances. A key open research question is how the learning agent might itself evolve a useful curriculum of learning tasks.

NELL also has many limitations, which suggest additional areas for research into never-ending learning agents:

- *Self reflection and an explicit agenda of learning subgoals*. At present, NELL suffers from the fact that it has a very weak ability to monitor its own performance and progress. It does not notice, for example, that it has learned no useful new members of the "country" category for the past year, and it continues to work on this problem although its knowledge in this area is saturated. Furthermore, it makes no attempt to allocate its learning effort to tasks that will be especially productive (e.g., collecting new Web text describing entities about which it has only low confidence beliefs). It is clear that developing a self-reflection capability to monitor and estimate its own accuracy, and to plan specific learning actions in response to perceived needs, would allow the system to use its computational effort more productively.
- *Pervasive plasticity*. Although NELL is able to modify many aspects of its behavior through learning, other parts of its behavior are cast in stone, unmodifiable. For example, NELL's method for detecting noun phrases in text is a fixed procedure not open to learning. In designing never-ending learning agents, it will be important to understand how to architect the agent so that as many aspects of its behavior as possible are plastic—that is, open to learning. Otherwise, the agent runs the risk of reaching a performance plateau in which further improvement requires modifications to a part of the system that is not itself modifiable.
- *Representation and reasoning*. At present, NELL uses a simple frame based knowledge representation, augmented by the PRA reasoning system which performs tractable but limited types of reasoning based on restricted Horn clauses. NELL's competence is already limited in part by its lack of more powerful reasoning

components; for example, it currently lacks methods for representing and reasoning about time and space. Hence, core AI problems of representation and tractable reasoning are also core research problems for never-ending learning agents. In addition, recent research in natural language has shown that working wth non-symbolic vector embeddings of words, phrases and entities, learned via deep neural networks, has many advantages. In NELL, the recent addition of the LE method has similarly yielded improvements in NELL's ability to extract new instances of categories and relations. However, an even more dramatic adoption of vector embeddings learned via deep networks would be possible, for example, providing a continuous space of category and relation predicates represented by vectors and matrices, fundamentally changing the framing of the ontology extension problem (i.e., if every relation is represented by a matrix, the set of possible matrices *is* the set of possible relations in the ontology).

The study of never-ending learning raises important conceptual and theoretical problems as well, including:

- *The relationship between* consistency *and* correctness. An autonomous learning agent can never truly perceive whether it is correct—it can at best detect only that it is internally consistent. For example, even if it observes that its predictions (e.g., new beliefs predicted by NELL's learned Horn clauses) are consistent with what it perceives (e.g., what NELL reads from text), it cannot distinguish whether that observed *consistency* is due to correct predictions and correct perceptions, or incorrect predictions and correspondingly incorrect perceptions. This is important in understanding never-ending learning, because it suggests organizing the learning agent to become increasingly consistent over time, which is precisely how NELL uses its consistency constraints to guide learning. A key open theoretical question therefore is "under what conditions can one guarantee that an increasingly consistent learning agent is also an increasingly correct agent?" Platanios et al.[32] provides one step in this direction, by providing an approach that will soon allow NELL to estimate its accuracy based on the observed consistency rate among its learned functions, but much remains to be understood about this fundamental theoretical question.
- *Convergence guarantees in principle and in practice*. A second fundamental question for never-ending learning agents is "what agent architecture is sufficient to guarantee that the agent can in principle generate a sequence of self-modifications that will transform it from its initial state to an increasingly high performance agent, without hitting performance plateaus?" Note this may require that the architecture support pervasive plasticity, the ability to change its representations, etcetera. One issue here is whether the architecture has sufficient self-modification operations to allow it to produce ever-improving modifications to itself *in principle*. A second, related issue is whether its learning mechanisms will make these potential changes, converging *in practice* given a tractable amount of computation and training experience.

Acknowledgment

We thank the anonymous reviewers for their constructive comments, and thank Lucas Navarro, Bill McDowell, Oscar Romero and Amos Azaria for help in the empirical evaluation of NELL. This research was supported in part by DARPA under contract number FA8750-13-2-0005, by NSF grants IIS-1065251 and CCF-1116892, by AFOSR award FA9550-17-1-0218, and by several generous gifts from Google and Yahoo!. We also gratefully acknowledge graduate fellowship support over the years from Yahoo, Microsoft, Google, Fulbright, CAPES, and FAPESP.

