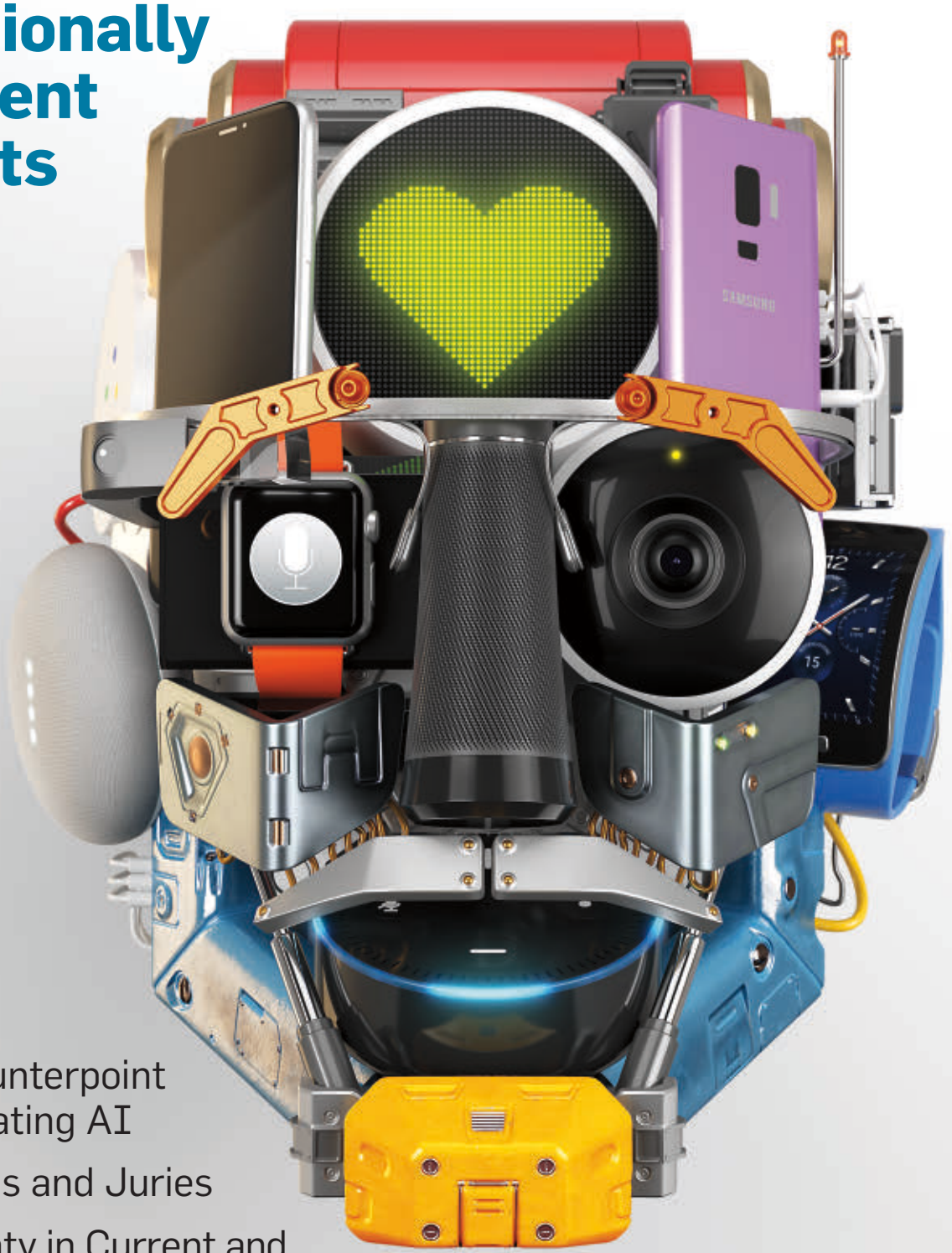
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

12/2018 VOL.61 NO.12

# Designing Emotionally Sentient Agents



Point/Counterpoint on Regulating AI

AI Judges and Juries

Uncertainty in Current and Future Health Wearables

Q&A with Peter G. Neumann

Association for Computing Machinery

ICWS Founded in 2003

# 2019 Ai · Blockchain · Cloud · bigData

## Everything is connected

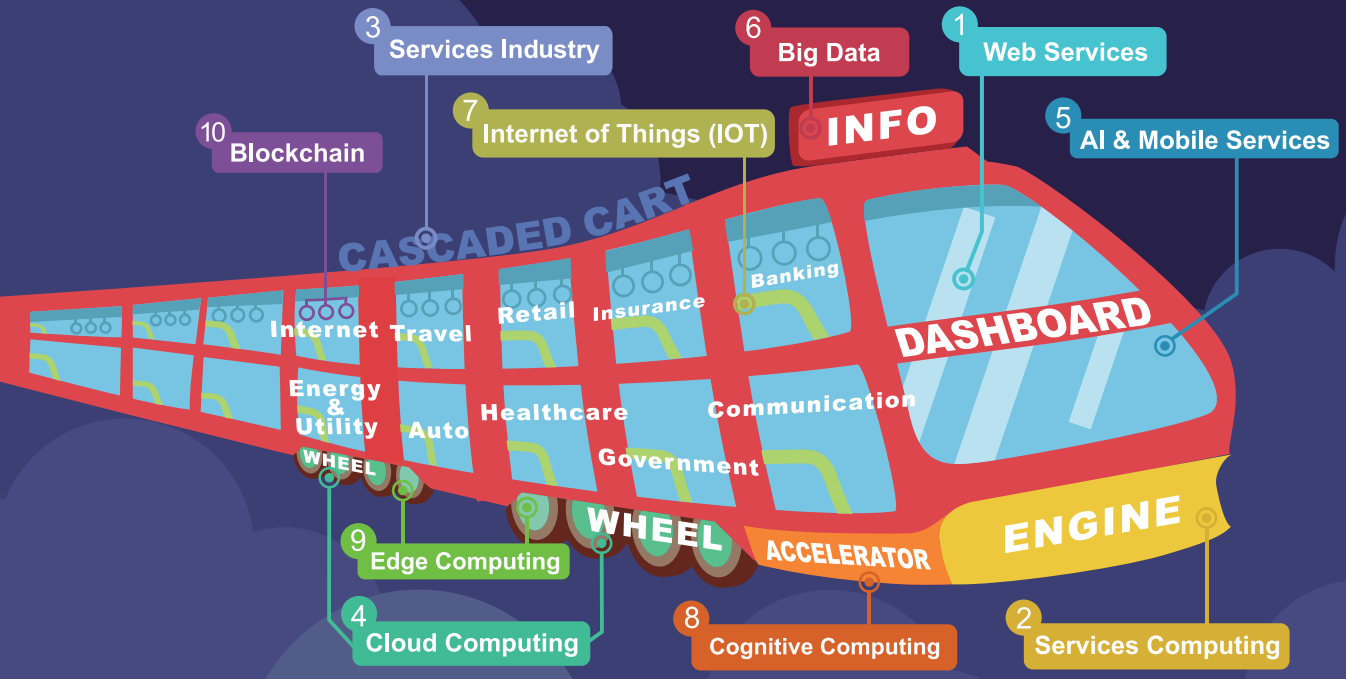
THEME: NEW ABCDE

June 25 – June 30, San Diego, USA

CELEBRATING THE 17th BIRTHDAY OF SCF

SCF
SERVICES CONFERENCE FEDERATION

1. 2019 International Conference on Web Services **(ICWS 2019)**
2. 2019 International Conference on Services Computing **(SCC 2019)**
3. 2019 World Congress on Services **(SERVICES 2019)**
4. 2019 International Conference on Cloud Computing **(CLOUD 2019)**
5. 2019 International Conference on AI & Mobile Services **(AIMS 2019)**
6. 2019 International Congress on Big Data **(BigData 2019)**
7. 2019 International Conference on Internet of Things **(ICIOT 2019)**
8. 2019 International Conference on Cognitive Computing **(ICCC 2019)**
9. 2019 International Conference on Edge Computing **(EDGE 2019)**
10. 2019 International Conference on Blockchain **(ICBC 2019)**




## Submission Deadlines

1/6/2019: ICWS 2019 (http://icws.org)
1/21/2019: SCC 2019 (http://theSCC.org)
1/25/2019: SERVICES 2019 (http://ServicesCongress.org)
1/6/2019: CLOUD 2019 (http://theCloudComputing.org)
1/21/2019: AIMS 2019 (http://ai1000.org)

1/28/2019: BigData 2019 (http://BigDataCongress.org)
2/17/2019: ICIOT 2019 (http://iciot.org)
2/17/2019: ICCC 2019 (http://theCognitiveComputing.org)
2/17/2019: EDGE 2019 (http://theEdgeComputing.org)
2/17/2019: ICBC 2019 (http://Blockchain1000.org)

Email:
confs@ServicesSociety.org

SERVICES SOCIETY
Leading not-for-profits organization ( US 501(c)(3) ) dedicated for serving 30,000+worldwide services computing professionals

ICWS.ORG

All the 10 conference Proceedings of SCF 2018 have been published by Springer. The Proceedings of SCF 2019 will be continuously published in the LNCS by Springer. Currently the Abstracting and Indexing services covered by Springer's data feeds for LNCS proceedings include ISI Conference Proceedings Citation Index (included in Web of Science), Engineering Index EI (Compendex and Inspec databases), DBLP, Google Scholar, IO-Port, MathScinet, Scopus, and ZBIMath.

The Next Generation Internet is a European-led initiative that brings together top internet innovators, researchers and policymakers, who are shaping the internet of tomorrow.

NEXT GENERATION INTERNET

*"We need to shake and move the foundations of the internet"*

We talk to Dr. Monique Calisti, CEO of Martel Innovate and HUB4NGI project coordinator, about why organizations and individuals should join the Next Generation Internet initiative.

## What is the Next Generation Internet initiative?

The Next Generation Internet (NGI) initiative was launched by the European Commission in 2016 under the auspices of Roberto Viola, Director-General for Communications Networks, Content & Technology, fostering the creation of an internet capable of offering more to people and to our society. The NGI Initiative taps into and seeks to lead a global upswell of players who are focusing on creating a democratized, decentralized, and human-centered internet.

## What does a "human-centered" internet mean?

The NGI's ambition is an internet that provides better services by giving people back control of their data and empowering them to participate. This means using technology in a way that upholds human dignity, privacy and security, while being transparent and trustworthy. It's an internet that serves the many, not the few, and is accessible to all. We need to shake and move the foundations of the internet if we are to achieve this.

## What technologies and disciplines will the NGI embrace?

The NGI initiative promotes a coherent integration of ethics, design and education with cutting-edge technologies such as—Artificial Intelligence, the Internet of Things, Interactive Technologies, Digital Learning and Smart Connectivity—to transform the internet and deliver social good and economic benefits to everyone.

## How can organizations join NGI and how much does it cost?

The NGI is free to join. Join by signing up to the NGI mailing list, taking the NGI survey, adding your organization to the NGI online map, meeting the community at upcoming events and participating in the NGI Open Calls, which offer a unique opportunity to fund researchers and innovators at work for a better internet.

# JOIN THE NEXT GENERATION INTERNET

NGI.EU | NGI.EU/MAP | @NGI4EU

Co-funded by the H2020 programme of the European Union

### Why join the NGI?

- Steer the digital transformation process in Europe and internationally
- Gain visibility among a vibrant, growing community of innovators who are creating the internet of tomorrow
- Get your research and innovation funded

**Visit www.ngi.eu for more information**

# COMMUNICATIONS OF THE ACM

## Departments



## Last Byte



## News



## Viewpoints



## Practice




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

IMAGE BY LIGHTSPRING

12/2018
VOL. 61 NO. 12

## Contributed Articles



## Review Articles



Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/designing-emotionally-sentient-agents




**About the Cover:** This month's cover story (p. 74) tells of the power that emotions play in our interactions with technology and the challenges in designing emotionally sentient agents to raise that interaction to a new level. Such agents can take many forms. How many can you find in our cover image? Cover illustration by MDI Digital.

## Research Highlights



Association for Computing Machinery
*Advancing Computing as a Science & Profession*

# COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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

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







**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

DOI:10.1145/3289429

Vinton G. Cerf

# Self-Authenticating Identifiers

Public key cryptography (PKC) has the elegant property that requires two keys—one public and one private. Either key can be used to encrypt or decrypt, but the peculiar property that makes PKC interesting is that you must use one key to encrypt and the other to decrypt. For example, the private key could be used to encrypt the digital hash of a binary message. The recipient of the message could compute the same hash and then decrypt the encrypted digital hash with the public key. If they match, the recipient can be more certain the sender holds the private key and is the originator of the message. Of course, the recipient must know the public key to do this confirmation. The concept is called a digital signature and led to the creation of certificate authorities (CAs) that could issue certificates binding an identifier (for example, a domain name) to the public key of a PKC pair.

The other function for confidentiality uses the public key to encrypt the message to be sent to the public key owner. Assuming only the owner has the private key, only the owner can decrypt the message, providing confidentiality. The sender of such a confidential message uses the certificate to determine which public key to use to send to the recipient.

These mechanisms lead to a concept called trust on first use or TOFU. If a party sends a message and says "this is my public key," at best, the recipient can use this information to confirm that a second message has come from the same source by checking the digital signature or by generating a random number, encrypting it in the putative public key of the sender, sending it to the originator of the second message, challenging the sender to decrypt the challenge and return it (perhaps encrypted in the public key of the challenger). The trust part enters into the picture because the recipient of the first message must trust that the public key is associated with a known or knowable party. The certificate idea was used to confirm that a third party has validated the certificate owner's bona fides, but some CAs were compromised, invalidating the trust.

Suppose, in lieu of domain names, one used a public key as an identifier and associated this with an Internet Protocol (IP) address. If one looked up the IP address in a registry of public key identifiers, one could then challenge the device at that IP address to show it still has the associated private key using a challenge/response protocol as suggested earlier. If the party registering the public key and its associated IP address has to show significant bona fides to the registry, this might produce a kind of TOFU-plus[a] that gives the party reaching the computer at the destination IP address more confidence that this is the intended destination.

**A function for confidentiality uses the public key to encrypt the message to be sent to the public key owner. Assuming only the owner has the private key, only the owner can decrypt the message, providing confidentiality.**

One might imagine applying this to the Internet of Things (IOT) in which the IOT device self-generates a public- and private-key pair and registers the public key. For example, with a hub or controller so the hub can confirm it has reached the right IOT device. By the same token, configuration of the IOT device into an ensemble could include incorporation of the public key of the controller into a list of valid devices that can command or obtain data from the now-configured IOT device. Both ends can verify they are talking to the originally configured devices, assuming no device has lost its private key. While perhaps not obvious, a failure to validate an identifier by this method does not give much information to the user (or program) seeking validation. Of course, the act of registering a device with the hub or controller must be a trusted process that might involve physical presence, Bluetooth key pairing, or other action such as proximity NFC that increases confidence in the registration process. One could imagine QR codes or even public key strings associated with the registering device that must be captured by a mobile camera or keyed in by the configuring party to increase trust in the process. C

a Thanks to Ted Hardie, IAB chair, for this terminology.

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


# LIVE HEALTHY

## DESIGN • CREATE • EXPLORE

**ACM Interaction Design & Children Conference**
**June 12-15, 2019 in Boise (Idaho, US)**

IDC 2019 will be the 18th in an annual series that encourages research concentrating on the design, development, and use of interactive technologies for children.
Ambitious and idea-driven researchers, designers, and educators from all over the world will gather for IDC to share and discuss these ideas. This year's theme is Live Healthy!

IDC 2019 welcomes submissions in the form of full papers, short papers (notes), demos, courses, workshops, work in progress or late breaking, and interactive child experiences.
There is also a call for participants to submit to the doctoral consortium, and the research and design competition.

We look forward to seeing all of you here in Boise, the City of Trees!


Association for Computing Machinery

**IDC.ACM.ORG/2019**

DOI:10.1145/3289134

# Reclaim Internet Greatness

Vinton G. Cerf's Cerf's Up column "The Internet in the 21st Century" (Sept. 2018) highlighted many challenges facing today's Internet, including risks to privacy, security, and society that did not exist when the network was originally being built in the late 1960s. His concern is warranted and will require us to strike a balance between protecting the democratic and egalitarian values that made the Internet great to begin with while ensuring those values are used for good. The fundamental issue, then, in creating a 21st-century Internet becomes what changes are warranted and who will be responsible for defining and administering them.

On the technology dimension, computer scientists and engineers must develop smarter systems for detecting, addressing, and preventing malicious content on the Web. Cerf's argument on behalf of user training is helpful but will not ultimately solve the problem of an untrustworthy, ungovernable, potentially malicious network. I myself recently fell for a phishing attack, which only proves that today's attacks can fool even savvy, experienced users. Meanwhile, bad actors worldwide exploit the same infrastructure that is used by billions of well-intentioned people every day, including notoriously scammers using Google AdWords in 2017 to impersonate Amazon.[1] There is clearly a need for systems deployed across all layers of the Internet stack to prevent, detect, and address such abuses.

Technology by itself is not the solution. Billions of ordinary people thus have a role to play in combatting such potentially destructive forces. Critical thinking and digital literacy must be taught in grade school, enabling even children to question the overload of information, including misinformation churned out by countless bad actors. Society must also reward, rather than stigmatize, people and organizations for doing the right thing, as when an organization admits it has suffered an attack, even if embarrassing to admit.

A free and democratic Internet available to all requires a governance structure that reflects the network's global scope. A major reason it continues to succeed is that no single government agency has total control. Instead, bits of it are maintained by various international non-governmental agencies, including the Internet Corporation for Assigned Names and Numbers. However, we are beginning to see national and regional governments more directly regulate the Internet their citizens access through filters or data privacy laws (such as the European Union's General Data Protection Regulation). Such walled gardens produce a fragmented experience for many users. Needed are international non-governmental organizations that address these concerns, with the legal authority to perform digital forensics to identify cyber attacks and criminals, perhaps even by issuing Interpol-type warrants for alleged criminals. They could also formally pass and enforce rules and publish guidance covering, say, data privacy as well.

**Reference**

1. Vaas, L. Scammers slip fake Amazon ad under Google's nose. *Naked Security* (Feb. 10, 2017); https://nakedsecurity.sophos.com/2017/02/10/scammers-slip-fake-amazon-ad-under-googles-nose/

**James Simpson,** York, U.K.

## Lookahead Search for Computer Chess

Vinton G. Cerf views himself as a layman on the subject of neural networks, and so do I. However, I think I understand heuristic (lookahead) search, especially for computer chess. I thus limit myself here to Cerf's assertion in his Cerf's Up column "On Neural Networks" (July 2018) that "AlphaGo Zero learned to play chess well enough to beat most (maybe all?) computer-based players in 24 hours." Since he did not mention "search" per se, such a statement suggests that learned neural networks would be likely to perform well in chess, as was reported in *Chess News* (https://en.chessbase.com/post/thefuture-is-here-alphazero-learns-chess) and mentioned by Cerf in the column.

"Lookahead search" means exploring the state-space rooted in a given (chess) position. Due to combinatorics, exhaustive exploration is normally impossible in computer chess, yielding two successful approaches:

*Minimaxing.* Exploring to bounded (though usually variable) depth, as Scheucher and I explained,[1] why minimaxing heuristic values works at all, despite its theoretical problem identified by ACM Turing Award laureate Judea Pearl and others; and

*Monte Carlo tree search.* Sampling through selective explorations to "leaf" positions (such as checkmate).

For my point here, the particular search approach needed to achieve excellent performance does not matter, only whether search is involved in the move decision. I conjecture that AlphaZero[2] would achieve superhuman performance in chess through minimaxing as well.

In describing AlphaZero (which I am convinced is what Cerf meant to say in the column, rather than "AlphaGo Zero"), Silver et al.[2] did not mention Monte Carlo tree search until somewhere in the middle of their paper, reflecting its status as not particularly important. In fact, AlphaZero did not learn but was simply programmed to perform search. Note that Silver et al. thoroughly analyzed the role of search in the predecessor AlphaGo in *Nature* in 2016.

Although neither AlphaZero nor its learned neural network appear to be publicly available, I have managed to explore a reimplementation called LCZero at http://lczero.org/. I recently matched up its neural network *without search* against my (and colleagues Helmut Horacek's and Marcus Wagner's) old Merlin chess program, which tied for sixth place out of 24 machines in the 1989 World Computer Chess Championship. I emulated the computer power it had at the time on a five-year-old laptop (with a single core)

by setting its performance time at two seconds per move rather than three minutes on a mainframe at the time. While I consider the moves played by LCZero's neural network generally plausible, even without search, Merlin beat it decisively in these games, as a neural network without search often blunders in chess positions where nontrivial tactics really *do* matter.

I thus consider Cerf's claim about computer chess and neural networks misleading, but is such a claim indeed worth pointing out here, in a letter to the editor? What worries me is the effect of the kind of "telephone game" begun by Silver et al.,[2] a message mischaracterized in *Chess News* and now again in *Communications* by an author of Cerf's stature. We can only imagine how such a telephone game would continue to play out in the regular nontechnical press if we did not cut the thread now.

**References**

1. Scheucher, A. and Kaindl, H. Benefits of using multivalued functions for minimaxing. *Artificial Intelligence 99*, 2 (Mar. 1999), 187–208.
2. Silver, D. et al. Mastering chess and Shogi by self-play with a general reinforcement learning algorithm. arXiv, 2017; https://arxiv.org/abs/1712.01815

**Hermann Kaindl,** Vienna, Austria

### Author Responds:

*Kaindl is correct. I meant to say AlphaZero and mistakenly wrote AlphaGo Zero. I also I agree that search is important for the AlphaZero system and should have drawn attention to it but honestly had not done enough due diligence with the DeepMind team, a deadline effect ...*

*I appreciate the additional color and clarity.*

**Vinton G. Cerf,** Mountain View, CA

### *Communications* Addresses the Intellectual Challenges of CS

I am not one given to writing complementary letters to publications but must say *Communications* (Sept. 2018) was brilliant in exploring several extremely relevant intellectual challenges with its readers, specifically Vinton G. Cerf's discussion of the "Treaty of Westphalia" in his Cerf's Up column "The Peace of Westphalia" and its relevance to ongoing international interference in elections around the world; Moshe Y. Vardi's discussion of "disruptive technology" in his Vardi's Insights column "Move Fast and Break Things," noting that while computer scientists should "celebrate" their achievements, they also need to, as Vardi put it, "drive very carefully"; the passion of the letters to the editor concerning a prior Vardi column "How the Hippies Destroyed the Internet" (July 2018) on the past (and future) of the Internet; and my favorite, Adam Barker's Viewpoint "An Academic's Observations from a Sabbatical at Google" on the importance and relevance of software practice to software academics worldwide.

I also noted the then-recently announced "China Region Special Section" (Nov. 2018). As a participant in a 1987 People to People international travel visit to China with other international computer scientists, I recall being amazed to find that, in what was at the time a fairly primitive country technologically, Chinese computer scientists were almost all theoreticians, rather than pragmatists. It took considerable thought for me to realize, or perhaps rationalize, that this was happening because labor was notably inexpensive in China at a time that doing things manually was significantly less costly than doing them with computer support.

**Robert L. Glass,** Toowong, QLD, Australia

### Before Inviting More Tech Workers into the U.S. ...

James Simpson's letter to the editor "Side with ACM Ethical Values" (Aug. 2018) concerning Moshe Y. Vardi's Vardi's Insights column "Computer Professionals for Social Responsibility" (Jan. 2018) rightly suggested that ACM policies are inextricably linked to ethical and moral values and hence to political considerations. Simpson even urged establishment of a new ACM special interest group dedicated to ethics and public policy.

But Simpson's implicit encouragement of ACM to lobby for immigration of tech workers into the U.S. fell flat on two accounts: First, it would avoid responsibility for home-growing the talent American industry needs. The news story "Broadening the Path for Women in STEM" by Esther Stein (also in Aug. 2018) included data on the declining participation of women in CS majors in the U.S. Meanwhile, separately, members of the U.S. team in the International Mathematical Olympiad (http://www.ams.org/news?_id=4446) are almost always entirely Asian-Americans. And second, and perhaps more important from an ethical point of view, U.S. hiring of foreign tech workers can amount to a form of "colonial brain drain" from other countries that need their own IT professionals at home.

A broad call to U.S. students to pursue STEM or CS careers is not the answer. American companies do not need workers who are poorly prepared, do not love what they do, or do not care about the moral dimensions of their work. Tech companies, as well as computer science and engineering generally, need people called to IT as a profession, not just a career. ACM should thus consider these issues before deciding to lobby on behalf of bringing in tens of thousands more foreign tech workers.