References

1. Balcan, M.-F., Blum, A. A PAC-style model for learning from labeled and unlabeled data. *Proc. of COLT* (2004).
2. Bengio, Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning 2*, 1 (2009), 1–127.
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (2009), ACM, 41–48.
4. Blum, A., Mitchell, T. Combining labeled and unlabeled data with co-training. *Proc. of COLT* (1998).
5. Brunskill, E., Leffler, B., Li, L., Littman, M.L., Roy, N. Corl: A continuous-state offset-dynamics reinforcement learner. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI)* (2012), 53–61.
6. Callan, J. Clueweb12 data set (2013) http://lemurproject.org/clueweb12/.
7. Callan, J., Hoy, M. Clueweb09 data set (2009) http://boston.lti.cs.cmu.edu/Data/clueweb09/.
8. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E.R., Mitchell, T.M. Toward an architecture for never-ending language learning. *AAAI 5*, 3 (2010a).
9. Carlson, A., Betteridge, J., Wang, R.C., Hruschka Jr., E.R., Mitchell, T.M. Coupled semi-supervised learning for information extraction. *Proc. of WSDM* (2010b).
10. Caruana, R. Multitask learning. *Machine Learning 28* (1997), 41–75.
11. Chen, Z., Liu, B. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning 10*, 3 (2016), 1–145.
12. Chen, X., Shrivastava, A., Gupta, A. Neil: Extracting visual knowledge from web data. *In Proceedings of ICCV* (2013).
13. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the 15th National Conference on Artificial Intelligence* (1998).
14. Dempster, A., Laird, N., Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Series B* (1977).
15. Donmez, P., Carbonell, J.G. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management* (2008), ACM, 619–628.
16. Duarte, M.C., Hruschka Jr., E.R. How to read the web in portuguese using the never-ending language learner's principles. In *Intelligent Systems Design and Applications (ISDA), 2014 14th International Conference on* (2014), IEEE, 162–167.
17. Etzioni, O.e.a. Web-scale information extraction in knowitall (preliminary results). In *WWW* (2004).
18. Etzioni, O.e.a. Open information extraction: The second generation. *Proc. of IJCAI* (2011).
19. Gardner, M., Talukdar, P., Krishnamurthy, J., Mitchell, T. Incorporating vector space similarity in random walk inference over knowledge bases. *Proc. of EMNLP* (2014).
20. Krishnamurthy, J., Mitchell, T.M. Which noun phrases denote which concepts. *Proc. of ACL* (2011).
21. Laird, J., Newell, A., Rosenbloom, P. SOAR: An architecture for general intelligence. *Artif. Intel. 33*, (1987), 1–64.
22. Langley, P., McKusick, K.B., Allen, J.A., Iba, W.F., Thompson, K. A design for the ICARUS architecture. *SIGART Bull. 2*, 4 (1991), 104–109.
23. Lao, N., Mitchell, T., Cohen, W.W. Random walk inference and learning in a large scale knowledge base. *Proc. of EMNLP* (2011).
24. Lenat, D.B. Eurisko: A program that learns new heuristics and domain concepts. *Artif. Intel. 21*, 1–2 (1983), 61–98.
25. Maaten, L.v.d., Hinton, G. Visualizing data using t-SNE. *J. Machine Learning Res. 9*, Nov (2008):2579–2605.
26. Mitchell, T.M., Allen, J., Chalasani, P., Cheng, J., Etzioni, O., Ringuette, M.N., Schlimmer, J.C. THEO: A framework for self-improving systems. *Arch. for Intel.* (1991), 323–356.
27. Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N.,

Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., Welling, J. Never-ending learning. In *AAAI Conference on Artificial Intelligence* (2015), AAAI, 2302–2310.
28. Mohamed, T., Hruschka Jr., E.R., Mitchell, T.M. Discovering relations between noun categories. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (2011), Association for Computational Linguistics,Edinburgh, Scotland, UK, 1447–1455.
29. Muggleton, S., Buntine, W. Machine invention of first-order predicates by inverting resolution. *Inductive logic programming* (1992), 261–280.
30. Nigam, K., McCallum, A., Thrun, S., Mitchell, T. Text classification using labeled and unlabeled documents. *Machine Learning 39* (2000), 103–134.
31. Pedro, S.D., Hruschka Jr, E.R. Conversing learning: Active learning and active social interaction for human supervision in never-ending learning systems. In *Advances in Artificial Intelligence–IBERAMIA 2012* (Springer, 2012), 231–240.
32. Platanios, E.A., Blum, A., Mitchell, T.M. Estimating Accuracy from Unlabeled Data. *Proc. of UAI* (2014).
33. Platanios, E.A., Dubey, A., Mitchell, T.M. Estimating Accuracy from Unlabeled Data: A Bayesian Approach. In *Proceedings of the International Conference on Machine Learning* (2016).
34. Platanios, E.A., Poon, H., Mitchell, T.M., Horvitz, E. Estimating Accuracy from Unlabeled Data: A Probabilistic Logic Approach (2017). preprint, https://arxiv.org/abs/1705.07086.
35. Pujara, J., Miao, H., Getoor, L., Cohen, W. Knowledge graph identification. *ISWC* (2013).
36. Samadi, M., Veloso, M.M., Blum, M. Openeval: Web information query evaluation. In *AAAI* (2013).
37. Suchanek, F.M., Kasneci, G., Weikum, G. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)* (2007), ACM Press, New York, NY, USA.
38. Thrun, S., Mitchell, T. Lifelong robot learning. *Rob. Auton. Sys. 15*, (1995), 25–46.
39. Thrun, S., Pratt, L. (eds) *Learning to learn*, Kluwer Academic Publishers, Norwell, MA, USA, 1998.
40. Tong, S., Koller, D. Active learning for structure in bayesian networks. *IJCAI* (2001).
41. Wang, R.C., Cohen, W.W. Language-independent set expansion of named entities using the web. *Proc. of ICDM* (2007).
42. Wieting, J., Bansal, M., Gimpel, K., Livescu, K. Towards universal paraphrastic sentence embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR)* (2015).
43. Wijaya, D.T. *VerbKB: A Knowledge Base of Verbs for Natural Language Understanding*. Ph.D. Dissertation, Carnegie Mellon University, 2016.
44. Yang, B., Mitchell, T. Leveraging knowledge bases in lstms for improving machine reading. *ACL* (2017).

T. Mitchell (tom.mitchell@cs.cmu.edu), Carnegie Mellon University, USA.

W. Cohen, B. Yang, J. Betteridge, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, K. Mazaitis, N. Nakashole, E. Platanios, M. Samadi, D. Wijaya, A. Gupta, X. Chen, and A. Saparov, Carnegie Mellon University, USA.

E. Hruschka, Federal University of São Carlos, Brazil.

P. Talukdar, Indian Institute of Science, India.

A. Carlson and N. Lao, Google Inc., USA.

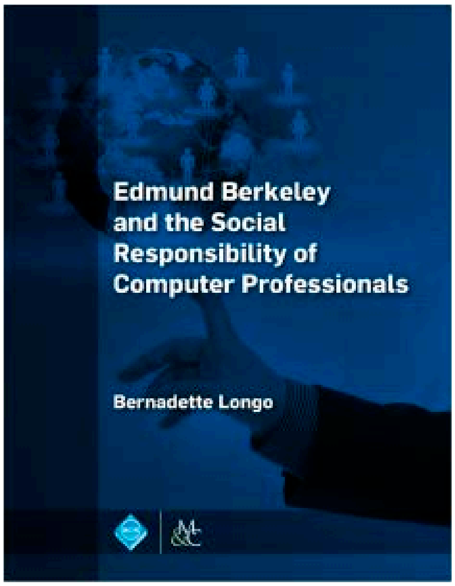
T. Mohamed and R. Wang, Research carried out while at Carnegie Mellon University, USA.