**Paul J. Campbell,** Beloit, WI, 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.

Association for Computing Machinery

## ACM Chuck Thacker Breakthrough in Computing Award

# The "ACM Breakthrough Award"

## Nominations Solicited

Nominations are invited for the inaugural 2018 ***ACM Charles P. "Chuck" Thacker Breakthrough in Computing Award*** (the "ACM Breakthrough Award").

ACM Turing Laureate Charles P. (Chuck) Thacker (1943–2017) received the 2009 ACM A.M. Turing Award for "the pioneering design and realization of the first modern personal computer—the Alto at Xerox PARC—and seminal inventions and contributions to local area networks (including the Ethernet), multiprocessor workstations, snooping cache coherence protocols, and tablet personal computers."

The award was established in recognition of Thacker's pioneering contributions in computing.

These contributions are considered by the community to have propelled the world in the early 1970s from a visionary idea to the reality of modern personal computing, providing people with an early glimpse of how computing would deeply influence us all. The award also celebrates Thacker's long-term inspirational mentorship of generations of computer scientists.

The Breakthrough Award will recognize individuals with the same out-of-the-box thinking and "can-do" approach to solving the unsolved that Thacker exhibited. The recipient should be someone who has made a surprising or disruptive leapfrog in computing ideas or technologies that provides a new capability or understanding that influences the course of computing technologies in a deep and significant manner through its numerous downstream influences and outcomes. Due to the breakthrough nature of the award it is expected that it will be presented biennially and will not be presented if there is no candidate who meets the criteria in a particular year.

The award is accompanied by a prize of $100,000 and would be presented at the annual ACM Awards Banquet. The award recipient would be expected to give the *ACM Breakthrough Lecture* at a major ACM conference of his or her choice during the year following the announcement. The travel expenses of the recipient, and a companion, to attend the Lecture are supported by the award. Financial support of the Thacker Award is provided by Microsoft.

Nomination information and the online submission form are available on:
*https://awards.acm.org/thacker/nominations*

The deadline for nominations/endorsements is:
**January 15, 2019, End of Day, AoE, UTC-12 hours.**

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

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:10.1145/3282874 http://cacm.acm.org/blogs/blog-cacm

# Securing Agent 111, and the Job of Software Architect

*John Arquilla describes the new state of cyberspying, while Yegor Bugayenko considers the importance of a software architect to development projects.*

**John Arquilla**
**From 007 to 'Agent 111'**
http://bit.ly/2DO1wmc
June 18, 2018

The information revolution has changed just about every aspect of society and security in our time, so it's no surprise that the spy business has been transformed as well. Yes, there are still human "moles" who scurry about inside organizations, gathering up vital information for their foreign masters, and no doubt those "sleepers" deported from the U.S. back to Russia in a 2010 prisoner swap were not the last of their kind; a real-life version of the television series "The Americans" likely continues, in many countries.

Yet adventurous James Bond-like spies have been eclipsed by a new generation of operatives who don't travel the world (not physically, anyway) or drink martinis, shaken *or* stirred. Indeed, most of their time is spent tapping away at keyboards in cool, windowless rooms, their favored beverage some brand of highly caffeinated energy drink. Bond is giving way to Agent 111 ("007" in binary), who oftimes might just be a smart bot.

The latest exploit of some Chinese Agent (or agents) 111, made public this month, has to do with sensitive data about American submarine operations. Access apparently was gained by hacking a private contractor doing work in this area for the U.S. Department of Defense. By infiltrating in this indirect manner, cyber-spies were able to vacuum up over 600GB of data that, when the pieces are put together, may provide a valuable picture of how the U.S. Navy intends to operate in contested waters like the East China Sea.

This serious breach, a coup for Chinese intelligence, came in the wake of a string of damaging hacks aimed at strategic targets in the U.S. One of the worst was revealed on March 15 (talk about "Beware the Ides!") in a report issued by the FBI and the Department of Homeland Security that asserted a well-crafted Russian-sponsored intrusion effort had gotten in to our power and water infrastructures. Given these systems are highly reliant upon automated controls, the idea some latter-day virtual James Bond might be able to "cybotage" them is most troubling. For those who worry about how such hacks might hurt our military, give Pete Singer and August Cole's *Ghost Fleet* (http://bit.ly/2y4v2xC) a close read.

Back in 2015, one of the things U.S. President Barack Obama and China's President Xi Jinping discussed when they met was the matter of curbing hostile cyber activities aimed at the theft of commercial intellectual property. This Information-Age form of industrial espionage was costing the U.S. hundreds of billions of dollars each year. Both leaders agreed to declare a moratorium on this aspect of cyber-spying, though the Trump Administration has recently charged the Chinese with serial violations to it. Yet it is important to note, of the Obama-Xi agreement, that conducting cyber espionage in the military and security realms was *not* addressed. This omission signaled to intelligence agencies in both countries—and to their counterparts around the world—that a new "cool war" was under way, and it was not to be curtailed.

There are two problems with tacit acceptance of cyberspace-based spying on militaries and other actors. The first is that intrusions, though they may be for intelligence-gathering purposes, are observationally equivalent to attack preparations. How is one to know whether the mapping of one's systems is prelude to an imminent attack, or to an attack at some undetermined time in the future? Either way, this form of cyber espionage is unsettling, because of the threat of actual attack that may undergird it.

The second problem is that the line between military and non-military targets can be blurry, given that much of advanced information technology is inherently "dual use;" that is, the hardware and software that enliven commerce can do the same for conflict. In terms of the Obama-Xi agreement, hackers might legitimately claim in going after sensitive intellectual property—for example, plans to the F-35 fighter plane—that all the tech related to design and production of this aircraft were fair game. Indeed, one need only look at the Chinese knock-off of the F-35 to see the strong similarities, and to infer what happened.

That raises another point about the threat posed by Agent 111: by gaining access to massive amounts of highly sensitive information via cyber-spying, as in this most recent intrusion into the computers of the U.S. Navy contractor, sufficient knowledge may be gained to allow the intruding party to leap immediately to the most advanced technology without having to go through the typically long, repetitive cycles of research, development and design. Thus, Agent 111 is key to a beneficial phenomenon Alexander Gerschenkron labeled "late modernization."

In short, Agent 111 may prove far more effective—and far more lethal, in military effects—than 007 could have hoped to be. Further, cyber-spying is nearly impossible to deter, and when it comes to the views of heads of state, it seems to be accepted, in the context of military and security affairs at least, as "just a new form of espionage." The only viable answer, given the sorry trail of high-level intrusions into American and other countries' information systems, is that full emphasis must be placed on improving defenses. Firewalls and antivirals will simply not do. The Cloud, the Fog, and the ubiquitous use of strong encryption should be emphasized as first steps toward mitigating the terrible vulnerabilities that can, thanks to the human and virtual Agents 111 coming on line (literally), hold any nation at grave risk.

**Yegor Bugayenko**
**The Era of Hackers Is Over**
**http://bit.ly/2Ovu1ZX**
**July 5, 2018**

*How efficient is your current software project, and could it potentially benefit from the addition of a software architect? More importantly, what exactly does a software architect do, and what can they provide to your team? With the world of software development rapidly moving towards more agile workflows amidst democracy in the front seat, the importance of the software architect is understated. A position misunderstood by many is a crucial component that delivers unparalleled guidance in the project pipeline, assigning responsibility to an individual who can turn a company vision into code.*

Some might believe the title of software architect is merely a status symbol placed upon a senior coder, signaling a specific level of respect should be delivered; this assumption is wrong. The job of the architect is one that can be highly significant if it is adequately bestowed and the person who receives the title has the qualifications to lead a team. Most importantly, the individual must be able to take the blame for project failures.

The software architect is the individual who takes the blame for when a project fails or is praised when the software, and the team, succeeds. Now, we must understand what is meant when using the word "blame" and why such a large association would be placed with an individual. The software architect is your team's guide; they are selected to carry the initial vision to a fully solidified working piece of code. As leaders, they elect to take the responsibility for the direction in which they lead their team.

Lead Software Engineer at EPAM Systems Nikolay Ashanin compared the responsibility of a software architect to that of a bridge worker in the 19th century in his published article *The Path to Becoming a Software Architect* (http://bit.ly/2O3L7ig) and said at that time the key group of engineers, architects, and workers stood under the bridge while the first vehicles were on it; they staked their lives upon the construction and strength of the structure.

When we say a software architect must absorb the blame for a project, we are merely saying the project outcome that is produced shall fall upon their shoulders. It is entirely up to the software architect to delegate responsibilities of a project utilizing their methodologies, whether that be additional toolsets, their authority, or mentorship and coaching.

Project managers do not always have the option to hire a software architect, as they are typically individuals who are curated by their company, learning and understanding their team over time. In an excellent article (http://bit.ly/2Ni0wpU) by Simon Brown of InfoQ, a division of C4 media that focuses on software development, Brown noted, "becoming a software architect isn't something that happens overnight or with a promotion. It's a role, not a rank."

Most importantly, the decision of a software architect must be treated as final. Otherwise, without a true final say in the matter, the individual won't be looked upon as an authoritative figure. Even a project manager must treat the software architect as the final decision maker when it comes to implementing and producing code. Rather than overruling the decisions of their architect, project managers should seek to replace the individual if product end-visions are not adequately aligning. An individual does not need to be fired, but perhaps placed back within the standard pool of programmers; over time, they might professionally grow to attempt the opportunity once more.

A software architect is the guiding rails for a project; they keep their team of developers moving forward and on-vision while accepting the responsibilities for the team's actions as a whole. Not only must an architect be able to lead, but also to understand the skills of their team, and how they can contribute to a finished project.

Beyond the ability to craft beautiful code, lead a team to completion, and work under pressure, a software architect must stand as a figure able to accept responsibility for a project; this is the characteristic that defines a true architect. More than simply a senior programmer, more than simply a leader, the software architect stands as a gatekeeper for quality and as a guiding vision for their team. In the end, whether the result is positive or negative, the software architect can stand up and take the praise or blame for what their team has accomplished. C

**John Arquilla** is professor and chair of defense analysis at the U.S. Naval Postgraduate School; the views expressed are his alone. **Yegor Bugayenko** is founder and CEO of software engineering and management platform Zerocracy.

# The 8th ACM International Symposium on Pervasive Displays

Palermo, Italy, 12-14 June 2019

PERDIS 2019

This is the premier venue for discussing opportunities and challenges raised by the emergence of pervasive display systems as a new communication medium for (semi-)public spaces. As a targeted topic venue, we offer participants a unique opportunity to network with a diverse but focused research community resulting in an extremely lively event with all the energy and excitement that characterizes the emergence of a new research area.

A selection of accepted papers will be invited to submit extended versions to a theme issue on Pervasive Displays of Springer's "Personal and Ubiquitous Computing" journal.

**We are looking forward to seeing you in Palermo in June 2019!**

## GENERAL CHAIRS

**MOHAMED KHAMIS**
University of Glasgow

**SALVATORE SORCE**
Università degli Studi di Palermo

## PROGRAM CHAIRS

**JESSICA CAUCHARD**
Interdisciplinary Center (IDC) Herzliya

**VITO GENTILE**
Università degli Studi di Palermo

## POSTERS CHAIRS

**JORGE GONCALVES** — University of Melbourne
**MATEUSZ MIKUSZ** — Lancaster University

## DEMOS CHAIRS

**TERESA ONORATI** — Universidad Carlos III de Madrid
**BASTIAN PFLEGING** — University of Stuttgart

## WORKSHOPS & TUTORIALS CHAIRS

**SALVATORE ANDOLINA** — Aalto University
**VILLE MÄKELÄ** — University of Tampere

## WEB CHAIR

**SARAH PRANGE**
University of Appl. Sciences Munich

## PUBLICITY CHAIR

**PASSANT ELAGROUDY**
University of Stuttgart

## PROCEEDINGS CHAIRS

**GIUSEPPE DESOLDA** — Università degli Studi di Bari
**UWE GRÜNEFELD** — University of Oldenburg

## WE ARE INTERESTED IN ...

- Applications and experience reports
- New forms or interactions
- Media façades
- Content design and visualization
- Research methods
- Art installations
- Audience behavior
- Usable security
- Systems architectures and frameworks

## TECHNICAL PAPERS

Submit solid contributions in one of our interest areas or related domains

Submission DL: Feb 07
Notification DL: Mar 28

## POSTERS & DEMOS

Submit proactive work in progress you want to get feedback about

Submission DL: Apr 09
Notification DL: Apr 22

## WORKSHOPS & TUTORIALS

Build a community around an emerging relevant topic or teach us about an exciting topic

Submission DL: Apr 09
Notification DL: Apr 22

www.pervasivedisplays.org

acm SIGMOBILE

Science | DOI:10.1145/3283206 Chris Edwards

# Learning to See

*Machine learning turns the spotlight on elusive viruses.*

How do you look for a needle in a haystack, when you are not sure what the needle looks like? This is the problem that faces scientists as they try to deal with increasingly complex datasets. One answer is to turn machine learning loose on the enormous volumes of data they have captured.

The problem of finding relevant data in genetic databases is one that Simon Roux, a researcher working at the U.S. Department of Energy's Joint Genome Institute, faced when investigating the role that an obscure and little-understood family of viruses plays in the environment.

There are many types of virus, called bacteriophages, that infect bacteria. Many of these either kill their hosts or are themselves rejected by an immune response. The bacteriophages that belong to the family inoviridae can remain in the host for long periods. This property has helped make one such "inovirus," known as M13, a popular choice among bioengineering researchers. The needle-shaped M13 infects the *Escherichia coli* (*E. coli*) bacterium, an organism that is very easy to cultivate under laboratory conditions. When the bacteria expel the virus particles they are forced to make by the viral DNA, the particles are available in large numbers and can be purified

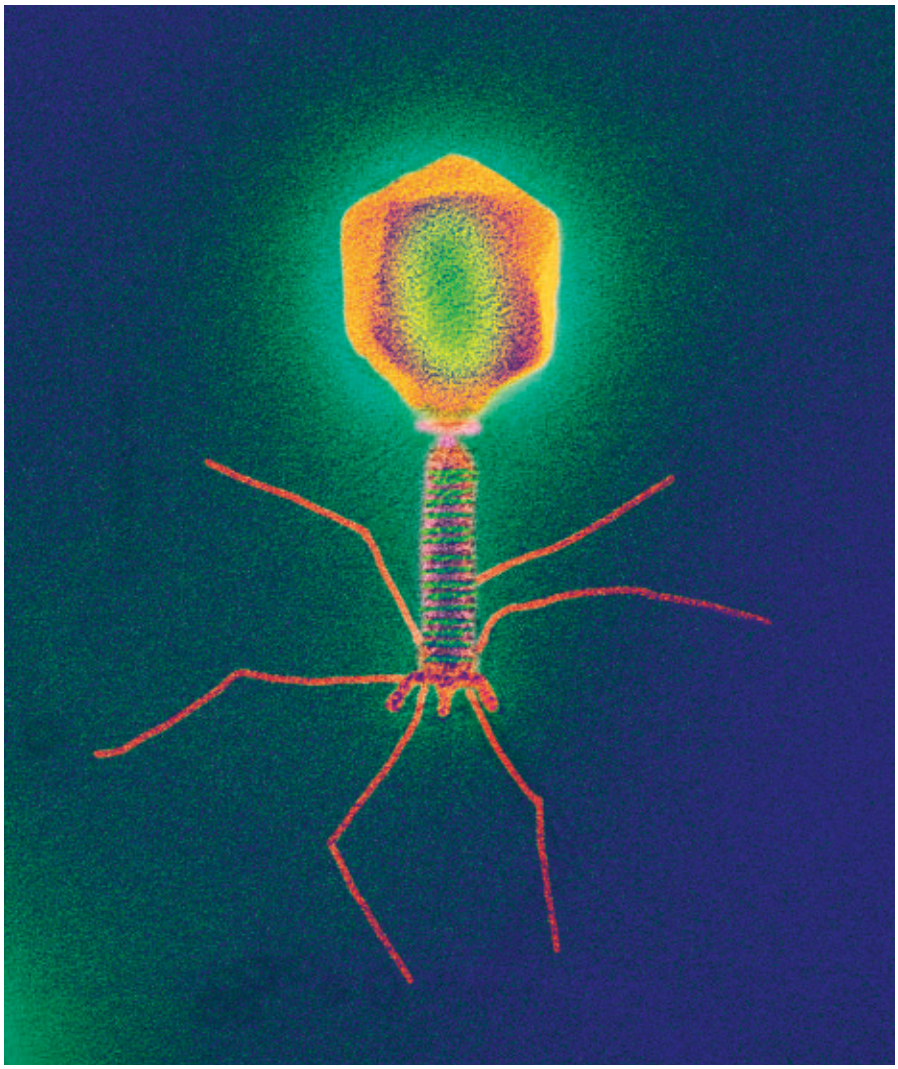


IMAGE BY DEPT. OF MICROBIOLOGY, BIOZENTRUM/SCIENCE PHOTO LIBRARY

**Colored transmission electron micrograph of a T4 bacteriophage virus, magnified 100,000 times.**

easily, chemically treated to sterilize them, and then formed into artificial structures. A team at the Massachusetts Institute of Technology (MIT) led by Angela Belcher has used M13 scaffolds to make electrical batteries. Seung-Wuk Lee of the University of California, Berkeley (UC Berkeley) has used the genetically engineered versions of the same inovirus to create piezoelectric generators.

Members of inoviridae can show a much darker side, too. One inovirus has been found to make cholera bacteria much more deadly. Says Roux, "You might think, as we have these viruses with great applications and others with a big impact, we must know a lot about them. But we don't."

There are fewer than 100 confirmed species of inovirus. They even seem to elude methods that were developed specifically to find and identify novel species of microorganism. One such technique, the meta-genomic survey, takes advantage of the high-speed "next-generation" gene-sequencing (NGS) hardware now available to biologists. Derived from the "shotgun" sequencing used on the Human Genome Project more than 15 years ago, NGS makes it possible to reconstruct genomes from multiple species that may be contained in a single sample, instead of trying to isolate first the DNA of each organism.

The first step is to shred DNA extracted from a biological sample before using enzymes to make enough copies for sequencing. High-performance computers then attempt to piece together the resulting jigsaw into longer sequences. The algorithms do this by aligning segments that appear to overlap before assembling them into different candidate genomes. Normally, in a metagenomic survey, researchers hand-check the results to try to weed out false matches.

With bacteria and higher organisms, it is relatively straightforward to ensure that each genome represents a single species. One commonly employed technique looks for variations of one or two essential large genes. Because these particular genes are fundamental to the survival of the organism, such genes exhibit relatively minor deviations across species, and organisms from the same family will have common changes that are not seen in more distant relatives.

In some cases, metagenomics has revealed thousands of previously unknown organisms lurking in samples from a single location. A group led by Jill Banfield at UC Berkeley took samples from sediment beds at an abandoned uranium mine in Colorado in 2015. From those samples, NGS and computer analysis coupled with manual curation reconstructed more than 2,500 partial and complete genomes, and found among them were nearly 50 new families of bacteria. Further work led to the team proposing a new "tree of life" they believe better explains the evolutionary relationships between microorganisms than traditional models.

For both bacteria and viruses, metagenomic surveys have produced genomes suitable for study without demanding that each species be cultured in the lab. For many species, that is impossible using current techniques. Viruses present a significant problem as they are closely associated with their hosts and do not grow in isolation.

Siddharth Krishnamurthy, a researcher at the Washington University School of Medicine in St. Louis, says, "Without these large genomic databases and algorithmic approaches to populate them, we would be unaware of whole families of viruses that have never been cultured."

Yet within these databases, members of the *inoviridae* family are suspiciously absent. Roux's hunch was that inoviruses are commonly found in the environment and that detection was the main problem. It seemed traditional genome-identification and binning tactics do not work well on them. One possibility was to use a tool called VirSorter, developed at the University of Arizona when Roux worked there. This software looks for characteristic nucleotide patterns in genomes, such as sequences that code the protein shells in which viruses wrap their DNA payloads for transport to new victims.

**"In principle, if we knew the sequence of every virus on the planet, there would be no value in using a machine learning algorithm for virus identification."**

"This work started when we realized that these viruses were missed by the probabilistic techniques used in VirSorter. The short story is that these inovirus genomes are too short and their genes are too variable for a VirSorter-like approach to identify," Roux says.

One approach that some groups have tried is to look at the statistical composition of the many tiny fragments of DNA that the sequencer reads. Although the reasons why are not yet understood, analysis of known viral genomes has shown that closely related genomes show a bias in the way nucleotides are used even in short sequences, known as k-mers.

The DiscoVir tool developed by Krishnamurthy and colleagues uses machine learning trained on k-mer data to sift, from bacterial and fungal material in metagenomic surveys, the genomes of unidentified viruses that infect plants and animals, rather than bacteria. Machine learning makes it possible to use features that do not rely on similarity to known genetic sequences and apply rules that are more likely to find virus candidates.

"In principle, if we knew the sequence of every virus on the planet, there would be no value in using a machine learning algorithm for virus identification," Krishnamurthy says. "The greatest asset that I believe machine learning brings to viral identification is the ability of these algorithms to identify different combinations of variables that can lead to the positive prediction of a virus.

"Things like support vector machines and random forests don't require all viruses to have the same properties. This is an important feature of viral classification because biologically, there are no molecular attributes that are specific to all viruses that are not present in any non-viruses, which is one of the reasons why it's so hard to

give an all-encompassing definition of a virus," Krishnamurthy adds.

Roux and colleagues used a different set of features in the machine-learning algorithms they used to find their missing inoviruses. He explains, "We manually identified a set of 10 'fuzzy' but distinctive features of *inoviridae*. None of these features is individually a clear sign, but some combinations of typically four or five of them is usually a great indicator."

Roux's group tried a number of machine learning algorithms, but found that a random-forest classifier provided the best results. Deep learning methods could not be used, because the training set was too small.

Armed with a way of finding inoviruses in existing sequence data, the results were startling. The software identified thousands of probable inovirus genomes, many of which could be classified into six broad families. The team removed by hand around 70 sequences that did not seem to be inovirus genomes or were too unusual to considered viable candidates.

"This doesn't guarantee that every sequence we highlighted is an inovirus genome, but we feel confident that we do not have a large subset of our sequences that do not represent plausible inoviruses," Roux says. "Beyond these *in silico* analyses, the real 'proof' will have to be done through lab experiments. But having looked at a lot of these sequences for the past year, I can tell that, to my eye, it really looks as though we found several thousand plausible inoviruses that had not been previously reported."

Some appear to infect bacteria that, up to now, were thought not to have any inoviruses associated with them, and appear to be far more numerous than the dozens listed in existing virus databases imply. "The *inoviridae* are a full viral order and associated with all types of bacteria. They are basically everywhere," Roux says.

Krishnamurthy sees an important role for machine learning alongside conventional techniques in the continuing quest to map the Earth's biological diversity. "They have the potential to work synergistically with their alignment-based counterparts. The minute an alignment-independent classifier finds one member virus of a novel family, alignment-based methods can be used to rapidly scan previously sequenced data to find more closely related members, so that the members of the viral family can be expanded."

**The software identified thousands of probable inovirus genomes, many of which could be classified into six broad families.**

In turn, as the genome databases expand and become more accurate, better training data becomes available to the machine learning models, which will let them, in Roux's view, find "gold in other people's data surplus." Similar work is likely to reveal more from the genome and other data that scientists have already obtained. He adds, "The 'omics approaches generate so much data that no one can look at every potentially interesting piece in their dataset". C


**Further Reading**

*Wooley, J.C., Godzik, A., and Friedberg, I.*
A Primer on Metagenomics,
*PLOS Computational Biology* 6(2): e1000667 (2010)

*Anantharaman, K., et. al.*
Thousands of Microbial Genomes Shed Light on Interconnected Biogeochemical Processes in an Aquifer System,
*Nature Communications* (2016) 7:13219

*Krishnamurthy, S. and Wang, D.*
Origins and Challenges of Viral Dark Matter,
*Virus Research*, 239 (2017) 136-142

*Roux, S., Hallam, S.J, Woyke, T., and Sullivan, M.B.*
Viral Dark Matter and Virus-Host Interactions Resolved from Publicly Available Microbial Genomes,
*ELife* (2015) 4:e08490



**Chris Edwards** is a Surrey, U.K.-based writer who reports on electronics, IT, and synthetic biology.




# ACM Member News

## SEEKING OUT HCI ACROSS BORDERS

**"'If the user can't use it, it doesn't work!' is my motto," says Susan M. Dray, a recognized leader in Human-Computer Interaction (HCI) and User Experience (UX).**

**Dray's career, and her interest in HCI, spans decades; she co-founded the ACM Special Interest Group on Computer–Human Interaction (SIGCHI) in 1982.**

**She earned her bachelor of arts degree in psychology from Mills College in Oakland, CA, and both her master's and Ph.D. degrees in psychology from the University of California, Los Angeles. After graduation, Dray started working in industrial research at multinational conglomerate Honeywell. She spent nine years there before moving to American Express, where she consulted internally on IT system design and the impact of technology on the organization.**

**Dray eventually launched her own company, Dray & Associates, a Minneapolis, MN-based user experience consulting firm with clients in the Fortune 500, including Hewlett Packard and Microsoft.**

**Currently serving SIGCHI as Vice President at Large, Dray—who has had the pleasure of working in 28 countries—has been leading "SIGCHI Across Borders," an initiative to reach out to people who work in HCI, whether they are academics, researchers, or practitioners, to identify what these communities need, particularly in countries where HCI is still in its infancy.**

**Dray spent June 2018 sharing her HCI and UX knowledge at Swansea University in the U.K., as part of a residency at the university's Computational Foundry. After returning to the U.S., she took a four-month semi-sabbatical to sail up the East Coast with her husband.**

***—John Delaney***

Technology | DOI:10.1145/3283224 Keith Kirkpatrick

# Technology for the Deaf

*Why aren't better assistive technologies available for those communicating using ASL?*

A NURSE ASKS a patient to describe her symptoms. A fast-food worker greets a customer and asks for his order. A tourist asks a police officer for directions to a local point of interest.