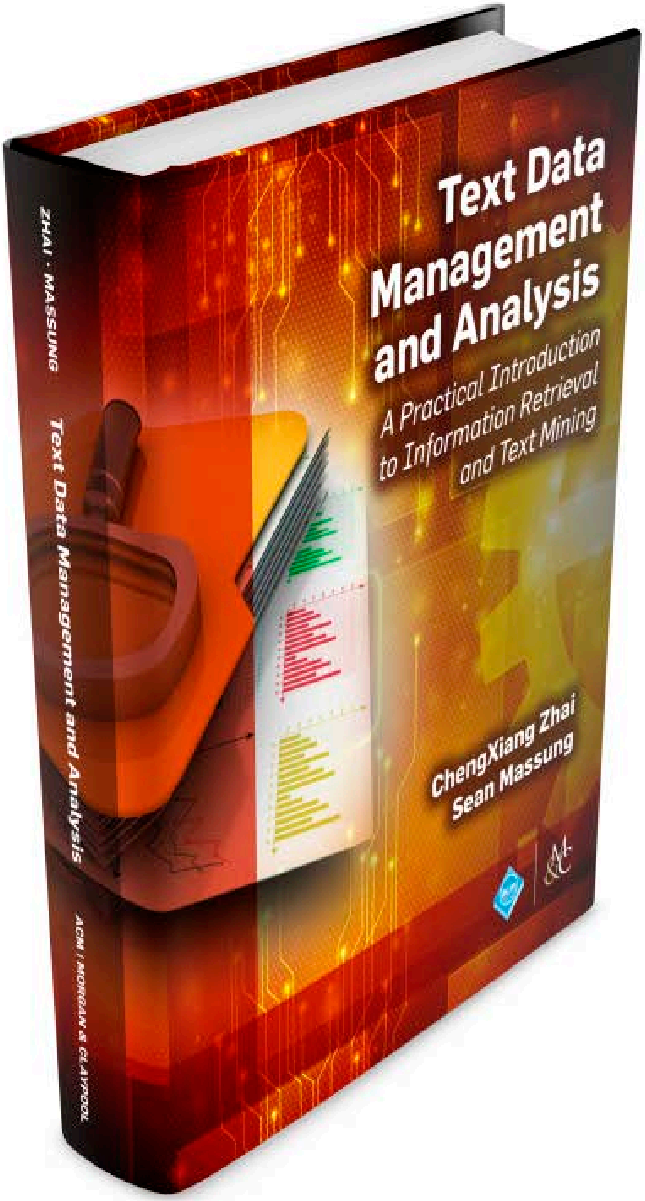
A. Ritter, Ohio State University, USA.

B. Settles, Duolingo, USA.

M. Greaves, Alpine Data Labs, USA.

J. Welling, Pittsburgh Supercomputing Center, USA.


Edmund Berkeley and the Social Responsibility of Computer Professionals
Bernadette Longo


Text Data Management and Analysis
A Practical Introduction to Information Retrieval and Text Mining
ChengXiang Zhai
Sean Massung
ZHAI · MASSUNG
Text Data Management and Analysis
ACM | MORGAN & CLAYPOOL

Purdue University

Assistant/Associate Professor of Practice

The Department of Computer Science at Purdue University is soliciting applications for Professor of Practice positions at the Assistant or Associate Professor level to begin Fall 2018. These are newly created positions offering three- to five-year appointments that are renewable based on satisfactory performance for faculty with primary responsibilities in teaching and service. Applicants should hold a PhD in computer science or a related field, or a BS degree in computer science or a related discipline and commensurate experience in teaching or industry. Applicants should be committed to excellence in teaching, and should have the ability to teach a broad collection of core courses in the undergraduate curriculum. Applicants will also be expected to develop and supervise project courses for undergraduates. Review of applications and candidate interviews will begin on April 1, 2018, and will continue until the positions are filled.

The Department of Computer Science offers a stimulating and nurturing educational environment with thriving undergraduate and graduate programs and active research programs in most areas of computer science. Additional information about the department is available at http://www.cs.purdue.edu. Salary and benefits will be competitive.

Purdue University's Department of Computer Science is committed to advancing diversity in all areas of faculty effort, including scholarship, instruction, and engagement. Candidates should address at least one of these areas in their cover letter, indicating their past experiences, current interests or activities, and/or future goals to promote a climate that values diversity, and inclusion.

Applicants are strongly encouraged to apply online at https://hiring.science.purdue.edu. Alternately hard-copy applications may be sent to: Professor of Practice Search Chair, Department of Computer Science, 305 N. University St., Purdue University, West Lafayette IN 47907. A background check will be required for employment.