For those with all of their physical faculties intact, each of these scenarios can be viewed as a routine occurrence of everyday life, as they are able to easily and efficiently interact without any assistance. However, each of these interactions are significantly more difficult when a person is deaf, and must rely on the use of sign language to communicate.

In a perfect world, a person that is well-versed in communicating via sign language would be available at all times and at all places to communicate with a deaf person, particularly in settings there is a safety, convenience, or legal imperative to ensure real-time, accurate communication. However, it is exceptionally challenging, from both a logistical and cost perspective, to have a signer available at all times and in all places.

That's why, in many cases, sign language interpreting services are provided by Video Remote Interpreting, which uses a live interpreter that is connected to the person needing sign language services via a videoconferencing link. Institutions such as hospitals, clinics, and courts often prefer to use these services, because they can save money (interpreters not only bill for the actual translation service, but for the time and expenses incurred traveling to and from a job).

However, video interpreters sometimes do not match the accuracy of live interpreters, says Howard Rosenblum, CEO of the National Association of the Deaf (NAD), the self-described "premier civil rights organization of, by, and for deaf and hard of hearing individuals in the United States of America."

"This technology has failed too often to provide effective communications, and the stakes are higher in hospital and court settings," Rosenblum says, noting that "for in-person communications, sometimes technology is more of an impediment than a solution." Indeed, technical issues such as slow or intermittent network bandwidth often make the interpreting experience choppy, resulting in confusion or misunderstanding between the interpreter and the deaf person.

That's why researchers have been seeking ways in which a more effective technological solution or tool might handle the conversion of sign language to speech, which would be useful for a deaf person to communicate with a person who does not understand sign language, either via an audio solution or a visual, text-based solution. Similarly, there is a desire to allow real-time, audio-based speech or text to be delivered to a person who is deaf, often through sign language, via a portable device that can be carried and used at any time.

Nonetheless, sign languages, such as the commonly used American Sign Language (ASL), are able to convey words, phrases, and sentences through a complex combination of hand movements and positions, which are then augmented by facial expressions and body gestures. The result is a complex communication system that requires a combination of sensors, natural language processing, speech recognition technology, and machine learning technology, in order to capture and process words or phrases.

**These prototype SignAloud gloves translate the gestures of American Sign Language into spoken English.**

PHOTO BY DENNIS WISE/UNIVERSITY OF WASHINGTON

One system designed to allow people fluent in ASL to communicate with non-signers is SignAloud, which was developed in 2016 by a pair of University of Washington undergraduate students. The system consists of a pair of gloves that are designed to recognize the hand gestures that correspond to words and phrases used in American Sign Language (ASL).

Worn by the signer, each glove is fitted with motion-capture sensors that record the position and movements of the hand wearing it, then sends that data to a central computer via a wireless Bluetooth link. The data is fed through various sequential statistical regressions, which are similar to a neural network, for processing. When the data matches an ASL gesture, the associated word or phrase is spoken through a speaker. The idea is to allow for real-time translation of ASL into spoken English.

Despite the promise of SignAloud, whose inventors received the $10,000 Lemelson-MIT Student Prize, there was significant criticism of the product from the deaf community, who complained that SignAloud did not capture the nuances of sign language, which relies on secondary signals such as eyebrow movements, shifts in the signer's body, and motions of the mouth, to fully convey meaning and intent. Furthermore, strict word-for-word translations of ASL, like other languages, often results in an inaccurate translation, as each language requires sentence structure and context in order to make sense.

That has not stopped other companies from developing similar products, such as the BrightSign Glove, developed by Hadeel Ayoub as a relatively inexpensive (pricing is expected to be in the hundreds-of-dollars range) way to allow two-way communication between those who sign and those who do not. BrightSign's technology is slightly different than SignAloud; users record and name their own gestures to correspond with specific words or phrases, thereby ensuring that the lack of facial cues or body motions will not impact meaning. As a result, BrightSign users can take advantage of a 97% accuracy rate when using the gloves.

BrightSign is readying several versions of the glove for commercialization, including a version aimed at children, with a substantial wristband with its own embedded screen and audio output. Another version, targeted at the adult deaf community, can send translations directly to the wearer's smartphone, which can then enunciate the words or phrases.

**"The challenge is that every person signs with their own flair and nuance, just like every person has a different sound or inflection on how they pronounce certain words."**

The company says it has about 700 customers on its preorder list, and is trying to secure about $1.4 million in capital from investors, which would allow the company to fulfill all existing preorders.

Other tools are being developed to address the technological challenges of translating ASL to speech, although the complexity of ASL and other sign languages present significant technological challenges to handle these tasks in real time, which is needed to ensure smooth communication.

"There are several companies that are developing software and databases, including with the use of AI and machine learning, to create programs on computers that can 'read' a person that is signing in ASL," Rosenblum says, noting that these tools not only read hand-signing, but also capture facial cues and other visible markers. Using cameras to capture these signs and cues, the systems then use machine learning to identify and recognize specific movements and gestures, and then match them to specific words or phrases which can then be sent to a speech or text generator that can be read or heard by a non-signing individual.

"However, the challenge is that every person signs with their own flair and nuance, just like every person has a different sound or inflection on how they pronounce certain words," Rosenblum says. To manage the variances in the way people sign, videos of people signing must be input and processed by a machine learning algorithm to train the system to account for these stylistic variances. As such, the systems need lots of time and data in order to improve accuracy.

Another major issue is allowing people who don't sign to communicate in real time with those who do sign. One application that appears to be functioning well enough for some users to utilize today is Hand Talk. This app allows a non-signer to input words and phrases by speaking to the app located on a deaf person's phone. The app engine translates the words in real time into Libras, the sign language used in Brazil. Then, an animated avatar known as Hugo will begin signing on the deaf person's smartphone screen.

Unlike other apps that are using machine learning to train an algorithm, Hand Talk's founder Ronaldo Tenorio and his team program thousands of example sentences every month and match them with three-dimensional (3D) animations of sign language, including Hugo's facial expressions, which carry meaning in sign language. Improvements to the application are pushed out through regular app updates.

According to the company, the app handles six million monthly translations on Hand Talk, and has reached one million downloads, approximately one-sixth of Brazil's deaf population.

Still, for applications that will be useful across a wide range of languages, cultures, and situations, developers likely will need to use machine learning algorithms to learn all the possible variations, nuances, and cadences of conversational sign language. Further, ASL and other sign languages are very complex, with signs bleeding into one another, anticipating the shape or location of the following sign, which is similar to how some spoken sounds take on the characteristics of adjacent sounds. As such, Rosenblum says, "the capacity or development of computers being able to "read" the zillions of variations of rendering ASL is extremely difficult and probably will take a decade to accomplish."

A key reason why even advanced technologies that use machine learning to train and ingest the many variations of sign language do not work as seamlessly as a live signer is due to the lack of participation of deaf or hard-of-hearing people in the development process, thereby missing key linguistic, stylistic, and usability concerns of signers.

"That's a huge problem," Rosenblum says. "Several companies do have deaf and hard-of-hearing engineers, scientists, or other highly trained professionals, but this is more of an exception than the rule."

Perhaps the biggest reason why technology for the deaf is not as functional as it could be is because technology is driven, in large part, by the lack of regulatory requirements covering non-signer to signer communications, and vice versa. Improvements in accessibility within the television and video industries was driven by regulation, and may serve as an example of how real-time communications may eventually be regulated.

"For individuals with hearing loss, videos need captioning or a transcript of what is verbally communicated," says Nancy Kastl, Testing Practice Director at the digital technology consulting firm, SPR. "For individuals with vision loss, the captioning or transcript (readable by a screen reader) should include a description of the scenes or actions, if there are segments with music only or no dialogue."

**"The capacity or development of computers to 'read' the zillions of variations of rendering ASL is extremely difficult, and probably will take a decade to accomplish."**

Likewise, Rosenblum says that "many of the best advances in technology for deaf and hard of hearing people have been because laws demanded them," noting that the text and video relay systems provided by telecommunications companies were very basic and voluntary prior to the adoption of the Americans with Disabilities Act (ADA) of 1990.

Furthermore, the closed captioning of television content for the hearing impaired "in the original analog format was mandated by the Telecommunications Act of 1996, and expanded to digital access online through the 21st Century Communications and Video Accessibility Act of 2010, as well as by the lawsuit of NAD v. Netflix in 2012," Rosenblum says, noting that the suit required Netflix to ensure that 100% of its streaming content is made available with closed captions for the hearing impaired.


**Further Reading**

*Cooper, H., Holt, B., and Bowden, R.*
Sign Language Recognition,
*Visual Analysis of Humans*, 2011
http://info.ee.surrey.ac.uk/Personal/H.M/research/papers/SLR-LAP.pdf

*Erard, M.*
Why Sign Language Gloves Don't Help Deaf People, *The Atlantic*, November 9, 2017,
https://www.theatlantic.com/technology/archive/2017/11/why-sign-language-gloves-dont-help-deaf-people/545441/

25 Basic ASL Signs For Beginners,
American Sign Language Institute,
Oct. 22, 2016, https://www.youtube.com/watch?v=Raa0vBXA8OQ


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



ACM News

# Detecting Illness by Watching You Type

**Researchers are experimenting with artificial intelligence (AI) software that can tell whether you suffer from Parkinson's disease, schizophrenia, depression, or other mental disorders, from watching the way you type.**

**In a University of Texas study published earlier this year, for example, researchers were able to identify typists suffering from Parkinson's disease by capturing how study subjects worked a keyboard over time, then running that data through pattern-finding AI software.**

**"We envision a future where keystroke and touch-screen tracking will become a standard metric in any digital device and added to your electronic medical record," says Teresa Arroyo Gallego, a co-author of the study.**

**Meanwhile, researchers involved in similar work at Palo Alto, CA-based healthcare innovation company Mindstrong Health say they've been able to diagnose schizophrenia by analyzing typing keystroke patterns, as well looking closely at scrolling, swiping, and tapping behaviors.**

**"We believe that digital biomarkers are the foundation for measurement-based mental health care, for which there is a massive unmet patient need," says Mindstrong Health founder and CEO Paul Dagum.**

**Researchers at Hillsborough, CA-based NeuraMetrix are using keystroke analysis to detect afflictions including Alzheimer's disease, depression, Huntington's disease, and REM sleep disorder.**

**In the Texas study, researchers say they engineered software that could capture down to the millisecond how long a typist held down a key before moving to the next key, as well as capturing 'flight time'—the number of milliseconds it takes a typist to actually move a finger from one key to the next.**

**Armed with that data, diagnosing typists with Parkinson's was just a matter of training the AI software to find typing patterns shared by people suffering from the disease, then running new data through the trained AI software to find matches, which they did.**

**A great advantage of the Texas researchers' diagnostic method is its sheer convenience. Patients can work with their smartphones and other digital devices as usual, and software installed on those devices will transmit their use-history over the Internet to the computers of researchers.**

**Even better, the Texas researchers' work appears to diagnose Parkinson's disease much earlier than usual.**

**Says Timothy Ellmore, an associate professor in psychology at the City College of New York, "The data from these keyboard tracking techniques need further validation to objectively track progression of Parkinson's signs." Still, he says, "Looking ahead, the tool could be really useful in augmenting the current tools available to clinicians."**

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

Society | DOI:10.1145/3283222 Logan Kugler

# AI Judges and Juries

*Artificial intelligence is changing the legal industry.*

When the head of the U.S. Supreme Court says artificial intelligence (AI) is having a significant impact on how the legal system in this country works, you pay attention. That's exactly what happened when Chief Justice John Roberts was asked the following question:

"Can you foresee a day when smart machines, driven with artificial intelligences, will assist with courtroom fact-finding or, more controversially even, judicial decision-making?"

His answer startled the audience.

"It's a day that's here and it's putting a significant strain on how the judiciary goes about doing things," he said, as reported by *The New York Times*.

In the last decade, the field of AI has experienced a renaissance. The field was long in the grip of an "AI winter," in which progress and funding dried up for decades, but technological breakthroughs in AI's power and accuracy changed all that. Today, giants like Google, Microsoft, and Amazon rely on AI to power their current and future profit centers.

Yet AI isn't just affecting tech giants and cutting-edge startups; it is transforming one of the oldest disciplines on the planet: the application of the law.

AI is already used to analyze documents and data during the legal discovery process, thanks to its ability to parse through millions of words faster (and more cheaply) than human beings. That alone could automate away or completely change the almost 300,000 paralegal and legal assistant jobs estimated to exist by the U.S. Bureau of Labor Statistics. However, that is just the beginning of AI's potential impact; it also is being used today to influence the outcomes of actual cases.

In one high-profile 2017 case, a man named Eric Loomis was sentenced to six years in prison thanks, in part, to recommendations from AI algorithms. The system analyzed data about Loomis and made sentencing recommendations to a human judge on the suggested length of Loomis' sentence.

Make no mistake: AI-enhanced courtrooms may be more science fact than science fiction—for better or for worse.

## The Predictable, Reliable Choice?

Artificial intelligence holds some promise for the world of legal decisions.

In Canada, Randy Goebel, a professor in the computer science department of the University of Alberta working in conjunction with Japanese researchers, developed an algorithm that can pass the Japanese bar exam. Now, the team is working to develop AI that can "weigh contradicting legal evidence, rule on cases, and predict the outcomes of future trials," according to Canadian broadcaster CBC. The goal is to use machines to help humans make better legal decisions.

This is already being attempted in U.S. courtrooms. In the Loomis case, AI was used to evaluate individual defendants. The algorithm used was created and built into software called Compas by a company called Northpointe Inc. The algorithm indicated Loomis had "a high risk of violence, high risk of recidivism, [and] high pretrial risk." This influenced the six-year sentence he received, though the sentencing judges were advised to take note of the algorithm's limitations.

Criminal justice algorithms like the one in the Loomis case use personal data such as age, sex, and employment history to recommend sentencing, reports the Electronic Privacy Information Center (EPIC). The technology is relatively common in the U.S. legal system.

"Criminal justice algorithms are used across the country, but the specific tools differ by state or even county," says EPIC.

The case for using AI-based systems to assist in the legal process hinges on the perceived ability of machines to be more impartial than humans. "Humans can be swayed by emotion. Humans can be convinced. Humans get tired or have a bad day," says Tracy Greenwood, an expert in e-discovery, the process of using machines to per-

IMAGE BY ANDREY POPOV

form legal discovery work faster and more accurately than humans.

"In a high crime city, a judge might start to hand out harsher sentences towards the upper end of the sentencing guidelines. In court, if a judge does not like one of the lawyers, that can affect the judge's opinion," says Greenwood.

The argument is that machines could potentially analyze facts and influence judgments dispassionately, without human bias, irrationality, or mistakes creeping into the process.

For instance, the Japanese bar exam AI developed by Goebel and his team is now considered "a world leader in the field," according to CBC. It succeeded on the exam where at least one human failed: one of Goebel's colleagues failed the Japanese bar exam.

Human fallibility is not an isolated problem in the legal field. According to an investigation by U.K.-based newspaper *The Guardian*, local, state, and federal courts in the U.S. are rife with judges who "routinely hide their connections to litigants and their lawyers." The investigation learned that oversight bodies found wrongdoing and issued disciplinary action in nearly half (47%) of complaints about judge conflict of interest they investigated.

However, oversight bodies rarely look into complaints at all—90% of over 37,000 complaints investigated were dismissed by state court authorities "without conducting any substantive inquiry," according to the investigation.

Conflict of interest is not the only human bias that plagues the U.S. legal system; racial bias, explicit or implicit, also is common.

"Minorities have less access to the courts to begin with, and tend to have worse outcomes due to systemic factors limiting their quality of representation, and subconscious or conscious bias," says Oliver Pulleyblank, founder of Vancouver, British Columbia-based legal firm Pulleyblank Law.

Intelligent machines, however, do not carry the same baggage. Acting as dispassionate arbiters looking at "just the facts," machines hold the potential to influence the legal decision-making process in a more consistent, standardized way than humans do.

The benefits would be significant.

"To introduce a system with much greater certainty and predictability would open up the law to many more people," says Pulleyblank. The high cost and uncertain outcomes of cases discourage many from pursuing valid legal action.

"Very few people can afford to litigate matters," says Pulleyblank, "even those who can generally shouldn't, because legal victories are so often hollow after all the expenses have been paid."

However, when you look more deeply at machine-assisted legal decisions, you find they may not be as impartial or consistent as they seem.

### "Unbiased" Machines Created by Biased Humans

In the Loomis algorithm-assisted case, the defendant claimed the algorithm's report violated his right to due process, but there was no way to examine how the report was generated; the company that produces the Compas software containing the algorithm, Northpointe, keeps its workings under wraps.

"The key to our product is the algorithms, and they're proprietary. We've created them, and we don't release them because it's certainly a core piece of our business," Northpointe executives were reported as saying by *The New York Times*.

This is the so-called "black box" problem that haunts the field of artificial intelligence.

Algorithms are applied to massive datasets. The algorithms produce results based upon their "secret sauce"—how they use the data. Giving up the secret sauce of an algorithm is akin to giving up your entire competitive advantage.

The result? Most systems that use AI are completely opaque to anyone except their creators. We are unable to determine why an algorithm produced a specific output, recommendation, or assessment.

**"To introduce a system with much greater certainty and predictability would open up the law to many more people."**

This is a major problem when it comes to using machines as judge and jury: because we lack even the most basic understanding of how the algorithms work, we cannot know if they are producing poor results until *after* the damage is done.

ProPublica, an "independent, nonprofit newsroom that produces investigative journalism with moral force," according to its website, studied the "risk scores," assessments created by Northpointe's algorithm, of 7,000 people who were arrested in Broward County, FL. These scores are used to determine release dates and bails in courtrooms, as they purportedly predict the defendant's likelihood to commit crime again.

As it turns out, these algorithms may be biased.

In the cases investigated, ProPublica says the algorithms wrongly labeled black defendants as future criminals at a rate nearly twice that of white defendants (who were mislabeled as "low risk" more often than black defendants).

Because the algorithms do not operate transparently, it is difficult to tell if this was an assessment error, or if the algorithms were coded with rules that reflect the biases of the people who created them.

In addition to bias, the algorithms' predictions just are not that accurate.

"Only 20% of the people predicted to commit violent crimes actually went on to do so," says ProPublica. Fewer violent crimes committed is a good thing, but based on this assessment, decisions were made that treated 80% of defendants as likely violent criminals when they were not.

Critics claim that algorithms need to be far more transparent before they can be relied on to influence legal decisions.

Even then, another huge problem with having AI take on a larger role in the legal system is that there is no guarantee machines can handle the nuances of the law effectively, says Pulleyblank.

"Many legal problems require judges to balance distinct interests against

each other," he says. He cites the example of a sexual assault victim bringing a case against their attacker. The judge is required to balance the victim's need for privacy with the principle that justice should take place in the open for all to see. There's no easy answer, but the decision to publish the victim's name or keep the proceedings behind closed doors is one a judge has to make—and one that has major effects on the case.

"What it depends on is not 'the law'," says Pulleyblank. "There is no clear legal answer to how those values will be balanced in any given case. Rather, it depends on the judge."

These types of contextual considerations crop up constantly in all manner of cases. "Machines are good at identifying what has been tried and what has not been tried, but they lack judgment," says Greenwood. He says machines may produce consistent results, but lack other critical skills to ensure justice is served. "A machine will not lecture a defendant in a criminal case and tell him to get his life together."

Pulleyblank agrees that making the law more "predictable" using machines may cause more problems than it solves. "Whenever you seek to make the law more predictable, you risk sacrificing fairness," he says.

In ProPublica's investigation, the algorithm assessed two defendants. One was a seasoned criminal; the other a young girl with a prior misdemeanor. Both had stolen items of the same value, but the machine failed to contextualize the fact that the young girl had stolen a bicycle and had no serious criminal record. She was deemed a likely repeat offender, just like the career criminal. To the machine, these two people had both committed crimes and had past charges. It failed to contextualize; as a result, the algorithm used in this case got the situation very wrong.

Yet introducing context and circumstance inherently reduces the predictability and consistency of the law's application, so the balance between machine predictability and human judgment is a tenuous one.

"This is the order versus fairness dichotomy that has long been the subject of legal thought," says Pulleyblank.

This leads both Pulleyblank and Greenwood to the same conclusion: machines probably will come to heavily assist humans in the legal profession. The industry will transform as a result, but to completely replace humans in the legal process would likely require changing the law itself.

**"In order to allow predictable non-human judicial decisions, the law would have to change in a fairly fundamental way."**

"In order to allow predictable non-human judicial decisions, the law would have to change in a fairly fundamental way," says Pulleyblank, "and if the law does not change, there is simply too much discretion inherent in the law as it exists for the public to accept that discretion being exercised by machines."

While machines might have superior predictive power, humans will issue the final verdict on their use.


**Further Reading**

*Liptak, A.*
Sent to Prison by a Software Program's Secret Algorithms, *The New York Times*, May 1, 2017, https://www.nytimes.com/2017/05/01/us/politics/sent-to-prison-by-a-software-programs-secret-algorithms.html

*Angwin, J.*
Machine Bias, *ProPublica*, May 26, 2016 https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Algorithms in the Criminal Justice System, *Electronic Privacy Information Center*, https://epic.org/algorithmic-transparency/crim-justice/

*Snowden, W.*
Robot judges? Edmonton research crafting artificial intelligence for courts, *CBC*, Sept. 19. 2017, http://www.cbc.ca/news/canada/edmonton/legal-artificial-intelligence-alberta-japan-1.4296763



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




Milestones

# Estrin Awarded MacArthur 'Genius Grant'

**Cornell Tech computer science professor Deborah Estrin was among the 25 people named 2018 MacArthur Fellows. The MacArthur Foundation said Estrin was chosen for "designing open source platforms that leverage mobile devices and data to address socio-technological challenges such as personal health management."**

**Estrin, 58, earned her bachelor of science degree from the University of California at Berkeley, and master of science and doctorate degrees from the Massachusetts Institute of Technology. She taught at the University of Southern California and the University of California at Los Angeles prior to joining the faculty of Cornell Tech in 2012, where she is a professor in the Department of Computer Science and associate dean.**

**Estrin was named an ACM fellow in 2000 "for significant contributions to the design of scalable Internet protocols, and for service to the networking community." In 2006, she received the ACM Athena Lecturer Award, bestowed annually in celebration of women researchers who have made fundamental contributions to computer science.**

**The MacArthur Foundation said Estrin "has demonstrated a remarkable ability to anticipate the applicability of technological advances to a variety of fields. She made fundamental contributions to improving the scalability and broader utility of the emerging Internet through her work on network routing (the process that determines how data is forwarded from source to destination). She then went on to build the foundational protocols for wireless sensor networks—that is, connectivity among distributed autonomous sensors that record conditions in a specific environment. "**

**"I was and remain very humbled and grateful," said Estrin. "I feel a sense of commitment to do good by it, and to live up to it."**

# 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 70 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,

Cherri M. Pancake
President
Association for Computing Machinery

Association for Computing Machinery

*Advancing Computing as a Science & Profession*

# SHAPE THE FUTURE OF COMPUTING.
# JOIN ACM TODAY.

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

## SELECT ONE MEMBERSHIP OPTION

### ACM PROFESSIONAL MEMBERSHIP:

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

### ACM STUDENT MEMBERSHIP:

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

❑ **Join ACM-W:** ACM-W supports, celebrates, and advocates internationally for the full engagement of women in computing. Membership in ACM-W is open to all ACM members and is free of charge.