Purdue University is an EEO/AA employer. All individuals, including minorities, women, individuals with disabilities, and veterans are encouraged to apply.

Southern University of Science and Technology (SUSTech)

Tenure-Track Faculty Positions

The Department of Computer Science and Engineering (CSE, http://cse.sustc.edu.cn/en/), Southern University of Science and Technology (SUSTech) has multiple Tenure-track faculty openings at all ranks, including Professor/Associate Professor/Assistant Professor. We are looking for outstanding candidates with demonstrated research achievements and keen interest in teaching, in the following areas (but are not restricted to):

► Data Science
► Artificial Intelligence
► Computer Systems (including Networks, Cloud Computing, IoT, Software Engineering, etc.)
► Cognitive Robotics and Autonomous Systems
► Cybersecurity (including Cryptography)

Applicants should have an earned Ph.D. degree and demonstrated achievements in both research and teaching. The teaching language at SUSTech is bilingual, either English or Putonghua. It is perfectly acceptable to use English in all lectures, assignments, exams. In fact, our existing faculty members include several non-Chinese speaking professors.

Established in 2012, the Southern University of Science and Technology (SUSTech) is a public institution funded by the municipal of Shenzhen, a special economic zone city in China. Shenzhen is a major city located in Southern China, situated immediately north to Hong Kong Special Administrative Region. As one of China's major gateways to the world, Shenzhen is the country's fastest-growing city in the past two decades. The city is the high-tech and manufacturing hub of southern China. As a picturesque coastal city, Shenzhen is also a popular tourist destination and was named one of the world's 31 must-see tourist destinations in 2010 by The New York Times.

SUSTech is a pioneer in higher education reform in China. The mission of the University is to become a globally recognized research university which emphasizes academic excellence and promotes innovation, creativity and entrepreneurship.

SUSTech is committed to increase the diversity of its faculty, and has a range of family-friendly policies in place. The university offers competitive salaries and fringe benefits including medical insurance, retirement and housing subsidy, which are among the best in China. Salary and rank will commensurate with qualifications and experience.

We provide some of the best start-up packages in the sector to our faculty members, including one PhD studentship per year, in addition to a significant amount of start-up funding (which can be used to fund additional PhD students and postdocs, research travels, and research equipments).

To apply, please provide a cover letter identifying the primary area of research, curriculum vitae, and research and teaching statements, and forward them to cshire@sustc.edu.cn.

ZJU-UIUC Institute

Zhejiang University / University of Illinois at Urbana-Champaign Institute

Tenure-Track Faculty Positions in Computer Engineering, Computer Science, and Data Sciences

The newly launched Zhejiang University/University of Illinois at Urbana-Champaign Institute (the ZJU-UIUC Institute) invites highly qualified candidates for multiple tenure-track faculty positions at all levels and in areas of engineering and science that match its multidisciplinary mission. We especially seek candidates in computer engineering, computer science, and data sciences.

The ZJU-UIUC Institute is an engineering college on the new Zhejiang University (ZJU) International Campus in Haining, China, about 120 km southwest of Shanghai. The ZJU-UIUC Institute conducts teaching and research in broad program themes of engineering and system sciences; information and data sciences; and energy, environment, and infrastructure sciences. Undergraduate degrees are offered in civil engineering, computer engineering, electrical engineering, and mechanical engineering.

Applications are welcome from relevant engineering disciplines, and especially from computer engineering, computer science, information engineering, data science, and advanced analytics. The Institute has interdisciplinary interests and graduate programs that address but are not limited to high performance computing, data science and engineering, communications and networking, intelligent infrastructure and urban systems, digital manufacturing, environmental sustainability, energy and water integration, and atomic-scale materials and processes. Classes and student activities are conducted in English.

Successful candidates will initiate and lead collaborative research and teaching, and perform academic and professional service duties associated with the ZJU-UIUC Institute. They will be leaders for teaching and research innovation, giving students a meaningful and interactive engineering education.