**Priority Code: CAPP**

### Payment Information

Name

ACM Member #

Mailing Address

City/State/Province

ZIP/Postal Code/Country

❑ Please do not release my postal address to third parties

Email

❑ Yes, please send me ACM Announcements via email

❑ No, please do not send me ACM Announcements via email

### Purposes of ACM

ACM is dedicated to:

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

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

❑ AMEX ❑ VISA/MasterCard ❑ Check/money order

Total Amount Due

Credit Card #

Exp. Date

Signature

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

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

**Satisfaction Guaranteed!**

## BE CREATIVE. STAY CONNECTED. KEEP INVENTING.

Association for Computing Machinery

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

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

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

DOI:10.1145/3286868

Ted G. Lewis and Peter J. Denning

# The Profession of IT
# Learning Machine Learning

*A discussion of the rapidly evolving realm of machine learning.*

Machine learning has evolved from an out-of-favor subdiscipline of computer science and artificial intelligence (AI) to a leading-edge frontier of research in both AI and computer systems architecture. Over the past decade investments in both hardware and software for machine learning have risen at an exponential rate matched only by similar investments in blockchain technology. This column is a technology check for professionals in a Q&A format on how this field has evolved and what big questions it faces.

**Q: The modern surge in AI is powered by neural networks. When did the neural network field start? What was the first implementation?**

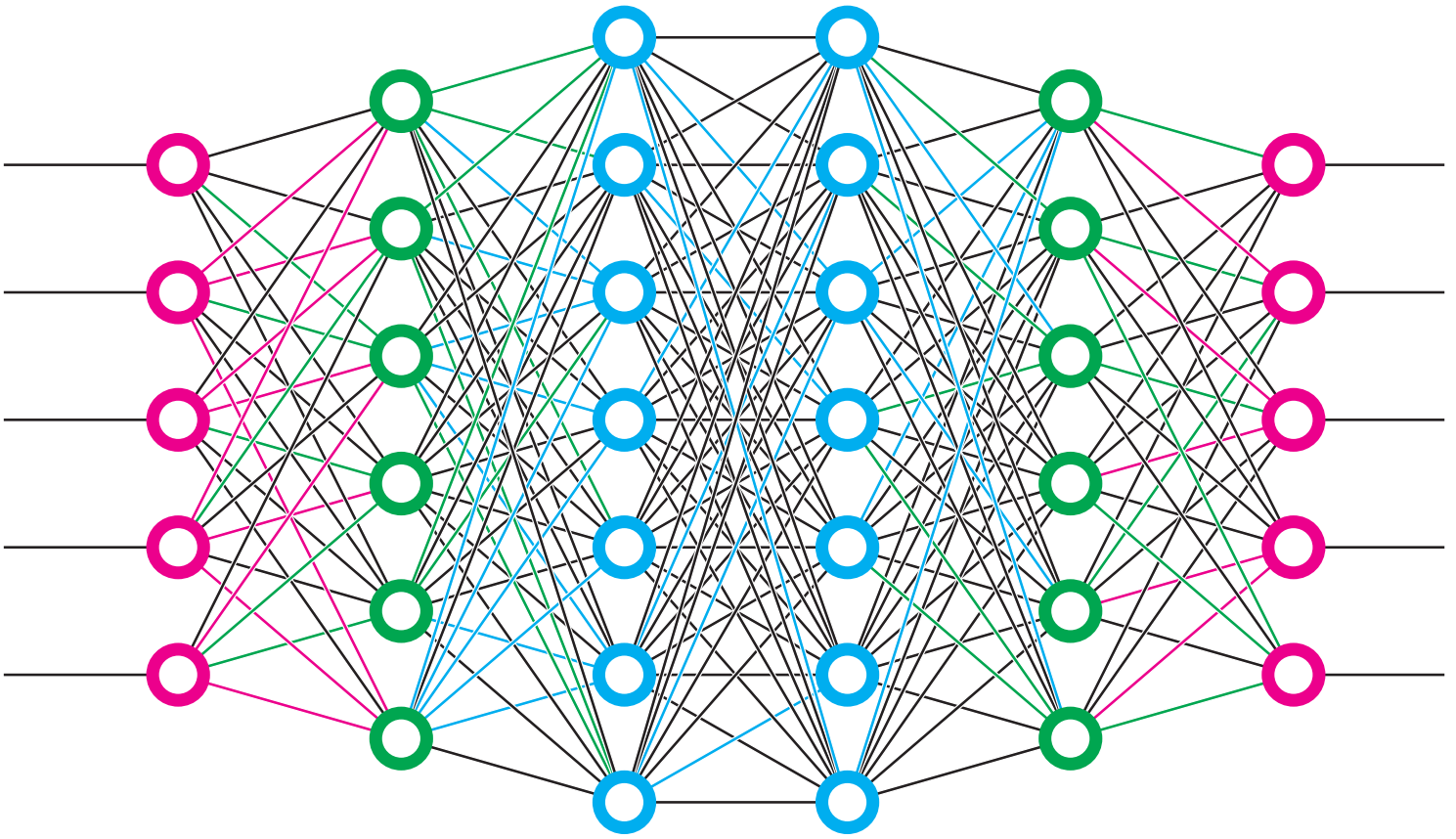
A. The early 1940s was a time of increasing attention to automatic computers. At the time, a "computer" was a professional job title for humans and computation was seen as a human intelligent activity. Some believed that the logical computations of the brain were made possible by the neuronal structure of the brain. In 1943 Warren McCulloch and Walter Pitts wrote a famous proposal to build computers whose components resembled neurons.[4] Each neuron received inputs from many others and delivered its outputs to many others. Inputs had weights and when the weighted input sum exceeded a threshold the neuron switched from the 0 to the 1 state. They wrote: "Because of the 'all-or-none' character of nervous activity, neural events and the relations among them can be treated by means of propositional logic." It was logical that a machine organized like the brain would be good with logic! McCulloch and Pitts established the foundation for future artificial neural networks.

**It takes an enormous amount of computation to train a large network on a large training set.**

In 1957, Frank Rosenblatt demonstrated the first artificial neural network machine for the U.S. Navy. He called it the Perceptron. It was a single-layer machine as illustrated schematically in the accompanying figure using photocells as input receptors organized as two-dimensional array. The Perceptron was able to recognize handwritten digits 0 through 9. The figure also outlines a genealogy of the neural network descendants of the Perceptron; we provide it for information, but we will not discuss all its branches here.

**Q: Neural networks are good for mapping input patterns into output patterns. What does this mean?**

A pattern is a very long sequence of bits, for example, the megabits making up an image or gigabits representing a person's viewing history of films. A recognizer or classifier network maps a pattern into another that has meaning to humans. A recognizer network can,



for example, take the bitmap from a camera shown the digit "9" and output the ASCII code for "9". A recommender network maps a pattern into a string representing an action the human might decide to take next. Netflix has developed a neural network that takes your entire film viewing history along with all the ratings you and others gave those items and returns a recommendation of a new film that you would probably rate highly.

**Q: How do you program a neural network?**

You don't. You teach it to learn how to do the function you want. Suppose you want to teach a network a function F that maps X patterns into Y patterns. You gather a large number of samples (X,Y), called the training set. For each sample you use a training algorithm to adjust the connection weights inside the network so that the network outputs Y when given X at its input. There are so many neurons, connections, and possible weights that the training algorithm can successfully embed a large number of pairs (X,Y) into the network. It takes an enormous amount of computation to train a large network on a large training set. We now have the hardware power and training algorithms to do this. The trained network will implement all the trained (X,Y) combinations very reliably.

Once the network is trained, it can compute its outputs very rapidly. It has a fixed function until its training is updated.

We want to use our networks to compute not only trained maps, but untrained ones as well. That means to compute Y=F(X) for a new pattern X not in the training set. An important question is how much trust can be put in the responses to untrained input patterns.

If we keep track of the new (untrained) inputs and their correct outputs, we can run the training algorithm again with the additional training pairs. The process of training is called learning, and of retraining reinforcement learning.

The network does not learn on its own. It depends on the training algorithm, which is in effect an automatic programmer.

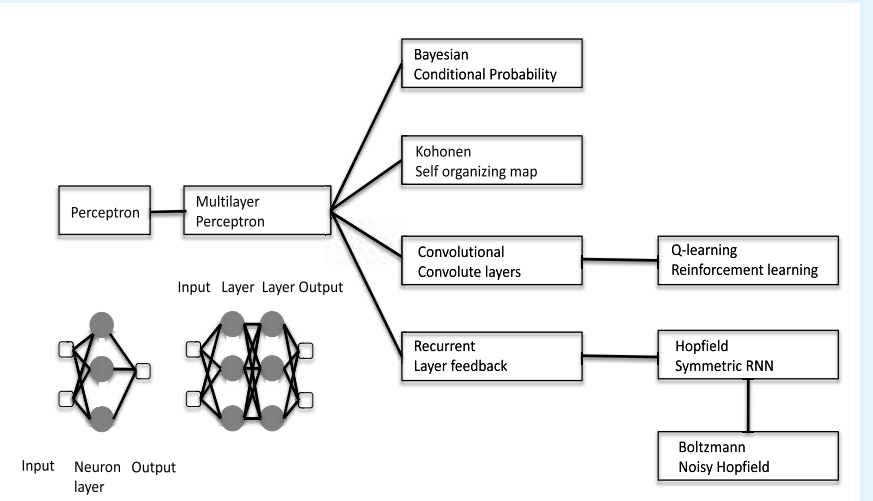
**Q: In 1969, Marvin Minsky and Seymour Papert published a book showing that Perceptrons could not recognize important patterns.[5] What effect did that have on the field?**

The entire field faltered following publication of the Minsky-Papert book. They proved that single-layered perceptrons would only work as classifiers when the data was "linearly separable"—meaning that the multidimensional data space could be separated into regions bounded by hyperplanes. Data not clustered in this way could not be recognized. This finding caused interest in perceptrons to die off, until researchers discovered that adding layers and feedback to the neural network overcame the problem.[1–3] The multilayered perceptron (MLP) can reliably classify patterns of data in nonlinear spaces. More layers mean more accuracy and, of course, more computation. The modern term "deep learning" acknowledges the many-layers depth of a neural network.

An open problem for research today is finding the smallest number of neurons and layers to implement a given function.

IMAGE FROM SHUTTERSTOCK.COM

**Figure 1. An abbreviated taxonomy of the evolution of neural networks shows a progression from simple one-layer Perceptron to multilayered Perceptron (MLP), convolutional and recurrent networks with memory and adaptable reinforcement learning algorithms.**




Back in favor because of the MLP breakthrough, neural networks advanced rapidly. We have gone well beyond recognizing numbers and handwriting, to networks that recognize and label faces in photographs. New methods have since been added that allow recognition in video moving images; see the figure for some of the keywords.

**Q: What propelled the advances?**

Two things. The abundance of data, especially from online social networks like Twitter and Facebook, large-scale sensor networks such as smartphones giving positional data for traffic maps, or searches for correlations between previously separate large databases. The questions that could be answered if we could process that data by recognizing and recommending were a very strong motivating force.

The other big factor is the proliferation of low-cost massively parallel hardware such as the Nvidia GPU (Graphics Processing Unit) used in graphics cards. GPUs rapidly process large matrices representing the positions of objects. They are super-fast linear-algebra machines. Training a network involves computing connection matrices and using a network involves evaluations of matrix multiplications. GPUs do these things really well.

**Q: These networks are now used for critical functions such as medical diagnosis or crowd surveillance to detect possible terrorists. Some military strategists talk about using them as automatic fire-control systems. How can we trust the networks?**

This is a hot question. We know that a network is quite reliable when its inputs come from its training set. But these critical systems will have inputs corresponding to new, often unanticipated situations. There are numerous examples where a network gives poor responses for untrained inputs. This is called the "fragility" problem. How can we know that the network's response will not cause a disaster or catastrophe?

The answer is we cannot. The "programs" learned by neural networks are in effect enormous, incomprehensible matrices of weights connecting millions of neurons. There is no more hope of figuring out what these weights mean or how they will respond to a new input than in looking for a person's motivations by scanning the brain's connections. We have no means to "explain" why a medical network reached a particular conclusion. At this point, we can only experiment with the network to see how it performs for untrained inputs, building our trust slowly and deliberately.

**Q: Computers were in the news 20 years ago for beating the grandmaster chess players, and today for beating the world's Go master champion. Do these advances signal a time when machines**

**can do all human mental tasks better than today's humans can?**

First, let's clarify a misconception about the IBM computer that beat chess grandmaster Garry Kasparov in 1997. It was not a neural network. It was a program to search and evaluate board positions much faster than Kasparov. In effect, the human neural network called "Kasparov" was beaten by the IBM computer using smart search algorithms. Kasparov bounced back with Advanced Chess in which humans assisted by computers played matches; the human teams often beat solo computers.

The neural network AlphaGo won four of five Go games against the world champion Le Se-dol. According to DeepMind researcher David Silver, "The most important idea in AlphaGo Zero is that it learns completely tabula rasa, that means it starts completely from a blank slate, and figures out for itself only from self-play and without any human knowledge."[a] AlphaGo Zero contains four CPUs and a single neural network and software that initially know nothing about Go or any other game. It learned to play Go without supervision by simply playing against itself. The results look "alien" to humans because they are often completely new: AlphaGo creates moves that humans have not discovered in more than 2,500 years of playing Go.

We think this is a development of singular significance. The time-honored method of neural networks learning by being trained from training sets of data can in some cases be replaced by machines learning from each other without any training data.

So far, there is little threat from these networks becoming superintelligent machines. The AlphaGo experience happened in a game with well-defined rules about allowable moves and a well-defined metric defining the winner. The game was played in a well-defined mathematical space. Presumably this training method could be extended to swarms of robots attacking and defending. But could it master a sport like basketball? Playing a violin? And what about games the purpose of which is to continue rather than to win?

**Q: Where do you see this going, next?**

The 10–15 year roadmap is pretty clear. There is now much theory behind neural networks. Even much of the software is becoming off-the-shelf through open source such as Google's TensorFlow and Nvidia's CUDA tools. The next breakthrough is likely to be in hardware. Cheaper and faster hardware will pave the way for consumer-level products that fit in a smartphone or drive a car.

Hardware is already trending toward chip sets with massively parallel neural networks built in. The Von Neumann architecture, long criticized for its processor-memory bottleneck, is giving way to processing-in-memory machines where simulated neural network nodes are embedded in memory. Imagine random access memory in small blocks, each encapsulated in a node of a neural network. Such networks will perform big-data analytics, recognition, and recommending without needing the full power of a general-purpose arithmetic logic unit. Who knows what will emerge in the worldwide network of interconnected devices bearing power neural network chips? C


**References**

1. Cybenko, G. Approximation by superpositions of a Sigmoid function. *Math.Control Signals Systems 2*, (1989), 303–314.
2. Hopfield, J. and Tank, D. Neural computation of decisions in optimization problems. *Biological Cybernetics 52* (1985), 141–152.
3. Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks 3*, 2 (1989), 359–366.
4. McCulloch, W.S. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics 5*, 115 (1943); https://doi.org/10.1007/BF02478259
5. Minsky, M. and Papert, S. *Perceptrons*. MIT Press, 1969.



**Ted G. Lewis** (tedglewis@redshift.com) is an author and consultant with more than 30 books on computing and hi-tech business, a retired professor of computer science, most recently at the Naval Postgraduate School, Monterey, CA, Fortune 500 executive, and the co-founder of the Center for Homeland Defense and Security at the Naval Postgraduate School, Monterey, CA.

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


a https://ab.co/2ypCCnP



# Calendar of Events

**December 2–3**
**VRCAI '18: International Conference on Virtual Reality Continuum and Its Applications in Industry,**
**Hachioji, Japan,**
**Sponsored: ACM/SIG,**
**Contact: Koji Mikami,**
**Email: mikami@stf.teu.ac.jp**

**December 4–7**
**CoNEXT '18: The 14th International Conference on Emerging Networking EXperiments and Technologies,**
**Heraklion, Greece,**
**Sponsored: ACM/SIG,**
**Contact: Alberto Dainotti,**
**Email: alberto@caida.org**

**December 10–14**
**Middleware '18: 19th International Middleware Conference,**
**Rennes, France,**
**Sponsored: ACM/SIG,**
**Contact: Guillaume Pierre,**
**Email: guillaume.pierre@irisa.fr**

**December 13–14**
**CVMP '18: European Conference on Visual Media Production,**
**London, U.K.,**
**Sponsored: ACM/SIG,**
**Contact: Abhijeet Ghosh,**
**Email: abhijeetg@gmail.com**

## 2019

### January

**January 14–18**
**AFIRM '19: ACM SIGIR/SIGKDD African Workshop on Machine Learning for Data Mining and Search,**
**Cape Town, South Africa**
**Co-Sponsored: ACM/SIG,**
**Contact: Hussein Suleman,**
**Email: hussein@cs.uct.ac.za**

**January 29–31**
**FAT* '19: Conference on Fairness, Accountability, and Transparency,**
**Atlanta, GA,**
**Sponsored: ACM/SIG,**
**Contact: danah boyd,**
**Email: danah@datasociety.net**

DOI:10.1145/3286870

George V. Neville-Neil

Article development led by acmqueue
queue.acm.org

# Kode Vicious
# A Chance Gardener

*Harvesting open source products and planting the next crop.*

**Dear KV,**

I am working at a startup where we use a lot of open source code software, not just for our operating systems, but also at the core of several products. We have been building our systems on top of open source for several years, but at this point we only consume the software, we never have time to contribute patches. Working with 30 to 40 different projects, given our small staff, would introduce a lot of engineering overhead that the company simply cannot absorb at the moment.

It also seems to me and the rest of what passes for management at our company that open source is like a massive garden of weeds. New projects pop up all the time, and it is impossible to know if these are really harmful or helpful to our overall systems, so we have to try them out or risk being left out of some new type of system. The other day one of the engineers complained he felt like a gardener whose only tool was a machete, which is not a precision tool.

Larger companies clearly know how to work with open source projects, but how can a startup or even a medium-size company, which lacks the resources to look at all this stuff, cope in the open source world. What is the best way to interact with all these projects?

**A Chance Gardener**

**Dear Chance,**

You have hit upon an excellent metaphor for open source software: a garden. I have to admit, I might liken it more to kindergarten, but let's proceed with your original metaphor.

Many people who have not worked directly with open source assume it is a single thing, or a single idea, when, in fact, it is a term that is applied in as many different ways as there are open source projects and communities. Open source truly is like a garden, one with many different species of plants, some of which are beneficial and give nourishment and others of which are poison.

Separating the wheat from the chaff in such a large and diverse ecosystem is a nontrivial undertaking, but it is one that KV has addressed in several previous columns, including the letter to "Acquisitive."[1] Deciding to use a piece of software always comes down to the quality of the software in question, whether the software is closed, open, or somewhere in between. I find your question is more intriguing from the standpoint of how one interacts with open source projects.

You mentioned that your company consumes open source, and, in fact, this is what most people and companies do—consume—and this is the first stage of working with open source. When you are consuming open source, the most important thing to remember is not to sever the plant from the roots. You should be consuming the software directly from the source, even if you are not following every single change to the upstream source tree.

The worst thing you can do is copy the source tree once and then ignore upstream development for a period of time. Letting your local tree get even

IMAGE BY ENEEMS

a few months out of date on a fast-moving project means you are missing a large number of changes and bug fixes. Often the bug fixes are also security fixes, and we all know what happens when people build products without proper integration of security fixes. Nothing. That's right, pretty much everyone gets a pass because we all know software breaks, and there is currently no liability for building insecure products.

Another way to sever the roots between your system and the open source projects from which you consume code is to make your own changes in the master branch of the tree. Mixing your changes directly into the tree, instead of on a development branch, is a great way to make any update to the software nearly impossible. A great way to ensure you have not severed your software from the root of the tree is to have your own internal CI (continuous integration) system. Many open source projects have their own CI systems, which you can directly integrate into your own development systems, and they can verify whether you have broken the system or if the upstream software itself is broken.

Continuing to stretch the metaphor, perhaps close to the point of breaking, we can think now about the next stage of tending the open source garden. If you had a vegetable garden, but you never tended it, you would get few, if any, vegetables from the garden and it would wither and die. Open source projects are no different in this respect from vegetables: We must tend the garden if we expect it to remain productive; otherwise, we are just being destructive.

There are many ways to tend a garden. Perhaps the first thing that comes to mind is weeding, which we might think of as debugging and patching. Contributing patches back to an open source project is a way to help it improve and grow strong. Most open source projects have a defined process whereby code contributions can be made. Although you mention the overhead involved in having your developers contribute to a project, you should turn this thinking on its head and realize that what they are doing when they submit patches to the upstream project is reducing your company's technical debt. If you keep a patch private, then it must be reintegrated every time you consume a new version of the open source code. After a while, these patches can number in the tens of thousands, or more, lines of code, which is a huge amount of technical debt for you to maintain.

After some period of having your developers email patches and submit pull requests, you will realize that what you want on some projects are your own gardeners. Having members of your team working directly on the open source projects that are most important to the company is a great way to make sure that your company has a front-row seat in how this software is developed.

It is actually a very natural progression for a company to go from being a pure consumer of open source to interacting with the project via patch submission and then becoming a direct contributor. No one would expect a company to be a direct contributor to all the open source projects it consumes, as most companies consume far more software than they would ever produce, which is the bounty of the open source garden. It ought to be the goal of every company consuming open source to contribute something back, however, so that its garden continues to bear fruit, instead of rotting vegetables.

**KV**

**Related articles on queue.acm.org**

**Forced Exception-Handling**
*Kode Vicious*
https://queue.acm.org/detail.cfm?id=3064643

**Outsourcing Responsibility**
*Kode Vicious*
https://queue.acm.org/detail.cfm?id=2639483

**Using Free and Open-Source Tools to Manage Software Quality**
*Phelim Dowling and Kevin McGrath*
https://queue.acm.org/detail.cfm?id=2767182

**Reference**

1. Neville-Neil, G. Lazarus code. *Commun. ACM 58*, 6 (June 2015), 32–33; https://dl.acm.org/citation.cfm?id=2753172

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

DOI:10.1145/3197382

Oren Etzioni

# Point/Counterpoint Point: Should AI Technology Be Regulated? Yes, and Here's How.

*Considering the difficult technical and sociological issues affecting the regulation of artificial intelligence research and applications.*

Government regulation is necessary to prevent harm. But regulation is also a blunt and slow-moving instrument that is easily subject to political interference and distortion. When applied to fast-moving fields like AI, misplaced regulations have the potential to stifle innovation and derail the enormous potential benefits that AI can bring in vehicle safety, improved productivity, and much more. We certainly do not want rules hastily cobbled as a knee-jerk response to a popular outcry against AI stoked by alarmists such as Elon Musk (who has urged U.S. governors to regulate AI "before it's too late").

To address this conundrum, I propose a middle way: that we avoid regulating AI research, but move to regulate AI applications in arenas such as transportation, medicine, politics, and entertainment. This approach not only balances the benefits of research with the potential harms of AI systems, it is also more practical. It hits the happy medium between not enough and too much regulation.

IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

## Regulation Is a Tricky Thing

AI research is now being conducted globally, by every country and every leading technology company. Russian President Vladimir Putin has said "Artificial intelligence is the future, not only for Russia, but for all humankind. It comes with colossal opportunities, but also threats that are difficult to predict. Whoever becomes the leader in this sphere will become the ruler of the world." The AI train has left the station; AI research will continue unabated and the U.S. must keep up with other nations or suffer economically and security-wise as a result.

A problem with regulating AI is that it is difficult to define what AI is. AI used to be chess-playing machines; now it is integrated into our social media, our cars, our medical

devices, and more. The technology is progressing so fast, and gets integrated into our lives so quickly, that the line between dumb and smart machines is inevitably fuzzy.

Even the concept of "harm" is difficult to put into an algorithm. Self-driving cars have the potential to sharply reduce highway accidents, but AI will also cause some accidents, and it's easier to fear the AI-generated accidents than the human-generated ones. "Don't stab people" seems pretty clear. But what about giving children vaccinations? That's stabbing people. Or let's say I ask my intelligent agent to reduce my hard disk utilization by 20%. Without common sense, the AI might delete one's not-yet-backed-up Ph.D. thesis. The Murphy's Law of AI is that when you give it a goal, it will do it, whether or not you like the implications of it achieving its goal (see the Sorcerer's Apprentice). AI has little common sense when it comes to defining vague concepts such as "harm," as co-author Daniel Weld and I first discussed in 1994.[a]

But given that regulation is difficult, yet entirely necessary, what are the broad precepts we should use to thread the needle between too much, and not enough, regulation? I suggest five broad guidelines for regulating AI applications.[b] Existing regulatory bodies, such as the Federal Trade Commission, the SEC, Homeland Security, and others, can use these guidelines to focus their efforts to ensure AI, in application, will not harm humans.