- Assistant professor candidates must have an earned doctorate, excellent academic credentials, strong research plans, and an outstanding ability to teach effectively. Additional industry, post-doctoral, professional service, or internship experiences are preferred.
- Mid-career candidates must be established leaders in their field; exhibit strong records of teaching, publication, and funded research; and demonstrate participation in interdisciplinary collaborations.
- Senior appointments are available for persons of international stature seeking to build substantial interdisciplinary research and teaching programs.

The Institute will recommend its qualified candidates for the One Thousand Talent Program for Young Outstanding Scientists, China.

The search process will continue until positions are filled or through June 30, 2018. Salary and research initiation support will be commensurate with qualifications and competitive with international norms, and housing benefits are provided. There is additional support for laboratory initiation. Application materials should include a cover letter with current contact information including email address, a complete curriculum vitae, statements of research and teaching goals, and the names of three or more references. Please submit applications at http://zjui.illinois.edu or to zjuihr@zju.edu.cn. For more information, please visit http://zjui.zju.edu.cn.

ZJU and UIUC are renowned for their engineering and computer science programs, and have a long history of collaboration. The ZJU-UIUC Institute leverages a new campus and creates a unique student experience of multidisciplinary collaboration, technical leadership, teamwork, and creative excellence. Individuals with diverse backgrounds, experiences, and ideas who embrace and value diversity and inclusivity are encouraged to apply and all qualified applicants will receive consideration for employment without regard to race, national origin, disability, age, or other personal characteristics.

Contact Person: Xisha (Emma) Zhang, Email: zjuihr@zju.edu.cn, Tel: +86 (571) 8757 2511
Web: http://zjui.zju.edu.cn, http://zjui.illinois.edu

上海科技大学
ShanghaiTech University

TENURE-TRACK AND TENURED POSITIONS

ShanghaiTech University invites highly qualified candidates to fill multiple tenure-track/tenured faculty positions as its core founding team in the School of Information Science and Technology (SIST). We seek candidates with exceptional academic records or demonstrated strong potentials in all cutting-edge research areas of information science and technology. They must be fluent in English. English-based overseas academic training or background is highly desired.

ShanghaiTech is founded as a world-class research university for training future generations of scientists, entrepreneurs, and technical leaders. Boasting a new modern campus in Zhangjiang Hightech Park of cosmopolitan Shanghai, ShanghaiTech shall trail-blaze a new education system in China. Besides establishing and maintaining a world-class research profile, faculty candidates are also expected to contribute substantially to both graduate and undergraduate educations.

Academic Disciplines: Candidates in all areas of information science and technology shall be considered. Our recruitment focus includes, but is not limited to: computer architecture, software engineering, database, computer security, VLSI, solid state and nano electronics, RF electronics, information and signal processing, networking, security, computational foundations, big data analytics, data mining, visualization, computer vision, bio-inspired computing systems, power electronics, power systems, machine and motor drive, power management IC as well as inter-disciplinary areas involving information science and technology.

Compensation and Benefits: Salary and startup funds are highly competitive, commensurate with experience and academic accomplishment. We also offer a comprehensive benefit package to employees and eligible dependents, including on-campus housing. All regular ShanghaiTech faculty members will join its new tenure-track system in accordance with international practice for progress evaluation and promotion.

Qualifications:

- Strong research productivity and demonstrated potentials;
- Ph.D. (Electrical Engineering, Computer Engineering, Computer Science, Statistics, Applied Math, Artificial Intelligence, Statistics or related field);
- A minimum relevant (including PhD) research experience of 4 years.

Applications: Submit (in English, PDF version) a cover letter, a 2-page research plan, a CV plus copies of 3 most significant publications, and names of three referees to: sist@shanghaitech.edu.cn. For more information, visit http://sist.shanghaitech.edu.cn/NewsDetail.asp?id=373

Deadline: The positions will be open until they are filled by appropriate candidates.

ADVERTISING IN CAREER OPPORTUNITIES

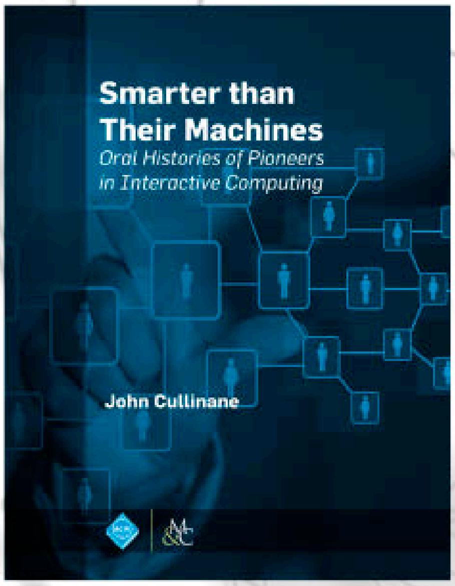
How to Submit a Classified Line Ad: Send an e-mail to *acmmediasales@acm.org*. Please include text, and indicate the issue/or issues where the ad will appear, and a contact name and number.

Estimates: An insertion order will then be e-mailed back to you. The ad will by typeset according to CACM guidelines. NO PROOFS can be sent. Classified line ads are NOT commissionable.

Deadlines: 20th of the month/2 months prior to issue date. For latest deadline info, please contact:
acmmediasales@acm.org

Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at:
http://jobs.acm.org

Ads are listed for a period of 30 days.
For More Information Contact:
ACM Media Sales
at 212-626-0686 or
acmmediasales@acm.org


Smarter than Their Machines
Oral Histories of Pioneers in Interactive Computing
John Cullinane
M&C

[CONTINUED FROM P. 120]

And the policy? Not a word.

“Worse than I expected,” I said.

“Yes,” said Clarkson. “It’s *always* worse than you expect, and it’ll go on getting worse.”

I shrugged. “What can we do?”

Clarkson leaned closer. “Here’s what *you* can do. Keep that card. Check out the reference on your father’s maps. Don’t, whatever you do, look it up online. Borrow the map, or buy your own copy—in a shop, with cash. See if your dad has an *A to Z* of the old hometown. Borrow that, and arrange to borrow your father’s car at short notice. Ten days before the next election, you’ll get a delivery. Take the package and sign for it. Inside will be a briefcase, quite heavy. Tell your partner that you have to go away for a couple of days. Leave your phone at home . . . ”

“What?” I said.

“That’s crucial,” said Clarkson. “Better yet, leave it with your partner, or a friend who lives locally, and ask them to carry it around.”

“But I’d be lost without it!”

“Exactly,” said Clarkson. “That’s the point. Go to the map reference. Nearby, you’ll see a big old oak tree. Look among the roots, and go to the address you find there.”

“And then?”

“Hand over the briefcase, and go home.”

This sounded far too much like a drug or bomb delivery for my liking. I said as much.

“If you’re worried,” said Clarkson, “phone the Leader’s office and ask for me. You’ll get a reply: ‘He’s sound.’ If you’re happy with that, put the phone down. If not, insist on speaking to me. You won’t get through to me, and you’ll never hear of this again.”

Two days later, I phoned, and put the phone down on the reply.

Whatever this was, I was in.

# # #

“I used to be father of the chapel,” Irene confided as she hurried along the dim-lit concrete corridor. For a woman in her 90s, she was remarkably quick on her feet.

“The what?” I said.

“In the print union, it’s what we called the branch secretary.”

The *A to Z*, all dog-eared pages and Sellotaped covers, had all but fallen apart. I tore a page trying to zoom it.

Even I knew what that was—a human version of a workplace rights app.

“Oh,” I said. “More than I need to know?”

“Why?”

“Security.”

She shot me a look as she turned a key in the lock of a steel door.

“Don’t be daft, lad. Now put both hands to this and give it a push.”

I set the briefcase down on the floor, and shoved. The hinges were well oiled, but the door was heavy. It swung open and I followed her in. Ancient fluorescents flickered on overhead. The chamber was large. What wasn’t stacked with oil-drum-size rolls of paper and barrels of ink was occupied by a machine of blue-anodized steel with dials and rotary handles, alongside a row of cabinets. The air smelled of old concrete and oil.

The door clicked shut behind me.

“What is this?”

“A printing press,” said Irene, scornfully.

“I mean, what is this place?”

“Part of the old Regional Seats of Government network. Mothballed and sold off. We bought it. Underground print shop, left over from the last Cold War.”

“And now in use in the current one?”

“Something like that. D’you get here all right?”

“Yes, fine,” I said, untruthfully.