### Five Guidelines for Regulating AI Applications

The first place to start is to set up regulations against AI-enabled weaponry and cyberweapons. Here is where I agree with Musk: In a letter to the United Nations, Musk and other technology leaders said, "Once developed, [autonomous weapons] will permit armed conflict to be fought at a scale greater than ever, and at timescales faster than humans can comprehend. These can be weapons of terror, weapons that despots and terrorists use against innocent populations, and weapons hacked to behave in undesirable ways." So as a start, we should not create AI-enabled killing machines. The first regulatory principle is: "Don't weaponize AI."

**A problem with regulating AI is that it is difficult to define what AI is.**

Now that the worst case is handled, let's look at how to regulate the more benign uses of AI.

The next guideline is an AI is subject to the full gamut of laws that apply to its human operator. You can't claim, like a kid to his teacher, the dog ate my homework. Saying "the AI did it" has to mean that you, as the owner, operator, or builder of the AI, did it. You are the responsible party that must ensure your AI does not hurt anyone, and if it does, you bear the fault. There will be times when it is the owner of the AI at fault, and at times, the manufacturer, but there is a well-developed body of existing law to handle these cases.

The third is that an AI shall clearly disclose that it is not human. This means Twitter chat bots, poker bots, and others must identify themselves as machines, not people. This is particularly important now that we have seen the ability of political bots to comment on news articles and generate propaganda and political discord.[c]

The fourth precept is that AI shall not retain or disclose confidential information without explicit prior approval from the source. This is a privacy necessity, which will protect us from others misusing the data collected from our smart devices, including Amazon Echo, Google Home, and smart TVs. Even seemingly innocuous house-cleaning robots create maps that could potentially be sold. This suggestion is a fairly radical departure from the current state of U.S. data policy, and would require some kind of new legislation to enact, but the privacy issues will only grow, and a more stringent privacy policy will become necessary to protect people and their information from bad actors.

And the fifth and last general rule of AI application regulation is that AI must not increase any bias that already exists in our systems. Today, AI uses data to predict the future. If the data says, (in a hypothetical example), that white people default on loans at rates of 60%, compared with only 20% of people of color, that race information is important to the algorithm. Unfortunately, predictive algorithms generalize to make predictions, which strengthens the patterns. AI is using the data to protect the underwriters, but in effect, it is institutionalizing bias into the underwriting process, and is introducing a morally reprehensible result. There are mathematical methods to ensure algorithms do not introduce extra bias; regulations must ensure those methods are used.

A related issue here is that AI, in all its forms (robotic, autonomous systems, embedded algorithms), must be accountable, interpretable, and transparent so that people can understand the decisions machines make. Predictive algorithms can be used by states to calculate future risk posed by inmates and have been used in sentencing decisions in court trials. AI and algorithms are used in decisions about who has access to public services and who undergoes extra scrutiny by law enforcement. All of these applications pose thorny questions about human rights, systemic bias, and perpetuating inequities.

This brings up one of the thorniest issues in AI regulation: It is not just a technological issue, with a technological fix, but a sociological issue that requires ethicists and others to bring their expertise to bear.

AI, particularly deep learning and machine reading, is really about big data. And data will always bear the marks of its history. When Google is training its algorithm to identify something, it looks to human history, held in those data sets. So if we are going to try to use that data to train a system, to make recommendations or to make autonomous decisions, we need to be deeply aware of how that history has worked and if we as a society want that outcome to continue. That's much big-

a Weld, D. and Etzioni, O. The First Law of Robotics (A Call to Arms), *Proceedings of AAAI*, 1994.

b I introduced three of these guidelines in a *New York Times* op-ed in September 2017; https://nyti.ms/2exsUJc

c See T. Walsh, Turing's Red Flag. *Commun. ACM 59*, 7 (July 2016), 34–37.

**We must recognize that regulations have a purpose: to protect humans and society from harm.**

ger than a purely technical question.

These five areas—no killing, responsibility, transparency, privacy, and bias—outline the general issues that AI, left unchecked, will cause us no end of harm. So it's up to us to check it.

### The Practical Application of Regulations

So how would regulations on AI technologies work? Just like all the other regulations and laws we have in place today to protect us from exploding air bags in cars, *E. coli* in our meat, and sexual predators in our workplaces. Instead of creating a new, single AI regulatory body, which would probably be unworkable, regulations should be embedded into existing regulatory infrastructure. Regulatory bodies will enact ordinances, or legislators will enact laws to protect us from the negative impacts of AI in applications.

Let's look at this in action. Let's say I have a driverless car, which gets in an accident. If it's my car, I am considered immediately responsible. There may be technological defects that caused the accident, in which case the manufacturer starts to share responsibility, for whatever percentage of the defect the manufacturer is responsible. So driverless cars will be subject to the same laws as people, overseen by Federal Motor Vehicle Safety Standards and motor vehicle driving laws.

Some might ask: But what about the trolley problem; How do we program the car to make a choice between hitting several people or just killing the driver? That's not an engineering problem, but a philosophical thought experiment. In reality, driverless cars will reduce the numbers of people hurt or killed in accidents; the edge cases where someone gets hurt because of a choice made by an algorithm are a small percentage of the cases. Look at Waymo, Google's autonomous driving division. It has logged over two million miles on U.S. streets and has only been at fault in one accident, making its cars by far the lowest at-fault rate of any driver class on the road—approximately 10 times lower than people aged 60–69 and 40 times lower than new drivers.

Now, there are probably AI applications that will be introduced in the future, that may cause harm, yet no existing regulatory body is in place. It's up to us as a culture to identify those applications as early as possible, and identify the regulatory agency to take that on. Part of that will require us to shift the frame through which we look at regulations, from onerous bureaucracy, to well-being protectors. We must recognize that regulations have a purpose: to protect humans and society from harm. One place to start having these conversations is through such organizations as the Partnership on AI, where Microsoft, Apple, and other leading AI research organizations, such as the Allen Institute for Artificial Intelligence, are collaborating to formulate best practices on AI technologies and serve as an open platform for discussion and engagement about AI and its influences on people and society. The AI Now Institute at New York University and the Berkman-Klein Center at Harvard University are also working on developing ethical guidelines for AI.

The difficulty of regulating AI does not absolve us from our responsibility to control AI applications. Not to do so would be, well, unintelligent. C

**Oren Etzioni** (orene@allenai.org) is Chief Executive Officer of the Allen Institute for Artificial Intelligence, Seattle, WA, USA, and Professor of Computer Science at the University of Washington.



Watch the author discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/point-counterpoint-on-ai-regulation

DOI:10.1145/3241035

Andrea O'Sullivan and Adam Thierer

# Counterpoint: Regulators Should Allow the Greatest Space for AI Innovation

*Permissionless innovation should be the governing policy for AI technologies.*

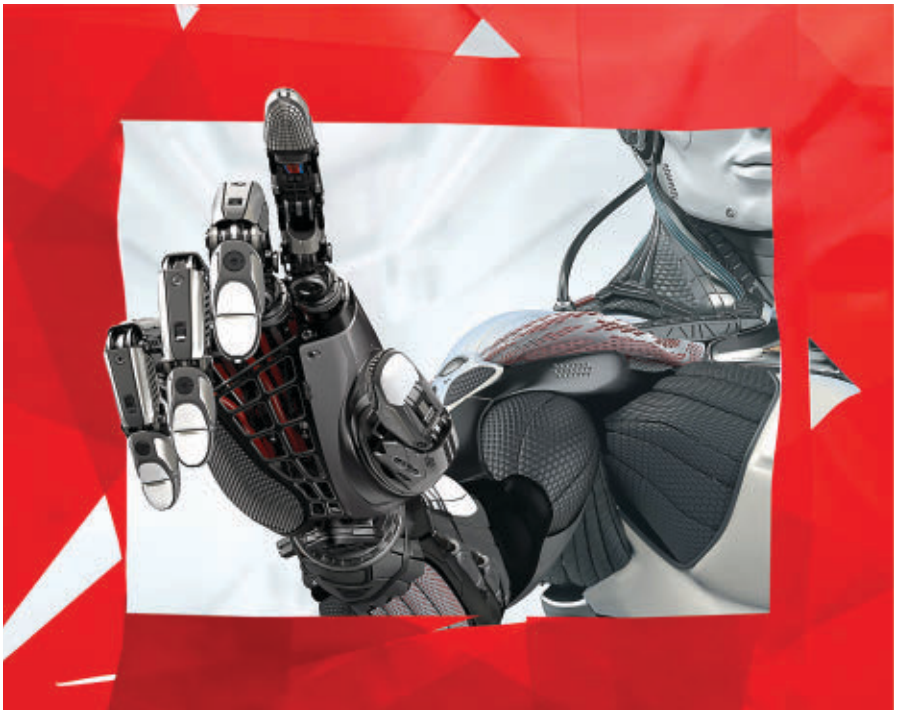


IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

EVERYONE WANTS TO be safe. But paradoxically, sometimes the policies we implement to guarantee our safety end up making us much worse off than if we had done nothing at all. It is counterintuitive, but this is the well-established calculus of the world of risk analysis.

When we consider the future of AI and the public policies that will shape its evolution, it is vital to keep that insight in mind. While AI-enabled technologies can pose some risks that should be taken seriously, it is important that public policy not freeze the development of life-enriching innovations in this space based on speculative fears of an uncertain future.

When considering policy for AI and related emerging technologies such as robotics and big data, policymakers face two general options regarding how best to respond to new technological developments: They can either choose to preemptively set limits or bans on new technologies if they believe the risks to society are simply too great to tolerate—an approach known as the "precautionary principle"—or they can decide to allow innovation to proceed mostly unhampered and intervene only in a post hoc or restitutionary manner, which we call "permissionless innovation."

We believe artificial intelligence technologies should largely be governed by a policy regime of permissionless innovation so that humanity can best extract all of the opportunities and benefits they promise. A precautionary approach could, alternatively, rob us of these life-saving benefits and leave us all much worse off.

## The Risk of Avoiding All Risks

Human psychology is such that the precautionary principle often initially seems appealing. We, as a species, are risk averse. People can quite easily conjure a parade of hypothetical horrible situations they believe new technologies will usher into society. Yet imagined best-case scenarios are not as readily apparent to our risk-adverse psyches.

This can ironically render us less safe. "If the highest aim of a captain were to preserve his ship, he would keep it in port forever," Saint Thomas Aquinas once wrote. Of course, captains aim

higher and take risks in braving the high seas precisely because progress and prosperity—for both them and society at large—depend upon it.

The same holds true for all new innovations. When we fail to consider the upsides new developments could bring, we can end up doing more harm to ourselves than if we allowed a new technique to continue without burden.

Consider drug regulation. On its face, it seems logical for a pharmaceutical regulator like the Food and Drug Administration (FDA) to maintain exacting regulations against new medicines until they can be proven almost entirely safe. After all, the risk of dangerous side effects harming or even killing people in the long term is a formidable one indeed.

But what about the people who could be saved by an experimental treatment that is unjustly delayed or prohibited? We don't see these people or their plights as readily, but they are just as real. Because their suffering or death is under the radar, the tragic effects of this kind of error are not accounted for. This was the unfortunate outcome of the FDA's delayed approval of a drug named Misoprostol to treat gastric ulcers in the early 1980s. Their dithering ended up costing up to 15,000 lives.[a]

Humans are already well aware of the first-order risks of new technologies like AI applications. We fear errors in the opposite direction: that policymakers and the public will underrate the improvements AI can bring, and will allow fears of worst-case scenarios to justify policies that ensure best-case scenarios never come about.

## What's at Stake

After centuries of speculation in both the academy and science fiction, AI is finally shaping our lives in important ways. While we are still far away from the kind of "strong," self-directing AI first anticipated by Mary Shelley's *Frankenstein* almost 200 years ago, narrower applications of machine learning and big data techniques are already integrated into the world around us in subtle but important ways.

Many are unaware of just how prevalent AI techniques already are.[b] They quietly help to more efficiently connect us with the information that is most valuable to us, whether the data in question is related to healthcare, consumer products and services, or just reconnecting with an old friend.

For example, neural networks can help doctors to diagnose medical ailments[c] and recommend treatments, thereby saving money on testing and office visits and potentially improving the likelihood of recovery and remission. Yelp[d] has developed a machine learning program that translates user-submitted restaurant photos into searchable data on a restaurant's cuisine and atmosphere. And the rise of AI-powered "virtual personal assistants"[e] on social media platforms will help us to better keep track of our relationships and obligations with little thought required on our parts.

And yet these marginal improvements in efficiency and matching will yield great dividends in our economy and our personal lives. Analysts[f] project savings and economic growth to exceed hundreds of billions or even trillions of dollars over the coming decade, thanks to improvements in manufacturing, transit, and health. The ease and convenience of tailored artificial assistance will likewise improve our overall qualities of life and leave us more time to do the things that really matter to us.

The U.S. in particular has been a leader in AI development, boasting the world's most innovative research facilities in the academy and industry. But that could soon change. Global challengers Russia and China[g] recognize the importance of shaping AI technologies and have poured substantial support and funding into boosting their national industries. If the U.S. falls behind, global innovation arbitrage[h] will kick in and technologists and companies will flock to countries where such creativity is treated more hospitably.

**Part of the reason the U.S. has been so successful with AI deployment is a relatively permissive policy regime.**

How can the U.S. stay ahead? Part of the reason the U.S. has been so successful with AI deployment is a relatively permissive policy regime. The U.S. houses some of the most successful technology companies in the world due to the federal government's explicit embrace of permissionless innovation in the 1990s.[i] Other countries, particularly in Europe,[j] that pursued a more precautionary approach ended up hemorrhaging talent to other more open environments.

To date, there is no central regulatory authority tasked with reviewing and approving each new instance of AI development in the U.S. Rather, regulators at disparate agencies apply existing rules in accordance with their established authorities—so the FDA oversees applications of health-related AI, the Securities and Exchange Commission (SEC) monitors automated trading, and the National Highway Transportation Safety Administration (NHTSA) is tasked with autonomous vehicle oversight. While imperfect, this approach has the benefit of limiting regulations to a narrow domain.

## Case Study: Autonomous Cars

Autonomous transport presents perhaps the most salient example of how AI will fundamentally change our future. Of course, driverless cars and commercial drones also generate some of the greatest anxieties regarding safety and control. As such, they provide a good example of the tensions between onerous regulation and a more permissive policy environment.

Americans in general are a bit worried by the concept of driverless cars. According to an October 2017 Pew

a https://bit.ly/1PokgDI
b https://bit.ly/2fpaUQ7
c https://bit.ly/2RZf7Km
d https://bit.ly/2R4SYZZ
e https://on.ft.com/2pX38QE
f https://on.ft.com/2pX38QE
g https://bit.ly/2eRiynC
h https://bit.ly/2pWKnwy
i https://bit.ly/2QZaXRb
j https://bit.ly/2CRKi66

Research Center, more than half of Americans say they would outright refuse to ride in a driverless car. Why? Many fear they cannot trust the software undergirding such technologies, and believe the cars will be dangerous. Furthermore, respondents to the Pew poll indicated they do not believe driverless cars will have much of a positive impact on road safety, with 30% reporting they believe road deaths would increase, and another 31% saying they would probably remain about the same. Yet our current human-operated system produces the equivalent of a massacre on the roads each year. 2016 saw the highest number of road fatalities in the past decade, with 40,000 needless deaths by human drivers. Put another way, 100 people were killed by a human driver each day. Autonomous vehicles, on the other hand, could reduce traffic fatalities by up to 90%.[k] This means the cost of delaying driverless car technologies due to regulatory anxieties would mean tens of thousands of needless deaths each year. A Mercatus Center model[l] suggests a regulatory delay of 5% could yield an additional 15,500 needless fatalities. A delay of 25% would mean 112,400 needless deaths. The difference between regulatory humility and regulatory dithering could literally be the difference between life and death for many.

### A Better Path Forward: Humility and Restraint

This illustration should not be construed as a call to "do nothing." Rather, it is meant to paint a picture of the real potential cost of bad policy. Rather than rushing to regulate in an attempt to formalize safety into law, we should first pause and consider the risks of avoiding all risks.

In our recent research paper, "Artificial Intelligence and Public Policy,"[m] co-authored with Raymond Russell, we outline a path forward for policymakers to embrace permissionless innovation for AI technologies. In general, we recommend regulators:

- Articulate and defend permissionless innovation as the general policy default.

**For most AI applications, the promised benefits far outweigh the imagined danger.**

- Identify and remove barriers to entry and innovation.
- Protect freedom of speech and expression.
- Retain and expand immunities for intermediaries from liability associated with third-party uses.
- Rely on existing legal solutions and the common law to solve problems.
- Wait for insurance markets and competitive responses to develop.
- Push for industry self-regulation and best practices.
- Promote education and empowerment solutions and be patient as social norms evolve to solve challenges.
- Adopt targeted, limited, legal measures for truly hard problems.
- Evaluate and reevaluate policy decisions to ensure they pass a strict benefit-cost analysis

Of course, these recommendations must be tailored to the kind of application under consideration. Social media and content aggregation services already enjoy liability protection under Section 230 of the Communications Decency Act of 1996, but the question of liability for software developers of autonomous vehicles is still being discussed.

In that regard, we should not forget the important role the courts and common law will play in disciplining bad actors. If algorithms are faulty and create serious errors or "bias," powerful remedies already exist in the form of product defects law, torts, contract law, property law, and class-action lawsuits.

Meanwhile, at the federal level, the Federal Trade Commission already possesses a wide range of consumer protection powers through its broad authority to police "unfair and deceptive practices." Similarly, at the state level, consumer protection offices and state attorneys general also address unfair practices and continue to advance their own privacy and data security policies, some of which are often more stringent than federal law.

So, we can dispense with the idea that AI is not regulated. Regulatory advocates and concerned policymakers might still be able to identify particular AI applications that present true and immediate threats to society (such as "killer robots" or other existential threats) and which require more serious consideration and potential control. Government uses of profiling software for law enforcement falls under this category, due to its capacity to violate established civil liberties.

But we should realize the vast majority of AI applications do not fit into this bucket; for most AI applications, the promised benefits far outweigh the imagined danger, which can so seductively inflame our anxieties and lead to full-blown technopanics.

The more sensible tone and policy disposition for AI was nicely articulated by *The One Hundred Year Study on Artificial Intelligence*,[n] a Stanford University-led project that brought together 17 of the leading experts to compile a comprehensive report on AI issues. "Misunderstanding about what AI is and is not, especially against a background of scare-mongering, could fuel opposition to technologies that could benefit everyone. This would be a tragic mistake," they argued. "Regulation that stifles innovation, or relocates it to other jurisdictions, would be similarly counterproductive."

That is precisely the sort of humility and patience that should guide our public policies toward AI going forward. As our machines get smart, it is vital for us to make our policies even smarter. C

k https://mck.co/2AjEPjh
l https://bit.ly/2QTMUDo
m https://bit.ly/2CqzQBp
n https://stanford.io/2bDmIhf

**Andrea O'Sullivan** (aosullivan@mercatus.gmu.edu) is the former Technology Policy Program Manager at the Mercatus Center at George Mason University, Fairfax, VA, USA.

**Adam Thierer** (athierer@mercatus.gmu.edu) is a Senior Research Fellow at the Mercatus Center at George Mason University, Fairfax, VA, USA.



Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/point-counterpoint-on-ai-regulation

DOI:10.1145/3195180

Ryen W. White

# Viewpoint
# Opportunities and Challenges in Search Interaction

*Seeking to address a wider range of user requests toward task completion.*

IMAGE BY ZYABTCH

Interacting with search systems, such as Web search engines, is the primary means of information access for most people. Search providers have invested billions of dollars developing search technologies, which power search engines and feature in many of today's virtual assistants (including Google Assistant, Amazon Alexa, Microsoft Cortana, and others). For decades, search has offered a plentiful selection of research challenges for computer scientists and the advertising models that fund industry investments are highly lucrative. Given the phenomenal success, search is often considered a "solved problem." There is some truth to this for fact-finding and navigational searches, but the interaction model and the underlying algorithms are still brittle in the face of complex tasks and other challenges, for example, presenting results in nonvisual settings such as smart speakers.[15] As a community, we need to invest in evolving search interaction to, among other things, address a broader range of requests, embrace new technologies, and support the often underserved "last mile" in search interaction: task completion.

## Search Interaction

The retrieval and comprehension of information is important in many settings. Billions of search queries reach search engines daily and searching skills are now even taught in schools. Search interaction has been studied by information science, information retrieval (IR), and human-computer interaction (HCI) researchers. Information scientists have examined the cognitive and behavioral mechanisms in the search process. IR researchers have developed new methods to collect and find information, including, recently, increased use of machine learning. HCI researchers have studied interactions with technology to develop interfaces to support activities such as information finding and sensemaking. Future opportunities are plentiful, including the three areas discussed in this Viewpoint:

- For more than a decade, search interaction has been immersed in a data revolution, using big (population) data[1] and small (personal) data[3] to model search activity and improve search experiences. This has used traditional data sources (queries, clicks), but richer data (browse, cursor, physiology, spatial context, and so forth) is emerging that enables search systems to more fully represent interests and intentions, unlocking sophisticated modeling methods such as deep learning.
- Support for search interaction has focused on helping searchers build queries and select results. Search systems must evolve to support more complex search activities, leveraging technological advances to meet people's growing expectations about search capabilities.
- Virtual assistants offer an alternative means to engage with search systems. Assistants support rudimentary question answering but will soon more fully comprehend question semantics, understand intent through dialog, and support task completion through skill chaining and skill recommendation.

## Data Revolution

There have been three documented "revolutions" in search-related research: cognitive (targeting intellectual processes); relevance (understanding different relevance types and criteria); and interactive (providing search support and capturing searcher preferences).[12] The interactive revolution continues to this day. We are also in the middle of a fourth revolution: the *data* revolution, driven by enhanced capabilities and interest in recording, analyzing, and learning from user data, both in the aggregate and individually. Application of data mining and machine learning models to query-click data has yielded improvements in ranking, query suggestion, and search advertising, as well as better understanding searchers, their activities, and their satisfaction and success.

Going forward, search *tasks* must be regarded as first-class elements in the search process. Session data is still only used to augment individual queries[3] when the focus should be on supporting end-to-end task completion. Web browser trails capture behavioral traces in online environments that can help direct others.[13] Search providers could mine such activity sequences to harness the procedural search knowledge of populations. These could be used in strategic search support, such as guided tours, that span full tasks, not just the starting points offered in today's search results.[4]

Web search engines have made substantial progress in better understanding the intended meaning of a query, for example, recognizing mentions of entities and common query patterns. These are used not only to improve the precision and recall of query results, but also to attempt to provide a direct "best answer" for the most likely query meaning. Beyond query text, many new signals are available to search systems through new interaction modalities such as touch and gesture, as well as sensors tracking signals including physiology, eye gaze, and locomotion. Cursor movements can also be collected at scale and help interpret user activity in the absence of click-through data. Beyond interactions, search engines are also increasingly using semantic data to better understand document content. This data is sourced from background knowledge graphs and from the documents themselves in the form of embedded semantic data using the common schema.org ontology and markup standards (microdata, RDFa and JSON-LD). Cloud services mean data collected from users and elsewhere is no longer siloed in specific machines or applications. Longitudinal data about searchers helps build rich models of their interests and expertise. Search personalization is operationalized using short- and long-term data from individuals,[3] which can be scaled to cohorts if data is sparse. Even non-search services (for example, productivity applications) can offer data to enrich contextual models and improve search effectiveness. For example, a search for "VAR" from a spreadsheet provides evidence the intent is variance, and not value at risk, for example. Other signals such as spatial context and time offer rich information about the search situation.

**The wealth of opportunity should not translate to dramatically increased complexity.**

Using activity data at massive scale offers incredible potential to understand the human condition. Although logs lack ground truth about search intent, success, experience, and attention, they can still help characterize search behavior, build machine-learned models, and make meaningful discoveries, for example, forecasting influenza in populations.[6] Access to this data is restricted to search providers or only available for purchase from analytics companies for a significant fee, hindering scientific progress. To help address this, search providers have released limited search log data and other resources (for example, Microsoft recently released MARCO,[a] a machine reading comprehension dataset), and some researchers broadly share user study data (an encouraging trend). Open data movements such as data.gov promote data availability, but not for search data, at least not yet.

In working with search interaction data (or any user data) to make intelligent inferences, privacy and data reliability are paramount. Privacy concerns stemming from the construction of user profiles and detailed surveillance of people's activities must be addressed. Systems should obtain user consent and offer clear explanations about what is being recorded and how it is being used. Search providers must act responsibly and correct any biases in search results,[7] in data collected from users and in user sampling. Humans are affected by many factors impacting recorded activities[14] (for example, cognitive biases, behavioral biases, common misconceptions, and misinformation and rumor). These factors can skew behavioral signals such as click-through rates used in ranking algorithms, creating "filter bubbles."[11] This must be considered during data collection and experimental analyses.[5]

Many of these lessons apply in domains beyond Web search. Much of search is domain specific, including legal, medical, and intellectual property. Even within Web search, there are different verticals (including images, video, news) each with its own presentation format and interaction method (for example, "infinite scrolling" in image search). Boundaries between vertical and generic search are blurring as content from verticals bleeds into general result pages, affecting search interactions.[10]

## Evolving Capabilities and Expectations

Advances in data availability coupled with new interaction paradigms (such as touch, gaze, large displays, gesture, spoken dialog), mobile computing capabilities (including tablets, smartphones, smartwatches), and the democratization of artificial intelligence, have created new opportunities for information access and use. Searchers can now interact with search systems in more lightweight and natural ways,[7] including while engaging in non-search tasks. Information visualization tools such as Microsoft SandDance[b]

a http://www.msmarco.org/

b https://www.sanddance.ms/

help people explore and understand data, building on prior HCI research on visualization.[2] Machine learning advances yield significant gains in conversational intelligence and question answering. Improvements in near- and far-field speech recognition coupled with new dialog research make conversational search feasible. Even within current interaction paradigms, deeply understanding query and document semantics can help provide more intelligent responses; for example, medical symptom answers on Google and multi-perspective answers on Bing.

Mobile devices such as smartphones and tablets are powerful and versatile. The integration of hardware such as accelerometers, gyroscopes, and proximity sensors provides rich contextual signals about user activities that are useful for search and recommendation. Evidence from self-reports and log analysis suggests people now demand search support in more situations—to resolve a diverse set of questions (or arguments!)—and question complexity continues to rise. Complex tasks spanning devices are also more frequent. Search systems can utilize downtime between task activities to perform "slow searches," for example, finding sets of relevant resources or using crowdworkers to compose answers.

Wearable and augmented reality applications support the presentation of relevant information just in time, in anticipation of its use. Hardware such as hearables (for example, Google Pixel Buds) or head-mounted displays (such as Google Glass, Microsoft HoloLens) provide continuous information access in any setting. For some tasks (for example, monitoring activities), relevant information can be offered proactively, capitalizing on signals such as user preferences and location. Proactive notifications need to be carefully gated and privacy must be respected, including the privacy of any collocated individuals.

The wealth of opportunity should not translate to dramatically increased complexity. The prevalence of the Google interface design has meant searchers expect simplicity, and rightly so: search activities are already sufficiently complex. Any new capabilities must be intuitive, simple, and add clear value.

## Virtual Assistants

Integration with virtual assistants such as Amazon Alexa, Google Assistant, or Microsoft Cortana allows search systems to extend their capabilities to better understand needs and support higher-order search activities such as learning, decision making, and action.[9] Search engines can provide an entry point to virtual assistants when search requests demand additional engagement (for example, are non-navigational). Search technology already powers some virtual assistants, and knowledge bases created for information finding have utility herein. End-to-end task completion (that is, from search interactions to action in the physical world) has traditionally been underserved by search engines. This can be achieved via first- and third-party skills in virtual assistants. Skills best suited to the current context can be recommended by assistants and even chained together to support multistage tasks.

Virtual assistants are particularly amenable to supporting search interaction: they are personal and contextual, they support dialog, and they are ubiquitous (across applications and devices). Deep understanding of searchers and their contexts is necessary to adapt system responses to the situation. Natural interactions, including multi-turn dialogs, enable search systems to clarify searcher needs. Conversational search is already attracting significant interest.[3] Ubiquity has advantages beyond availability, that is, richer data enables sophisticated inferences such as automatically detecting task completion or estimating task duration, as well as supporting rapid task resumption.

Despite its promise, search-assistant integration is not without challenges that require rethinking several aspects of search interaction. For example, although virtual assistants can foster dialog, natural language conversations can be inefficient ways to obtain answers or complete tasks. Virtual assistants often manifest in headless devices such as smart speakers and personal audio, making it difficult to communicate result lists or discover assistant capabilities.[16] Also, the traditional search-advertising model depends on visual attention and does not scale well to audio-only settings.

## Looking Ahead

We are just beginning a journey to a more enlightened society facilitated by interactions with search systems. Looking ahead, the data revolution in search interaction will gather pace, searchers will engage with search systems in new ways, and virtual assistants will serve as comprehensive search companions. Building on these and other pillars, search systems will empower people and support the activities they value. This important effort will only succeed given the expertise, collaboration, and commitment of communities within computer science and beyond. C


**References**

1. Agichtein, E., Brill, E., and Dumais, S. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, 19–26.
2. Ahlberg, C., Williamson, C., and Shneiderman, B. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1992), 619–626.
3. Bennett, P.N. et al. Modeling the impact of short-and long-term behavior on search personalization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, 185–194.
4. Bush, V. As we may think. *The Atlantic Monthly 176*, 1 (Jan. 1945), 101–108.
5. Eckles, D., Karrer, B., and Ugander, J. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference 5*, 1 (Jan. 2017).
6. Ginsberg, J. et al. Detecting influenza epidemics using search engine query data. *Nature 457*, 7232, (2009), 1012–1014.
7. Kay, M., Matuszek, C., and Munson, S.A. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems ACM*, 2015, 3819–3828.
8. Hearst, M.A. (2011). 'Natural' search user interfaces. *Commun. ACM 54*, 11 (Nov. 2011), 60–67.
9. Marchionini, G. Exploratory search: From finding to understanding. *Commun. ACM 49*, 4 (Apr. 2006), 41–46.
10. Metrikov, P. et al. Whole page optimization: How page elements interact with the position auction. In *Proceedings of the 15th ACM Conference on Economics and Computation*. ACM, 583–600, 2014.
11. Pariser, E. *The Filter Bubble: How the New Personalized Web is Changing What We Read and How We Think*. Penguin, New York, NY, 2011.
12. Robertson, S.E. and Hancock-Beaulieu, M.M. On the evaluation of IR systems. *Information Processing and Management 28*, 4 (Apr. 1992), 457–466.
13. White, R.W., Bilenko, M., and Cucerzan, S. Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 159–166, 2007.
14. White, R.W. Beliefs and biases in web search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 3–12, 2013.
15. White, R.W. *Interactions with Search Systems*. Cambridge University Press, New York, NY, 2016.
16. White, R.W. Skill discovery in virtual assistants. *Commun. ACM 61*, 11 (Nov. 2018), 108–115.



**Ryen W. White** (ryenw@microsoft.com) is a Partner Researcher and a Research Manager at Microsoft Research AI, Redmond, WA, USA.


# ANNOUNCING 20 WORKSHOPS PLANNED FOR IPDPS 2019 IN BRAZIL

IPDPS Workshops are the "bookends" to the three-day technical program of contributed papers, keynote speakers, PhD student forum, and industry participation. They provide an opportunity to explore special topics and present work that is more preliminary or cutting-edge than the more mature research presented in the main symposium. Each workshop has its own website and submission requirements, and the submission deadline for most workshops is after the main conference author notification dates. See the IPDPS Workshops page for links to Call for Papers for each workshop and due dates.

## IPDPS WORKSHOPS on MONDAY 20 MAY 2019 (*Check final schedule*)

| | |
|---|---|
| **HCW** | *Heterogeneity in Computing Workshop* |
| **RAW** | *Reconfigurable Architectures Workshop* |
| **HiCOMB** | *High Performance Computational Biology* |
| **GrAPL** | *Graph, Architectures, Programming, and Learning* |
| **EduPar** | *NSF/TCPP Workshop on Parallel and Distributed Computing Education* |
| **HIPS** | *High Level Programming Models and Supportive Environments* |
| **HPBDC** | *High-Performance Big Data and Cloud Computing* |
| **AsHES** | *Accelerators and Hybrid Exascale Systems* |
| **PDCO** | *Parallel and Distributed Combinatorics and Optimization* |
| **APDCM** | *Advances in Parallel and Distributed Computational Models* |

## IPDPS WORKSHOPS on FRIDAY 24 MAY 2019 (*Check final schedule*)

| | |
|---|---|
| **PDSEC** | *Parallel and Distributed Scientific and Engineering Computing* |
| **iWAPT** | *International Workshop on Automatic Performance Tunings* |
| **JSSPP** | *Job Scheduling Strategies for Parallel Processing* |
| **MPP** | *Parallel Programming Model: Special Edition on Edge/Fog/In-Situ Computing* |
| **ROME** | *Runtime, Operating Systems and Middleware towards Exascale* |
| **SNACS** | *Scalable Networks for Advanced Computing Systems Workshop* |
| **PAISE** | *Parallel AI and Systems for the Edge* |
| **WRA** | *Workshop on Resource Arbitration* |
| **BDDMC** | *Big Data Driven Mobile Computing* |
| **ScaDL** | *Scalable Deep Learning over Parallel and Distributed Infrastructure* |

### GENERAL CHAIR
**Vinod Rebello** (Fluminense Federal University, Brazil)

### PROGRAM CHAIR and VICE-CHAIR
**José Moreira** (IBM Research, USA)
**Alba Cristina Melo** (University of Brasilia, Brazil)

### WORKSHOPS CHAIR and VICE-CHAIR
**Erik Saule** (University of North Carolina Charlotte, USA)
**Jaroslaw Zola** (The State University of New York at Buffalo, USA)

### STUDENT PARTICIPATION CHAIRS
**Edson Borin** (University of Campinas, Brazil)
**Jay Lofstead** (Sandia National Laboratories, USA)

### INDUSTRIAL LIAISON CHAIR
**Márcio Castro** (Federal University of Santa Catarina, Brazil)

### PhD FORUM & STUDENT MENTORING
This event will include traditional poster presentations by PhD students enhanced by a program of mentoring and coaching in scientific writing and presentation skills and a special opportunity for students to hear from and interact with senior researchers attending the conference.

### INDUSTRY PARTICIPATION
IPDPS extends a special invitation for companies to become an IPDPS 2019 Industry Partner, especially those operating in Brazil and South America. It will be a unique opportunity to associate with an international community of top researchers and practitioners in fields related to parallel processing and distributed computing. Visit the IPDPS website to see ways to participate.

### IMPORTANT DATES

| | |
|---|---|
| Conference Preliminary Author Notification | *December 8, 2018* |
| Workshops' Call for Papers Deadlines | *Most Fall After December 8, 2018* |

### *IPDPS 2019 VENUE*
*Rio de Janeiro is one of the most visited cities in the Southern Hemisphere, known for its natural settings, plentiful beaches, and dramatic mountains, all to a backdrop of Samba and Bossa Nova rhythms. Its wonderful weather, unique gastronomy, and famous landmarks draw visitors from around the world. Join IPDPS 2019 in Rio at the Hilton Rio de Janeiro Copacabana Hotel to experience first-hand why this is such a special and memorable destination!*

IEEE COMPUTER SOCIETY
Sponsored by IEEE Computer Society
Technical Committee on Parallel Processing

In cooperation with
ACM SIGARCH & SIGHPC and IEEE TCCA & TCDP

DOI:10.1145/3267116

Article development led by acmqueue
queue.acm.org

**Learn from the past to prepare for the next battle.**

BY RICH BENNETT, CRAIG CALLAHAN, STACY JONES, MATT LEVINE, MERRILL MILLER, AND ANDY OZMENT

# How to Live in a Post-Meltdown and -Spectre World

THE WORLD OF vulnerability management is rapidly changing to keep pace with the complexity of potential threats requiring remediation. What will it look like to live in this world for the next 10 to 15 years?

In 1996, Aleph One published "Smashing the Stack for Fun and Profit."[1] For the next decade, stack smashing was a common form of exploitation, and the security community expended significant effort to finding defenses against it. The Spectre and Meltdown vulnerabilities may constitute an equally seminal moment, ushering in a decade or more of chronic risk-management issues. Indeed, two variants were recently released: SpectrePrime and MeltdownPrime, as detailed in a recent paper by Caroline Trippel, Daniel Lustig, and Margaret Martonosi.[3] Expect these to be the first of many.

Spectre and Meltdown create a risk landscape that has more questions than answers. This article addresses how these vulnerabilities were triaged when they were announced and the practical defenses that are available. Ultimately, these vulnerabilities present a unique set of circumstances, but

IMAGE BY ANDRIJ BORYS ASSOCIATES. USING PHOTO BY MONKEY BUSINESS IMAGES

for the vulnerability management program at Goldman Sachs, the response was just another day at the office.

While these vulnerabilities are theoretically fascinating, we have to live with their practical impact. As risk managers at Goldman Sachs, a large enterprise of approximately 35,000 employees, we had to respond rapidly to the announcement of the vulnerabilities. Moreover, we will have to continue managing the risks that will arise over the next decade from new variants or similar vulnerabilities.

We learned about the vulnerabilities when they were publicly announced on January 3, 2018. The announcement was made earlier than planned because word was already starting to leak. This meant that many vendors had not yet released patches or prepared customer communications about impact, mitigation strategies, and the timelines for patch availability. Vendors could not immediately help in understanding the vulnerabilities.

The first challenge when any major vulnerability is released is to gather information: which systems are impacted, when will patches be available, what compensating controls are in place, and is the vulnerability being actively exploited? It's even better to know if the vulnerability is being exploited by threat actors who have historically targeted your firm.

Meltdown and Spectre were particularly difficult to triage. It was clear early on that certain processor families were impacted, but the full scope was suspected to be much wider. Moreover, our hardware and software inventories

focus on operating systems, applications, and the overall computer model. They are not set up for rapidly revealing the brand and model number of the processors.

It would have been simpler to patch all our machines, but we were wary of news that patches might cause significant performance impacts.

Initially, estimates of performance impact from patching ranged wildly on blogs and articles and were not directly cited in official papers. On January 18, 2018, Eric Siron of Altaro.com summarized that sentiment, saying, "We've all seen the estimations that the Meltdown patch might affect performance in the range of 5% to 30%. What we haven't seen is a reliable data set indicating what happens in a real-world environment."[2] Those ranges were borne out in our own testing of patches, with some systems suffering worse slowdowns than others. Moreover, roundtables with other chief information security officers indicated similar ranges.

These patches had a particularly poor risk trade-off: high potential performance impact, imperfect security benefit. Normally, a patch fixes a vulnerability. Because these are fundamental design vulnerabilities—and worse, vulnerabilities in the hardware design—the patch opportunities are limited. Rather than fixing the underlying vulnerability, they essentially put up a labyrinth to stop an adversary from exploiting it, but the underlying vulnerability remains. Moreover, our experience with complex vulnerabilities is that the first patch is often flawed, so we expected that many of the patches would be updated over time—an expectation that has since proven true.

Although patching was clearly going to be problematic, our quick triage highlighted some good news. Exploiting these vulnerabilities required executing code locally on the victim machine. That led to considering which parts of the operating environment are likely to run untrusted code: hypervisors in the public cloud, employee endpoints such as desktops and laptops, mobile devices, and the browsers or applications that often open email attachments. Since patches could have significant performance impacts, every decision would have to involve a risk trade-off.

The conclusion was that desktops were at most risk, and testing showed that the performance impact would be manageable. We thus immediately began to patch all of our desktops. For servers, we decided to investigate further and make more nuanced, risk-based decisions. The risk of cyberattack had to be balanced against the operational risk of the patch breaking or significantly slowing the systems.

There was no information that the vulnerabilities were being actively exploited, which was reassuring. On the other hand, the nature of the vulnerabilities is such that exploitation is difficult to detect. If we know a vulnerability is being exploited, we will try to push a patch even if there is a high risk of the patch breaking some of the systems. With these vulnerabilities, the lack of known exploitation reinforced the decision to take more time assessing our servers.

To aid in this assessment of risk, we examined our patch strategy and compensating controls through the following lenses: public cloud, servers, employee endpoints, browsers, and email. These lenses also helped communicate the risks to our business leadership.

**Public cloud.** Research showed that attacks leveraging Meltdown and Spectre could target a public cloud environment. In certain cases, an attacker could defeat the technology used by the public cloud providers to ensure isolation between customers' instances. If a malicious user were able to bypass the hypervisor or container engine controls, then that user could access other customers' data collocated on the same hardware.

Thus, our most immediate concerns were public cloud providers. The public cloud risk could be further broken into instance-to-instance attacks and within-an-instance attacks.

In an instance-to-instance attack, a customer could attack another customer on the same hypervisor. Meltdown was the most obvious vector for this attack. An attacker could theoretically just pay for an instance on the public cloud and then target any other customer on that hardware. Fortunately, several large providers—including Amazon, Google, and Microsoft—had received advance notice and had completed, or nearly completed, an initial round of patching on their hypervisors to address these concerns. Moreover, some of the providers informed us that they had patched months before the vulnerabilities were publicized without any noticeable performance impact.

For a within-an-instance attack, the attacker would have to run code on the same instance. This would require access to the system or application to exploit the vulnerability. It was not immediately clear what needed to be done to completely protect against the multiple variants that could be used in this attack. The protections implemented by the public cloud providers remediated Meltdown, but the Spectre variants required multiple mitigations. Google published a binary modification technique called Retpoline that it used to patch its systems against Spectre Variant 2. This had the benefit of minimal performance impact compared with CPU patches. Mitigations for other providers included chip firmware, hypervisor patches, operating system patches, and even application rewrites.

Spectre remediation is made even more complicated because customers and cloud providers have to work in tandem, depending on the cloud service in use. Our impact analysis determined that the within-an-instance risk was not significantly increased by running instances in the public cloud: It was essentially the same risk faced with the internal servers. Accordingly, we treated it as we treated all of our servers: by making individual, risk-based decisions.

**Servers.** At Goldman Sachs, server performance is critical, so we must be careful in patching our servers. In financial services, many critical applications are time-sensitive and effective only if the processing is completed rapidly—for example, applications that perform trading or large-scale, complex risk calculations. This patch could have very real-world implications. If the hundreds of thousands of public cloud processors used every night to perform complex risk calculations had their processing speed reduced by 30%, in addition to the operational risks that

could be raised and potential concerns about robust and real-time risk management, our bill could potentially see a significant increase to compensate for the lost computing power.

Server patching then actually becomes a question of understanding the firm's thousands of internal applications and making a risk-based decision on them. For some applications, performance is critical, and the likelihood of running untrusted code is low. In those cases, we—and the other major firms we talked to—decided not to patch. For those applications, we relied upon compensating controls and the fact that they are very unlikely to run untrusted code. For other applications, we assessed the risk to be higher and patched their servers. To do this type of risk-based analysis, a firm has to understand both the behavior (application profiling) and risk (risk-based categorization) of its applications.

**Endpoints.** Employee endpoints, such as desktops and laptops, were also a high priority within our triage process, as they have access to the Internet via the Web and email. These are key channels through which threat actors looking to exploit these vulnerabilities could attempt to deliver malware.

The Goldman Sachs endpoint response had two key themes: patching and controls. Because user endpoints are much more likely to run untrusted code than servers, we decided to patch in all but the most exceptional circumstances. We thus rapidly deployed patches to our managed Windows, macOS, and iOS devices as they became available. Because of concern over potential end-user performance impact, it was hugely beneficial to be able to run repeatable, automated testing on an isolated set of endpoints before pushing the patches across the enterprise.

Patching was not focused exclusively on the operating system. We also considered the availability of patches for components on the employee desktop that could allow for untrusted code execution—for example, applications that open business-related documents. Unfortunately, even months later many of those applications have not been patched by their vendors.

**Meltdown and Spectre were particularly difficult to patch. It was clear early on that certain processor families were impacted, but the full scope was suspected to be much wider.**

Our assessments included the broader control set available on the endpoint—both preventative and detective. We were most interested in determining which layers of defense could play a role in mitigating risk. For prevention, we reviewed the configuration hardening for our builds and application whitelisting capability and concluded that they did not require any changes. We also use both signature-based and heuristic-based malware detection on our endpoints and on incoming email. Of course, the signature-based tools will have value only when exploits become public.

Not only is it important to look at all of the potential options to mitigate the risk, but also to have the foundational blocks in place for controls that can be adapted to mitigate a broad set of threats in a constantly evolving landscape.

**Browsers.** The Web contains plenty of malicious websites that could attempt to exploit these vulnerabilities. Even legitimate websites may inadvertently host malicious advertisements. Or, in the case of a watering-hole attack, an adversary could compromise a website that a company's employee population is known to visit and use it to deliver malicious code.

At Goldman Sachs we use a Web proxy and service to categorize domain names to reduce risk. Our proxy settings are extremely conservative, blocking entire categories of Web pages that are not relevant to our business or are potentially risky. That includes many of the servers used to host advertisements, so we already have a reasonable amount of advertisement blocking. The proxies also block the downloading of executable files.

In addition, Google Chrome and Microsoft Edge have site isolation capabilities that stop malicious code from impacting more than one tab in the browser window. Like patching, this is not a perfect mitigant for these vulnerabilities, but it does provide a partial control and another layer of defense. As this feature was ready even before some patches, it was implemented rapidly. Although we feared that it would break many internal or external sites, there were actually very few problems.

More specific patches for browsers came out from days to weeks after the initial vulnerability disclosure. We pushed those patches out rapidly. A few hundred of our developers use nonstandard browsers to test their applications, so we used application whitelisting on user endpoints to ensure that only managed browsers, or approved and patched exceptions, were being used.

Browser plug-ins can also execute on untrusted code. As a partial mitigant, specific plug-ins can be locked down to a set of whitelisted sites. Very few plug-ins have released patches, so this remains an area of concern.

Some firms have also chosen to virtualize browsers to isolate the application from the operating system. A browser can be virtualized either as a stand-alone application or as the entire desktop operating system. If any mission-critical Web applications are running on legacy browsers or with plug-ins, a virtualized browser can provide a more protected mechanism for doing so.

**Email** is another common vector for untrusted code. It is not a likely tool for exploiting these vulnerabilities as a majority of the attack vectors have included cache-timing attacks, which are difficult or impossible to exploit over email. Nonetheless, it is important to address phishing attacks as a means of general exploitation. Most firms, including Goldman Sachs, use a variety of techniques to block email-based attacks.

The simplest technique is to block certain types of attachments. If your business supports it, this is a relatively cheap control that can have a significant impact. Unfortunately, many businesses depend upon being able to share office documents, such as PDF or Excel files, that can include macros or other types of code.

Of course, phishing emails do not necessarily contain attachments. They can also contain links to malicious websites. We rewrite incoming URLs so that outbound calls have to go through a central control point where we can quickly implement a block. Outbound Web connections also have to go through the same proxy-based controls described earlier.

In addition, we use signature-based email blocking technologies within our layered approach. As long as there are no known exploits, however, there are no known signatures to deploy. This will be an area to track going forward when the exploits move from research proof-of-concept to being weaponized.

There will likely be more value in "combustion chambers," which open attachments in a virtual machine and look for malicious behavior. Some combustion chamber vendors are looking at running unpatched virtual machines and using them to detect the exploitation of these vulnerabilities.

**Hardware fixes.** While patches and controls are the focus here, hardware fixes are not totally out of the question. Intel indicated in its Q4 earnings call that chips with silicon changes (directly addressing Spectre and Meltdown) will begin to hit the market later this year. Similar to the operating system patches, however, the first generation of hardware fixes may not fully address the vulnerabilities. Moreover, it will be years before organizations upgrade all of their hardware with the new chips.

## Just Another Day of Vulnerability Management?

These vulnerabilities pushed the vulnerability management process at Goldman Sachs, but they did not break it. We are used to making risk trade-offs in this space: for example, do you patch more quickly to decrease the risk of cyber-exploitation, even if that increases the risk of an operational breakdown?

The risks posed by these vulnerabilities are a more challenging version of that scenario. The question is not just whether the patches would break a system, but whether they would have a significant performance impact. That risk has to be assessed in a distributed way, as it is unique to each application. At the same time, there is a lot of uncertainty about these vulnerabilities and how readily they could be exploited. We therefore have to balance operational risk with cyber-attack risk when both risks are unclear.

The scope of vulnerabilities such as Meltdown and Spectre is so vast that it can be difficult to address. At best, this is an incredibly complex situation for an organization like Goldman Sachs with dedicated threat, vulnerability management, and infrastructure teams. Navigation for a small or medium-sized business without dedicated triage teams is likely harder. We rely heavily on vendor coordination for clarity on patch dependency and still have to move forward with less-than-perfect answers at times.