Clarkson’s map reference had taken me to a moorland crossroads. I’d tramped to the nearest lonely tree, where I’d found a tobacco tin so old it didn’t have a health warning. Inside was a scrap of paper, with handwritten address and postcode. The *A to Z*, all dog-eared pages and Sellotaped covers, had all but fallen apart. I tore a page trying to zoom it.

Irene released a folding table from the wall, with a bang that echoed.

“Now, lad,” she said, “let’s see what you’ve got.”

I opened the briefcase and heaved out the two thick stacks of A4 that crammed it. Irene put on reading glasses and peered at a few sheets, tilting them this way and that.

“Typed on a typewriter,” she said approvingly, “with handwritten corrections. This is the real thing all right.” She indicated a corner in which an electric kettle and some mugs sat on an old chair. “Make yourself a cup of tea, and I’ll go let the lads and lasses in.”

The “lads and lasses” turned out to be four old men and two old women, who set about their work. Within two hours, the press was rolling.

It took me a little longer to understand what was going on.

A week later a woman I knew from the Party hammered on my door at an ungodly hour.

“Out,” she said. “Station. Now.”

Her car was overloaded with newspapers. She dropped me off at the station with a bundle to give away to anyone who would take a copy. At every station and shopping center in the country, others would be doing the same. We’d blanket the land with paper—leaflets, newssheets, posters, placards—whose production and distribution owed nothing to the net.

I had a shock when I jumped out of the car. Our opponents had a team already at the station, handing out copies of *their* free newspaper. They’d had the same idea. Or we’d had a leak, our elaborate precautions all for naught.

My fury faded. It didn’t matter. Now they had to fight us on the same ground—and the ground was level.

We lost that election, but we got democracy back.

Ken MacLeod (kenneth.m.macleod@gmail.com) is the author of 17 novels, from *The Star Fraction* (Orbit Books, London, 1995) to *The Corporation Wars: Emergence* (Orbit Books, London, 2018). He blogs at The Early Days of a Better Nation (http://kenmacleod.blogspot.com) and tweets as @amendlocke.

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/3197548

Ken MacLeod

Future Tense
Free Press

When all online news and comment can be digitally manipulated, some might recall a more trustworthy way to spread the word.

I BARELY RECOGNIZED Clarkson. He would have been the last of my high school acquaintances I'd have expected to meet at a Party event—some policy-launch shindig, all tepid canapés, foul coffee, and wobbly display boards—but there he was, 10 years older. Trim beard, sharp tie.

"Oh, by the way," he said, after the mutual reintroductions, "is your old man still in the Ramblers?"

"He stretches his legs on the moors occasionally," I said.

Clarkson put down his formerly fizzy water and stepped forward.

"He still has a car?"

"Uh huh."

"And ... let me think ... he hails from"—Clarkson named a Northern industrial town—"doesn't he?"

"Yeah," I said.

"Great, great!" Clarkson rubbed his hands, and stepped forward again. "I have something for you."

He passed me a business card. It predictably proclaimed him a "business consultant."

"Turn it over," said Clarkson.

A string of numbers and letters, inked carefully—an Ordnance Survey map reference.

"What does that—?"

"Just slip the card in your shirt pocket," said Clarkson.

I backed off a little. I could smell the last thing he'd eaten, a wafer of damp oatcake with a dab of hummus and a sliver of smoked salmon.

"You know how to read it?" he said.

"Sure," I said. "My dad has a bookcase full of OS maps."

Clarkson's cheek twitched. "Fits the profile."

"Profile?" I gave him a suspicious look. "What's all this about?"

Throughout, Clarkson had been backing me into a corner. I didn't like it, and let him know.

"It's *always* worse than you expect, and it'll go on getting worse."

"Sorry," he said, not shifting. "We need to talk. Quickly and quietly."

"OK," I said, still wary.

Clarkson waved vaguely behind his shoulder. Half a dozen people here were reporters, their head-mounted camera-mics discreetly hidden by strategic locks of hair. "Take a look at this event on your phone," he said.

"Oh, I know what to expect," I said.

"No, really."

I searched on the hashtag for the event. It brought up two sex scandals, a corruption allegation, an embarrassing social media post by one of the speakers from when she was 12 years old, and a dubious link to an oil company.

[CONTINUED ON P. 119]

IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

10010