Good cyber-hygiene practices remain foundational—the nature of the vulnerability is different, but the framework and approach to managing it are not. In a world of zero days and multidimensional vulnerabilities such as Spectre and Meltdown, the speed and effectiveness of the response to triage and prioritizing risk-reduction efforts are vital to all organizations. More high profile and complex vulnerabilities are sure to follow, so now is a good time to take lessons learned from Spectre and Meltdown and use them to help prepare for the next battle. C

**Related articles on queue.acm.org**

**Securing the Tangled Web**
*Christoph Kern*
https://queue.acm.org/detail.cfm?id=2663760

**One Step Ahead**
*Vlad Gorelik*
https://queue.acm.org/detail.cfm?id=1217266

**Understanding Software Patching**
*Joseph Dadzie*
https://queue.acm.org/detail.cfm?id=1053343

**References**

1. Aleph One. Smashing the stack for fun and profit. *Phrack 49*, 7 (1996); http://phrack.org/issues/49/14.html#article.
2. Siron, E. The actual performance impact of Spectre/Meltdown Hyper-V updates. Hyper-V Blog, 2018; https://www.altaro.com/hyper-v/meltdown-spectre-hyperv-performance/.
3. Trippel, C., Lustig, D., Martonosi, M. Meltdown Prime and Spectre Prime: automatically synthesized attacks exploiting invalidation-based coherence protocols, 2018. arXiv:1802.03802; https://arxiv.org/abs/1802.03802.

**Rich Bennett**, **Craig Callahan**, **Stacy Jones**, **Matt Levine**, **Merrill Miller**, and **Andy Ozment** are on the global technology risk and information security team at Goldman Sachs.

DOI:10.1145/3267122

Article development led by acmqueue
queue.acm.org

**How documentation enables SRE teams to manage new and existing services.**

BY SHYLAJA NUKALA AND VIVEK RAU

# Why SRE Documents Matter

SITE RELIABILITY ENGINEERING (SRE) is a job function, a mind-set, and a set of engineering approaches for making Web products and services run reliably. SREs operate at the intersection of software development and systems engineering to solve operational problems and engineer solutions to design, build, and run large-scale distributed systems scalably, reliably, and efficiently.

SRE core functions include:

- Monitoring and metrics: Establishing desired service behavior, measuring how the service is actually behaving, and correcting discrepancies.
- Emergency response: Noticing and responding effectively to service failures in order to preserve the service's conformance to its SLA (service-level agreement).
- Capacity planning: Projecting future demand and ensuring that a service has enough computing resources in appropriate locations to satisfy that demand.
- Service turn-up and turn-down: Deploying and removing computing resources for a service in a data center in a predictable fashion, often as a consequence of capacity planning.
- Change management: Altering the behavior of a service while preserving service reliability.
- Performance: Design, development, and engineering related to scalability, isolation, latency, throughput, and efficiency.

SREs focus on the life cycle of services—from inception and design, through deployment, operation, refinement, and eventual decommissioning.

Before services go live, SREs support them through activities such as system design consulting, developing software platforms and frameworks and capacity plans, and conducting launch reviews.

Once services are live, SREs support and maintain them by:

- Measuring and monitoring availability, latency, and overall system health.

- Reviewing planned system changes.
- Scaling systems sustainably through mechanisms such as automation.
- Evolving systems by pushing for changes that improve reliability and velocity.
- Conducting incident responses and blameless postmortems.

Once services reach end of life, SREs decommission them in a predictable fashion with clear messaging and documentation.

A mature SRE team likely has well-defined bodies of documentation associated with many SRE functions. If you manage an SRE team or intend to start one, this article will help you understand the types of documents your team needs to write and why each type is needed, allowing you to plan for and prioritize documentation work along with other team projects.

### A SRE's Story

Before discussing the nuances of SRE documentation, let's examine a night and day in the life of Zoë, a new SRE.

Zoë is on her second on-call shift as an SRE for Acme Inc.'s flagship AcmeSale product. She has been through her induction process as a team member, where she watched her colleagues while they were on-call, and she took notes as well as she could. Now she has the pager.

As luck would have it, the pager goes off at 2:30 A.M. The alert says "Ragnarok job flapping," and Zoë has no idea what it means. She flips through her notes and finds the link to the main dashboard page. Everything looks OK. She does a search on the Acme intranet to find any document referencing Ragnarok, and after precious minutes go by, she finds an outdated design document for the service, which turns out to be a critical dependency for AcmeSale.

Luckily, the design document links to a "Ragnarok Ops" page, and that page has links to a dashboard with charts that look like they might be useful. One of the charts displays a traffic dip that looks alarming. The Ops page also references a script called ragtool that can apparently fix problems like the one she is seeing, but this is the first time she has heard of it. At this point, she pages the backup on-call SRE for help because he has years of experience with the service and its management tools. Unfortunately, she gets no response. She checks her email and finds a message from her colleague saying he is offline for an hour because of a health emergency. After a moment of inner debate, she calls her tech lead, but the call goes to voicemail. It looks like she has to tackle this on her own.

After more searching to learn about this mysterious ragtool script, she finds a document with one-line descriptions of its command-line options, which also tells her where to find the script. She runs `ragtool-restart` and crosses her fingers. Nothing changes, and in fact the traffic drops even more. She reads frantically through more command-line options but is not sure whether they will do more harm than good. Finally, she concludes that `ragtool-ebalance e-dc=atlanta` might help, since another chart indicates that the Atlanta data center is having more trouble. Sure enough, the line on the traffic chart starts creeping upward, and she thinks she is out of the woods. MTTR (mean time to repair) is 45 minutes.

The next day Zoë has a postmortem discussion about the incident with her team. They are having this discussion because the incident was a major outage causing loss of revenue, and their manager has been asking them to do more postmortems. She asks the team how they would have handled the situation differently, and she hears three different approaches. There appears to be no standard troubleshooting process. Her colleagues also acknowledge that the "flapping" alert is poorly named, and that the failure was a result of a well-known bug in the product that hasn't been a high priority for the developer team.

Finally, Steve, her tech lead, asks, "Which version of ragtool did you use?" and then points out that the version she used was very old. A new release came out a week ago with brand-new documentation describing all its new features and even explaining how to fix the "Ragnarok job flapping" problem. It might have reduced the MTTR to five minutes.

The existence of the new version of ragtool comes as a surprise to about half the team, while the other half is somehow familiar with the new version and its user guide. The latest script and document are both under Steve's home directory, in the bin/folder, of course. Zoë writes this down in her notes for future reference, hoping devoutly that she will get through this shift without further alerts. She wonders whether her tech lead or anyone else will follow up on the problems uncovered during the postmortem discussion, or whether future SREs are doomed to repeat the same painful on-call experience.

Later that day Zoë attends an SRE onboarding session, where the SRE team meets with a product development team to talk about taking over their service. Steve leads the meeting, asking several pointed questions about operational procedures and current reliability problems with the service, and asking the developer team to make several operational and feature changes before the SRE team can take it over. Zoë has been to a few such meetings already, which are led either by Steve or another senior SRE. She realizes the questions asked and the actions assigned to the developers seem to vary quite a bit, depending on who is leading the meeting and what types of product failures the SRE team has dealt with in the past week.

She wishes vaguely that the team had more consistent standards and procedures but doesn't quite know how to achieve that goal. Later, she hears two of the developers joking near the coffee machine that many of the questions seemed quite unrelated to carrying a pager, and they had no idea where those questions came from. She wishes product development teams could understand that SREs do a lot more than carry pagers. Back at her desk, however, Zoë finds several urgent tickets to resolve, so she never follows up on those thoughts.

Luckily, all the characters and episodes in this story are fictional. Still, consider whether any part of the story resembles any of your real-life experiences. The solution to this fictional team's struggles is entirely obvious, and the next section expands on this solution.

### The Importance of Documentation

In the early stages of an SRE team's existence, the organization depends

heavily on the performance of highly skilled individuals on the team. The team preserves important operational concepts and principles as nuggets of "tribal knowledge" that are passed on verbally to new team members. If these concepts and principles are not codified and documented, they will often need to be relearned—painfully—through trial and error. Sometimes team members perform operational procedures as a strict sequence of steps defined by their predecessors in the distant past, without understanding the reasons these steps were initially prescribed. If this is allowed to continue, processes eventually become fragmented and tend to degenerate as the team scales up to handle new challenges.

SRE teams can prevent this process decay by creating high-quality documentation that lays the foundation for such teams to scale up and take a principled approach to managing new and unfamiliar services. These documents capture tribal knowledge in a form that is easily discoverable, searchable, and maintainable. New team members are trained through a systematic and well-planned induction and education program. These are the hallmarks of a mature SRE team.

The remainder of this article describes the various types of documents SREs create during the life cycle of the services they support.

## Documents for New Service Onboarding

SREs conduct a production readiness review (PRR) to ensure a service meets accepted standards of operational readiness, and that service owners have the guidance they need to take advantage of SRE knowledge about running large systems.

A service must go through this review process prior to its initial launch to production. (During this stage, the service has no SRE support; the product development team supports the service.) The goal of the prelaunch PRR is just to ensure the service meets certain minimum standards of reliability at the time of its launch.

A follow-on PRR can be performed before SRE takeover of a service, which may happen long after the initial launch. For example, when an SRE team decides to onboard a new service, the team conducts a thorough review of the production state and practices of the new service. The goals are to improve the service being onboarded from a reliability and operational sustainability perspective, as well as to provide SREs with preliminary knowledge about the service for its operation.

SREs conducting a PRR before service takeover may ask a more comprehensive set of questions and apply higher standards of reliability and operational ease than when conducting a PRR at the time of the initial launch. They may intentionally keep the launch-time PRR "lighter" than the service take-over PRR in order to avoid unduly slowing down the developer team.

In Zoë's SRE story, her team had no standardized PRR process or checklist, which means they might miss asking important questions during service takeover. Therefore, they run the risk of encountering many problems while managing a new service that were easily foreseeable and could have been addressed before SREs became responsible for running the service.

An SRE PRR/takeover requires the creation of a PRR template and a process doc that describes how SRE teams will engage with a new service, and how SRE teams will use the PRR template. The template used at the time of takeover might be more comprehensive than the one used at the time of initial launch.

A PRR template covers several areas and ensures that critical questions about each area are answered. The accompanying table lists some of the areas and related questions that the template covers.

The process doc should also identify the kinds of documentation that the SRE team should request from the product development team as a prerequisite for takeover. For example, they might ask the developer team to create initial playbook entries for standard problems.

In addition to these onboarding documents, the SRE organization must create overview documents that explain the SRE role and responsibilities in general terms to product development teams. This serves to set their expectations correctly. The first such document would explain what SRE is, covering all the topics listed at the beginning of this article, including core functions, the service life cycle, and support/maintenance responsibilities. A primary goal of this document is to ensure developer teams do not equate SREs with an Ops team or consider pager response to be their sole function. As shown in the earlier SRE story, when developers do not fully understand what SREs do before they hand off a service to SREs, miscommunication and misunderstandings can result.

Additionally, an engagement model document goes a little further in setting expectations by explaining how the SRE team will engage with developer teams during and after service takeover. Topics covered in this doc include:

- Service takeover criteria and the PRR process;

**Example PRR template areas**

| Area | Questions |
|---|---|
| Architecture and dependencies | What is your request flow from user to front end to back end? Are there different types of requests with different latency requirements? |
| Capacity planning | How much traffic and rate of growth do you expect during and after the launch? Have you obtained all the compute resources needed to support your traffic? |
| Failure modes | Do you have any single points of failure in your design? How do you mitigate unavailability of your dependencies? |
| Processes and automation | Are any manual processes required to keep the service running? |
| External dependencies | What third-party code, data, services, or events do the service or the launch depend upon? Do any partners depend on your service? If so, do they need to be notified of your launch? |

- SLO negotiation process and error budgets;
- New launch criteria and launch freeze policy (if applicable);
- Content and frequency of service status reports from the SRE team;
- SRE staffing requirements; and,
- Feature roadmap planning process and priority of reliability features (requested by SREs) versus new product functionality.

### Documents for Running a Service

The core operational documents SRE teams rely on to perform production services include service overviews, playbooks and procedures, postmortems, policies, and SLAs. (Note: this section appeared in the "Do Docs Better" chapter of *Seeking SRE*.[1])

**Service overviews** are critical for SRE understanding of the services they support. SREs need to know the system architecture, components and dependencies, and service contacts and owners. Service overviews are a collaborative effort between the development team and the SRE team and are designed to guide and prioritize SRE engagement and uncover areas for further investigation. These overviews are often an output of the PRR process, and they should be updated as services change (for example, new dependency).

A basic service overview provides SREs with enough information about the service to dig deeper. A complete service overview provides a thorough description of the service and how it interacts with the world around it, as well as links to dashboards, metrics, and related information that SREs need to solve unexpected issues.

**Playbook.** Also called a *runbook*, this quintessential operational doc lets on-call engineers respond to alerts generated by service monitoring. If Zoë's team, for example, had a playbook that explained what the "Ragnarok job flapping" alert meant and told her what to do, the incident could have been resolved in a matter of minutes. Playbooks reduce the time it takes to mitigate an incident, and they provide useful links to consoles and procedures.

Playbooks contain instructions for verification, troubleshooting, and escalation for each alert generated from network-monitoring processes. Playbooks typically match alert names generated from monitoring systems. They contain commands and steps that need to be tested and reviewed for accuracy. They often require updates when new troubleshooting processes become available and when new failure modes are uncovered or dependencies are added.

Playbooks are not exclusive to alerts and can also include *production procedures* for pushing releases, monitoring, and troubleshooting. Other examples of production procedures include service turnup and turndown, service maintenance, and emergency/escalation.

**Postmortem.** SREs work with large-scale, complex, distributed systems, and they also enhance services with new features and the addition of new systems. Therefore, incidents and outages are inevitable given SRE scale and velocity of change. The postmortem is an essential tool for SRE, representing its formalized process of learning from incidents. In the hypothetical SRE story, Zoë's team had no formal postmortem procedure or template and, therefore, no formal process for capturing the learning from an incident and preventing it from recurring, so they are doomed to repeat the same problems.

SRE teams need to create a standardized postmortem document template with sections that capture all the important information about an outage. This template will ideally be structured in a format that can be readily parsed by data-analysis tools that report on outage trends, using postmortems as a data source. Each postmortem derived from this template describes a production outage or paging event, including (at minimum):

- Timeline;
- Description of user impact;
- Root cause; and,
- Action items/lessons learned.

The postmortem is written by a member of the group that experienced the outage, preferably someone who was involved and can take responsibility for the follow-up. A postmortem needs to be written in a blameless manner. It should include the information needed to understand what happened, as well as a list of action items that would significantly reduce the possibility of recurrence, reduce the impact, and/or make recovery more straightforward. (For guidance on writing a postmortem, see the postmortem template described in *Site Reliability Engineering*.[2])

**Policies.** Policy documents mandate specific technical and nontechnical policies for production. Technical policies can apply to areas such as production-change logging, log retention, internal service naming (naming conventions engineers should adopt as they implement services), and use of and access to emergency credentials.

Policies can also apply to process. Escalation policies help engineers classify production issues as emergencies or non-emergencies and provide recommendations on the appropriate action for each category; on-call expectations policies outline the structure of the team and responsibilities of team members.

**Service-level agreement.** An SLA is a formal agreement with a customer on the performance a service commits to provide and what actions will be taken if that obligation is not met. SRE teams document their service(s) SLA for availability and latency, and monitor service performance relative to the SLA.

Documenting and publishing an SLA, and rigorously measuring the end-user experience and comparing it with the SLA, allows SRE teams to innovate more quickly while preserving a good user experience. SREs running services with well-defined SLAs will detect outages faster and therefore resolve them faster. Good SLAs also result in less friction between SRE and software engineer (SWE) teams because those teams can negotiate targets and results objectively, and avoid subjective discussions of risk.

Note that having an external legally enforceable agreement may not be applicable to most SRE teams. In these cases, SRE teams can go with a set of service-level objectives (SLOs). An SLO is a definition of the desired performance of a service for a single metric such as availability or latency.

**Documents for production products.** SRE teams aim to spend 50% of their time on project work, developing software that automates away manual work or improves the reliability of a managed service. Here, we describe documents that are related to the products and tools SREs develop.

These documents are important because they enable users to find out whether a product is right for them to

adopt, how to get started, and how to get support. They also provide a consistent user experience and facilitate product adoption.

An **About page** helps SREs and product development engineers understand what the product or tool is, what it does, and whether they should use it.

A **concepts guide** or glossary defines all the terms unique to the product. Defining terms helps maintain consistency in the docs and UI, API, or CLI (command-line interface) elements.

The goal of a **quickstart guide** is to get engineers up and running with a minimum of delay. It is helpful to new users who want to give the product a try.

**Codelabs.** Engineers can use these tutorials—combining explanation, example code, and code exercises—to get up to speed with the product. Codelabs can also provide in-depth scenarios that walk engineers step by step through a series of key tasks. These tutorials are typically longer than quickstart guides. They can cover more than one product or tool if they interact.

**How-to guide.** This type of document is for users who need to know how to accomplish a specific goal with the product. How-tos help users complete important specific tasks, and they are generally procedure based.

The **FAQ** page answers common questions, covers caveats that users should be aware of, and points users to reference documents and other pages on the site for more information.

The **support** page identifies how engineers can get help when they are stuck on something. It also includes an escalation flow, troubleshooting info, groups links, dashboard and SLO, and on-call information.

**API reference.** This guide provides descriptions of functions, classes, and methods, typically with minimal narrative or reader guidance. This documentation is usually generated from code comments and sometimes written by tech writers.

**Developer guide.** Engineers use this guide to find out how to program to a product's APIs. Such guides are necessary when SREs create products that expose APIs to developers, enabling creation of composite tools that call each other's APIs to accomplish more complex tasks.

**Playbooks contain instructions for verification, troubleshooting, and escalation for each alert generated from network-monitoring processes.**

### Documents for Reporting Service State

Here, we describe the documents that SRE teams produce to communicate the state of the services they support.

**Quarterly service review.** Information about the state of the service comes in two forms: A quarterly report reviewed by the SRE lead and shared with the SRE organization, and a presentation to the product development lead and team.

The goal of a quarterly report (and presentation) is to cover a "State of the Service" review, including details about performance, sustainability, risks, and overall production health.

SRE leads are interested in quarterly reports because they provide visibility into the following:

- *Burden of support (on-call, tickets, postmortems).* SRE leads know that when the burden of support exceeds 50% of the SRE team's resources, they must respond and change the priorities of their teams. The goal is to give early warning if this starts to trend in the wrong direction.
- *Performance of the SLA.* SRE leads typically want to know if the SLA is being missed or if the ecosystem has an unhealthy component that puts the product clients in jeopardy.
- *Risks.* SRE leads want to know what risks the SREs see to being able to deliver against the goals of the products and the business.

Quarterly reports also provide opportunities for the SRE team to:

- Highlight the benefit SRE is providing to the product development team, as well as the work of the SRE team.
- Request prioritization for resolving problems hindering the SRE team (sustainability).
- Request feedback on the SRE team's focus and priorities.
- Highlight broader contributions the team is making.

**Production best practices review.** With this review SRE teams are better able to adopt production best practices and get to a very stable state where they spend little time on operations. SRE teams prepare for these reviews by providing details such as team website and charter, on-call health details, projects vs. interrupts, SLOs, and capacity planning.

The best practices review helps the SRE team calibrate itself against the rest of the SRE organization and

improve across key operational areas such as on-call health, projects vs. interrupts, SLOs, and capacity planning.

### Documents for Running SRE Teams

SRE teams need to have a cohesive set of reliable, discoverable documentation to function effectively as a team.

**Team site.** Creating a team site is important because it provides a focal point for information and documents about the SRE team and its projects. At Google, for example, many SRE teams use g3doc (Google's internal doc platform, where documentation lives in source code alongside associated code), but some teams use a combination of Google Sites and g3doc, with the g3doc pages closely tied to the code/implementation details.

**Team charter.** SRE teams are expected to maintain a published charter that explains the rationale for the team and documents its current major engagements. A charter serves to establish the team identity, primary goals, and role relative to the rest of the organization.

A charter generally includes the following elements:

- A high-level explanation of the space in which the team operates. This includes the types of services the team engages with (and how), related systems, and examples.
- A short description of the top two or three services managed by the team. This section also highlights key technologies used and the challenges to running them, benefits of SRE engagement, and what SRE does.
- Key principles and values for the team.
- Links to the team site and docs.

Teams are also expected to publish a vision statement (an aspirational description of what the team would like to achieve in the long term) and a roadmap spanning multiple quarters.

### Documents for New SRE Onboarding

SRE teams invest in training materials and processes for new SREs because training results in faster onboarding to the production environment. SRE teams also benefit from having new members acquire the skills required to join the ranks of on-call as early as possible. In the absence of comprehensive training, as seen in Zoë's story, the on-call SRE can flounder during a crisis, turning a potentially minor incident into a major outage.

**SRE teams invest in training materials and processes for new SREs because training results in faster onboarding to the production environment.**

Many SRE teams use checklists for on-call training. An on-call checklist generally covers all the high-level areas team members should understand well, with subsections under each area. Examples of high-level areas include production concepts, front-end and back-end stack, automation and tools, and monitoring and logs. The checklist can also include instructions about preparing for on-call and tasks that need to be completed when on call.

SREs also use role-play training drills (referred to within Google as *Wheel of Misfortune*) as an educational tool for training team members. A Wheel of Misfortune exercise presents an outage scenario to the team, with a set of data and signals that the hypothetical on-call SRE will need to use as input to resolve the outage. Team members take turns playing the role of the on-call engineer in order to hone emergency mitigation and system-debugging skills. Wheel of Misfortune exercises should test the ability of individual SREs to know where to find the documentation most relevant to troubleshooting and resolving the outage at hand.

**Repository management.** SRE team information can be scattered across a number of sites, local team knowledge, and Google Drive folders, which can make it difficult to find correct and relevant information. As in the SRE example earlier, a critical operational tool and its user manual were unavailable to Zoë (the on-call SRE) because they were hidden under the home directory of her tech lead, and her inability to find them greatly prolonged a service outage. To eliminate this type of failure, it is important to define a consistent structure for all information and ensure all team members know where to store, find, and maintain information. A consistent structure will help team members find information quickly. New team members can ramp up more quickly, and on-call and on-duty engineers can resolve issues faster.

Here are some guidelines to create and manage a team documentation repository:

- Determine relevant stakeholders and conduct brief interviews to identify all needs;

▸ Locate as much documentation as possible and do a gap analysis on content;

▸ Set up a basic structure for the site so that new documentation can be created in the correct location;

▸ Port relevant existing documentation to a new location;

▸ Create a monitoring and reporting structure to track the progress of migration;

▸ Archive and tear down old documentation;

▸ Perform periodic checks to verify that consistency/quality is being maintained;

▸ Verify that commonly used search terms bring up the right documents near the top of the search results; and,

▸ Use signals such as Google Analytics to gauge usage.

A note on repository maintenance: it is important that docs are reviewed and updated on a regular basis. The owner's name should be visible, as well as the last reviewed date—all this information helps with the trustworthiness of the documentation. In Zoë's story she found and used an obsolete document for a critical operational tool and thereby missed an opportunity to resolve an incident quickly rather than experience a major outage. If documents cannot be trusted to be accurate and current, this can make SREs less effective, directly impacting the reliability of the services they manage.

**Repository availability.** SRE teams must ensure documentation is available even during an outage that makes the standard repository unavailable. At Google, SREs have personal copies of critical documentation. This copy is available on an encrypted compact storage device or similar detachable but secure physical media that all on-call SREs carry with them.

## Documents for Service Decommissioning

Once services reach end of life, SREs decommission them in a predictable fashion. Here, we provide messaging and documentation guidelines for service deprecation leading to eventual decommissioning.

It is important to announce decommissioning to current service users well ahead of time and provide them with a timeline and sequence of steps. Your announcement should explain when new users will no longer be accepted, how existing and newly found bugs will be handled, and when the service will completely stop functioning. Be clear about important dates and when you will be reducing SRE support for the service, and send interim announcements as the timeline progresses.

Sending an email message is not sufficient, and you must also update your main documentation pages, playbooks, and codelabs. Also, annotate header files if applicable. Capture the details of the announcement in a document (in addition to email), so it's easy to point users to the document. Keep the email as short as possible, while capturing the essential points. Provide additional details in the document, such as the business motivations for decommissioning the service, which tools your users can take advantage of when migrating to the replacement service, and what assistance is available during migration. You should also create a FAQ page for the project, growing the page over time as you field new questions from your users.

**Technical writers** provide a variety of services that make SREs effective and productive. These services extend well beyond writing individual documents based on requirements received from SRE teams.

Here is some guidance to technical writers on best practices for working with SRE teams.

▸ Technical writers should partner with SREs to provide operational documentation for running services and product documentation for SRE products and features.

▸ They can create and update doc repositories, restructure and reorganize repositories to align with user needs, and improve individual docs as part of the overall repository management effort.

▸ Writers should provide consulting to assess, assist, and address documentation and information management needs. This involves conducting doc assessments to gather requirements, enhancing docs and sites created by engineers, and advising teams on matters related to documentation creation, organization, redesign, findability, and maintenance.

▸ Writers should evaluate and improve documentation tools to provide the best solutions for SRE.

**Templates.** Tech writers also provide templates to make SRE documentation easier to create and use. Templates do the following:

▸ Make it easy for authors to create documentation by providing a clear structure so that engineers can populate it quickly with relevant information;

▸ Ensure documentation is complete by including sections for all required pieces of documentation; and,

▸ Make it easy for readers to understand the topic of the doc quickly, the type of information it's likely to contain, and how it's organized.

*Site Reliability Engineering* contains several examples of documentation templates. To view the templates, visit https://queue.acm.org/appendices/SRE_Templates.html

## Conclusion

Whether you are an SRE, a manager of SREs, or a technical writer, you now understand the critical importance of documentation for a well-functioning SRE team. Good documentation enables SRE teams to scale up and take a principled approach to managing new and existing services.

**Related articles on queue.acm.org**

**The Calculus of Service Availability**
*Ben Treynor, Mike Dahlin, Vivek Rau, and Betsy Beyer*
https://queue.acm.org/detail.cfm?id=3096459

**Resilience Engineering: Learning to Embrace Failure**
*A discussion with Jesse Robbins, Kripa Krishnan, John Allspaw, and Tom Limoncelli*
https://queue.acm.org/detail.cfm?id=2371297

**Reliable Cron across the Planet**
*Štepán Davidovic and Kavita Guliani*
https://queue.acm.org/detail.cfm?id=2745840

**References**

1. Blank-Edelman, D.N. *Seeking SRE: Conversations About Running Production Systems at Scale.* O'Reilly Media, 2018.
2. Murphy, N., Beyer, B., Jones, C., Petoff, J. *Site Reliability Engineering: How Google Runs Production Systems.* O'Reilly Media, 2016

**Shylaja Nukala** is a technical writing lead for Google Site Reliability Engineering. She leads the documentation, information management, and select training efforts for SRE, Cloud, and Google engineers.

**Vivek Rau** is a Site Reliability Engineer at Google, working on Customer Reliability Engineering (CRE). The CRE team teaches customers core SRE principles, enabling them to build and operate highly reliable products on the Google Cloud Platform.

DOI:10.1145/3267114

Article development led by acmqueue
queue.acm.org

**Five strategies for pushing through.**

BY KATE MATSUDAIRA

# How to Get Things Done When You Don't Feel Like It

HAVE YOU EVER come into work, sat down at your computer to begin a project, opened your editor, and then just stared at the screen? This happens to me all the time, so I understand your struggle.

Even if you love your job, you don't always feel like doing it every day. There are so many factors that influence your ability to show up to work with enthusiasm and then work hard all day long.

External events can take priority in your mind—family struggles, a breakup, a sick pet—and make it difficult to focus. Then, of course, there are the struggles at work that can make it hard to feel motivated. Getting a bad review can knock you off course. Likewise, if you work really hard on a project and your manager doesn't seem to value it at all, you might wonder why you are working so hard.

Other times you have to work on tasks you don't enjoy (for me, that is writing lots of tests, or documentation) or projects that aren't challenging. If your work is uninteresting or if a task you have been assigned seems beneath your ability, finding your motivation can be challenging.

So, what do you do? Many people turn to procrastination or ignoring the task—but that only postpones the inevitable. You can try to talk your way out of the assignment, and maybe your manager will support you, but at some point the work must get done.

If you want to be successful, then it serves you better to rise to the occasion no matter what. That means learning how to push through challenges and deliver valuable results.

Since this happens to me quite often, I have captured five of my best strategies for turning out amazing work even when I don't feel like it.

## Gamify Your Process

Dealing with a really big project used to hold me back. If the project had lots of tasks I didn't know how to do or that seemed really difficult, I resisted even starting because I was so overwhelmed by the scope.

Of course, this meant I procrastinated until only the minimum amount of time remained to complete the project. Then I would end up working crazy long hours, and sometimes I ended up with code that "worked" but was in no way ready for prime time (for example, a few bugs, not enough coverage of edge cases, minimal testing, working only in my dev environment because I could not make it work on staging, and so on.). This was super stressful and usually meant my work was not as good as it could have been if I had only started earlier.

This was one of the biggest obstacles early in my career: I had a tough time getting started.

I discovered that if I made the process of getting started easier, those

IMAGE BY LIGHTSPRING

first few steps on a daunting project became more tenable. Once I took a few steps, it was so much easier to keep going.

My solution was to approach a project by turning it into as many tiny steps as possible. That way I could get a few really easy wins under my belt. For example, each step would be a task such as "Search for _____ on Google" or "Have a conversation with _____." Crossing things off your to-do list gives your brain a happy little dopamine hit, even if the tasks are tiny—it keeps your motivation up and your excuses down.

Try breaking your next project into the smallest increments you possibly can. Each step should be really small (I try for tasks that take 15 minutes or less) and really easy to accomplish, so that you can get a win!

You have to overcome inertia. Little wins add up and make it easier to do that.

### Reserve Calendar Time for Every Project

Set aside time on your calendar specifically for working on a task you are having trouble starting. Treat it as seriously as you would any other appointment. You must show up and you must work on that project.

Reserve an amount of time that is realistic for making progress—at least 30 minutes to an hour. This strategy is key for busy people or managers. If you don't schedule the time to do meaningful strategic work, your time will fill up with tactical tasks.

And what if you don't feel like working on the task at the appointed time? Set a timer when you are starting work. Set it for 10 minutes and tell yourself you have to work only until the timer goes off.

Start working on the list of tiny steps you have created for yourself: Google something; set up your project; send one email message; review one document.

Almost always, taking one or two of these tiny steps will get your brain working, and it will be easier to keep going. You'll do one task, cross it off the list, and then do another. Your timer for 10 minutes will go off, and you'll just keep going because now you're engaged with the project.

If you are really not engaged with it after 10 minutes (though this rarely happens to me), then let yourself take a break—but block off another chunk of time on your calendar to come back to it soon.

### Get Other People Involved

Sometimes the best way to get yourself to do something is to make yourself accountable to another person.

According to a study by the Ameri-

can Society of Training and Development,[1] people who commit to someone else have a 65% chance of accomplishing the goals they set. That number goes up to 95% if you commit to a specific accountability appointment with that person.

Our brains are wired not to want to let down other people. If someone invests in you by agreeing to help you accomplish your goal, you are driven to do your part by a desire to live up to that commitment.

There are a few ways to do this:

- Set deadlines with your manager for when certain aspects of the project will be complete, and schedule regular check-ins on status.
- Ask for help on a part of the project. With the help of another person to reduce your workload, you can get other parts of the project done. Set a time to meet with your helper to combine your results.
- Make a recurring date with a peer to work together. For example, if you are both tasked with running a series of tedious tests that you both would rather put off, set a time to sit together and get them done.
- Embrace the scrum part of Agile and have daily standups with your teammates.

Delegating work can be especially helpful when you have a really big project in front of you. Sometimes the scope of a project is so overwhelming that it is hard to get started; if you can solicit help from your team to tackle some of the project, then you can focus your efforts on a more manageable workload.

## Talk About It

Externalizing problems can make them a lot easier to deal with. Things tend to get blown out of proportion in our minds, especially when we are stressed about them.

I can't tell you how many times I have started talking to someone about how stressed I am about a project—like I don't have any ideas for an article, or it's so difficult I have no clue how I will solve it—that by the time I am done talking, I actually come away full of inspiration. Other times, I am just so stressed about what could go wrong (or what is going wrong) that I fast become overwhelmed.

Scientific studies have shown that talking about feelings out loud actually decreases stress and the bad feelings we are experiencing. Brain imaging done at UCLA[2] demonstrated that when a person was shown a picture of an angry face, the amygdala became more active. This is the part of the brain responsible for activating the body's "alarm" system—it lets you know that you have something to fear and kicks your body into action to deal with that threat.

When the study participants were able to name what they saw, however, the simple act of putting the feeling they saw into words caused the amygdala activity to decrease. Not only that, but each participant's right ventrolateral prefrontal cortex then became activated. Other studies have indicated that this is the area of the brain associated with processing emotion and putting words to emotional experiences.

So, talking about your big project might be just the thing to help you get started.

Plus, if you talk to smart friends or mentors, they might have suggestions for how you can start or experiences to share about how they did something similar. You can become more relaxed and smarter at the same time.

## Practice the Art of "Precrastination"

Do you ever have trouble working from home because you get distracted by unwashed dishes in the sink or laundry that needs to be folded? You have probably been told that you are a procrastinator, but, in fact, you might be just the opposite.

I used to be a master procrastinator. I would find any excuse to keep from starting work, or even thinking about it. As I learned again and again, procrastination is a bad thing. It comes from a fear of getting started, so you actively keep yourself from making progress by doing things that keep your mind off of what you must do.

But there is something called "precrastination," and it is actually really good for you.

As you are working on a project, your brain needs to take breaks—not just to recharge, but also to form new connections and create new ideas. That is why getting up to wash the dishes, fold the laundry, take a shower, take a walk, or any other low-key activity that allows you to let your mind wander for a while can be really good for your productivity overall.

When you do something that feels satisfying, your brain releases dopamine (just as it does when you cross an item off a to-do list—because it feels good!). So, when you take a walk midway through your work session, your brain gets a hit of dopamine.

That dopamine triggers the parts of your brain associated with creativity and gets them working. That's when those magical aha! moments happen, because your brain is sending energy to the areas that help you make connections and see things in new ways.

Next time you are stuck on a project you don't want to start, try doing something that you know will be satisfying. You just might have a bright idea while you are rinsing off your dishes, and that will make you excited to run over to your computer and get to work. C


**Related articles on queue.acm.org**

**People and Process**
*James Champy*
https://queue.acm.org/detail.cfm?id=1122687

**Fresh Starts**
*Kate Matsudaira*
https://queue.acm.org/detail.cfm?id=2996549

**IM, Not IP (Information Pollution)**
*Jakob Nielsen*
https://queue.acm.org/detail.cfm?id=966731



**References**

1. Oppong, T. The accountability effect: A simple way to achieve your goals and boost your performance. The Mission (Jan. 16, 2017); https://medium.com/the-mission/the-accountability-effect-a-simple-way-to-achieve-your-goals-and-boost-your-performance-8a07c76ef53a.
2. Wolpert, S. Putting feelings into words produces therapeutic effects in the brain; UCLA neuroimaging study supports ancient Buddhist teachings. UCLA Newsroom (June 21, 2007); http://newsroom.ucla.edu/releases/Putting-Feelings-Into-Words-Produces-8047.



**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.

# AWARD NOMINATIONS SOLICITED

## As part of its mission, ACM brings broad recognition to outstanding technical and professional achievements in computing and information technology.

ACM welcomes nominations for those who deserve recognition for their accomplishments. Please refer to the ACM Awards website at **https://awards.acm.org** for guidelines on how to nominate, lists of the members of the 2018 Award Committees, and listings of past award recipients and their citations.

Nominations are due **January 15, 2019** with the exceptions of the Doctoral Dissertation Award (due **October 31, 2018**) and the ACM – IEEE CS George Michael Memorial HPC Fellowship (due **May 1, 2019**).

**A.M. Turing Award:** ACM's most prestigious award recognizes contributions of a technical nature which are of lasting and major technical importance to the computing community. The award is accompanied by a prize of $1,000,000 with financial support provided by Google.

**ACM Prize in Computing (previously known as 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. The award carries a prize of $250,000. Financial support is provided by Infosys Ltd.

**Distinguished Service Award:** recognizes outstanding service contributions to the computing community as a whole.

**Doctoral Dissertation Award:** presented annually to the author(s) of the best doctoral dissertation(s) in computer science and engineering, and is accompanied by a prize of $20,000. The Honorable Mention Award is accompanied by a prize totaling $10,000. Winning dissertations are published in the ACM Digital Library and the ACM Books Series.

**ACM – IEEE CS George Michael Memorial HPC Fellowships:** honors exceptional PhD students throughout the world whose research focus is on high-performance computing applications, networking, storage, or large-scale data analysis using the most powerful computers that are currently available. The Fellowships includes a $5,000 honorarium.

**Grace Murray Hopper Award:** presented to the outstanding young computer professional of the year, selected on the basis of a single recent major technical or service contribution. The candidate must have been 35 years of age or less at the time the qualifying contribution was made. A prize of $35,000 accompanies the award. Financial support is provided by Microsoft.

**Paris Kanellakis Theory and Practice Award:** honors specific theoretical accomplishments that have had a significant and demonstrable effect on the practice of computing. This award is accompanied by a prize of $10,000 and is endowed by contributions from the Kanellakis family, and financial support by ACM's SIGACT, SIGDA, SIGMOD, SIGPLAN, and the ACM SIG Project Fund, and individual contributions.

**Karl V. Karlstrom Outstanding Educator Award:** presented to an outstanding educator who is appointed to a recognized educational baccalaureate institution, recognized for advancing new teaching methodologies, effecting new curriculum development or expansion in computer science and engineering, or making a significant contribution to ACM's educational mission. The Karlstrom Award is accompanied by a prize of $10,000. Financial support is provided by Pearson Education.

**Eugene L. Lawler Award for Humanitarian Contributions within Computer Science and Informatics:** recognizes an individual or a group who have made a significant contribution through the use of computing technology; the award is intentionally defined broadly. This biennial, endowed award is accompanied by a prize of $5,000, and alternates with the ACM Policy Award.

**ACM – AAAI Allen Newell Award:** presented to individuals selected for career contributions that have breadth within computer science, or that bridge computer science and other disciplines. The $10,000 prize is provided by ACM and AAAI, and by individual contributions.

**Outstanding Contribution to ACM Award:** recognizes outstanding service contributions to the Association. Candidates are selected based on the value and degree of service overall.

**ACM Policy Award:** recognizes an individual or small group that had a significant positive impact on the formation or execution of public policy affecting computing or the computing community. The biennial award is accompanied by a $10,000 prize. The next award will be the 2019 award.

**Software System Award:** presented to an institution or individuals recognized for developing a software system that has had a lasting influence, reflected in contributions to concepts, in commercial acceptance, or both. A prize of $35,000 accompanies the award with financial support provided by IBM.

**ACM Athena Lecturer Award:** celebrates women researchers who have made fundamental contributions to computer science. The award includes a $25,000 honorarium.

For SIG-specific Awards, please visit **https://awards.acm.org/sig-awards**.

**Vinton G. Cerf**, ACM Awards Committee Co-Chair

**John R. White**, ACM Awards Committee Co-Chair

**Insup Lee**, SIG Governing Board Awards Committee Liaison

**Rosemary McGuinness**, ACM Awards Committee Liaison

DOI:10.1145/3186276

**Citizen-led initiatives via social media yield political influence, including even with a country's top political leaders.**

BY JUNYEONG LEE AND JAYLYN JEONGHYUN OH

# What Motivates a Citizen to Take the Initiative in e-Participation? The Case of a South Korean Parliamentary Hearing

ONLY FOUR YEARS after the greatest number of voters in Korean history elected Park Geun-hye as the country's first female president in 2013, more than one million people gathered for a candle-lit protest in Seoul to also make her the first publicly ousted president of South Korea (Korea hereafter). Amidst these two forms of civic engagement—vote and protest—is a new form of political communication that gained limited attention but was also a surprise to the Korean public. That is, a particular citizen watching a live broadcast of the second hearing in the parliamentary investigation into a political scandal involving president Park and Choi Soon-sil, her former confidante, on December 7, 2016, alerted a member of the country's National Assembly to alleged perjury by Kim Ki-choon, a key political figure in the Choi Soon-sil scandal, via instant messenger KakaoTalk that immediately altered the probe. This message had a stunning effect on the political process and proved to be a landmark not only in this case but in the transformation of e-participation into a popular form of political communication driven by information and communications technology (ICT) worldwide.

The emergence of e-participation had toppled the traditional invisible wall between ordinary citizens and the National Assembly. The Internet extended the exchange of information and thought among citizens, planners, and decision makers alike,[7] but individual citizens' voices only rarely reach the Assembly. The tip-off message granted ordinary citizens influence comparable to that of elected politicians in the country's political culture. More important, public reporting of Kim Ki-choon's alleged perjury was led not by the government but by ordinary citizens through social media. But such citizen-led e-participation, unlike government-led e-participation, has re-

## » key insights

- **Information and communication technology, including social media, live-streaming services, and digital archives, supports citizen-led e-participation.**
- **Collective intelligence and prior experience in online environments help individual citizens gain influence in parliamentary maneuvering and political decision making.**
- **Politicians can nurture e-participation among citizens who are motivated by public debate and thus enabled by communication technologies.**

PHOTO BY EUGENE PIKALOV/SHUTTERSTOCK.COM

Citizens demonstrating in Seoul, South Korea, April 7, 2018.

ceived only limited scholarly attention despite the recent surge in the study of social media. The focus of most e-participation initiatives worldwide excludes their role as a way for citizens to engage in political decision making, defined by Alarabiat et al.[1] as "truly [sic] participation" and as a method for linear communication.[1,11] For example, despite the considerable promise of e-participation initiatives worldwide, most are limited to information delivery and communication, as explored by Alarabiat et al.[1] and Dini and Sæbø.[8] Focusing on the roles of citizens and social media in e-participation,[24,26] a series of research projects by Porwol et al.[21] reviewed the related literature tlooking to define a model integrating social participation and other forms of communication to capture the level of engagement,[23] an e-participation evaluation model,[17] and an ICT exploitation framework for e-participation,[20] finding that existing models generally ignore "emerging phenomenon of spontaneous, citizen-led e-participation, particularly when hosted on a social-media platform."[21] While citizen-led e-participation is one side of the duality of e-participation,[21] citizen-led e-participation via social media is still in its infancy, even in fields related to e-participation, including citizen coproduction and collective action.[16,22] There is thus no clear-cut definition of citizen-led e-participation recognized by most scholars. Building on the definition of e-participation by its earlier researchers,[13,17,20,22,26] focusing on government-led initiatives as "... enhanced civic engagement through ICT enabling citizens to connect with elective representatives," here we focus on citizen-led initiatives involving voluntary and spontaneous participation through social media from a citizen's perspective.

Shedding light on the fact that citizen participation in Korea was not deliberate but rather spontaneous via social media, our research aims to delineate the factors enabling citizen-led e-participation in terms of the Korean National Assembly and the Korean public. It was and remains a significant example of "public partnership," the most active means for citizen participation following Cogan's and Sharpe's "public participation conundrum."[7] In our case, citizens were invited to help shape the ultimate decisions without taking a formal role, except for being citizens, as in the definition of public partnership. Public partnerships via effective use of technology thus suggests a new form of civil participation transcending all extant forms. Our analysis identifies the facilitating factors of citizen-led e-participation, hinting at a new form of civil participation led by individual citizens and promoting citizen participation proactively.

## Methodology

In order to identify these enablers in the case of the parliamentary investigation we witnessed in Korea, we applied qualitative content analysis similar to Ardichivili et al.[2] and Smith.[25] We collected the posts and comments (documented data) from a bulletin board of an online discussion community known as the "stock gallery (bulletin board)" at http://www.dcinside.com/, on December 7, 2016, the second day of parliamentary hearings in the investigation. As one of the leading online communities in Korea, stock gallery had the dominant role in calling out Kim Ki-choon's alleged perjury and cover up. We retrieved 1,794 posts using as a filter the keyword "Kim Ki-choon" and related comments, and collected supplementary posts, including online articles, summarized posts, and posts of parliamentarians and their staff, in regard to Kim's alleged perjury.

We iteratively employed the "constant comparative method," categorizing the posts indicating the factors leading to citizen participation. We conducted the coding independently. First, each of us coded the posts and identified the patterns in the data by reviewing the phrases used in the texts, collapsing and condensing certain phrases. We then categorized the phrases and patterns to describe the enabling factors experienced by citizens we compared and discussed. We then combined them with recontextualization of the data, looking to improve the method's accuracy. Finally, we repeated the process, recoding and reanalyzing some of the data and categories.

## Findings

Based on qualitative content analysis of the data we collected, we identified five major categories of enablers of citizen-led e-participation: live-streaming service and digital archives; direct communication channels between citizens and politicians; collective intelligence; prior experience with similar methods; and public interest, as we explore here:

*Live-streaming service and digital archives.* Following development of ICT, citizens today are able to access various types of content on the Internet, including live parliamentary hearings in our case. Since many types of content are recorded and saved online, citizens can find them whenever they want. In our case, they found them in the form of YouTube videos and online news articles. They then reported them with uniform resource locators to a member of the National Assembly. Members of the National Assembly and their aides confirmed Kim Ki-choon's misconduct in the form of perjury. Citizens were able to learn about it through a live-streaming service. One citizen said, "His [the perjurer's] biggest mistake was that he did not know that the audience of the live-aired hearing nowadays can retrieve the data of evidence, make it public, carry it to parliamentarians, and make it appear on the screen in the court in just a few minutes." Another said, "This is a world where you can see what happens with a few clicks."

*Direct communication channels between citizens and politicians.* Citizens being able to report directly to members of the National Assembly was possible because the private cellphone numbers of the National Assembly members had become available to the public. One notable citizen made a list of personal cellphone numbers of Assembly members, using numbers that had been available on their webpages, blogs, and social media accounts. A number of citizens then began to send their comments directly to those personal numbers via instant messenger (KakaoTalk) and text message (SMS). SMS was even

more revealing, as it displayed whether a message had been delivered to and read by its intended recipient(s). Moreover, citizens could provide information to politicians and receive feedback in real time. Being able to reach members of the National Assembly in real time, personally and directly, citizens could overcome the shortcomings of traditional indirect democracy. Various citizens said, "It is a direct complaint," "It is crazy that online representative democracy is just realized," and "This is a revolution of direct democracy." One member of the Assembly, Park Young-sun, communicated with citizens, not just with her political aides, thus making the communication direct.

*Collective intelligence.* Although an individual citizen's participation may seem solitary, the wisdom of collective intelligence was now being exercised behind the scenes. On the informant's first post revealing evidence of Kim's alleged perjury, he asked for ways to alert the members of the Assembly about it, resulting in 98 comments of support and collaboration. In the comments, citizens shared not only the personal cellphone numbers of Assembly members but also effective ways to ensure the evidence would be included in subsequent hearings. To attract even more citizen participation, they joined him by "liking" the post, helping push it to the top of the board and leaving the comment: "Make this post to the top." Having observed the whole process behind an historic alert by a citizen, another citizen said, "Collective intelligence made it happen." Online communities enable this mechanism and process of collective intelligence. Yu[27] wrote, "It is a combination of IT information network and collective intelligence of [the online community's] netizens."

*Prior experience in similar practice.* Some citizens were already aware of the powerful effect collective intelligence can have. Looking to get down to the as-yet-unspoken truth of a hot political issue, they are sometimes described collectively as "netizens investigation teams." Unlike our focused context, where a single netizen plays a key role in a real political moment and where the discussion is confined to a single bulletin board in an online community, netizens investigation teams exercise their influence within and across online communities. They also played a role in calling out Choi Soon-sil's corruption. The value of such investigative experience is recognized for achieving future goals and developing self-efficacy.[3] Since self-efficacy is a crucial ingredient in many user behaviors with ICT, it helps encourage netizens to persevere toward their goal of spreading the word. In another article, by Her,[10] their part in helping expose the scandal was described like this: "In October [2016], they [netizens investigation teams] uncovered a post by Choi Soon-sil's daughter, Jung Yoo-La, on her personal webpage, saying, 'If you do not have the ability, you blame your parents. Money is also an ability.'"

**Public disclosure of Kim Ki-choon's alleged perjury was led not by the government but by ordinary citizens through social media.**

*Public interest.* Interest is a crucial factor in citizen-led e-participation. The example of Korean citizens communicating directly with members of the National Assembly concerning perjury by the President's chief of staff was also related to a political corruption scandal involving President Park Geun-hye and other important political and business figures. It attracted significant public interest, not only because of related headlines in the country's major newspapers but also because it had become a topic of everyday conversation in online communities and private group-chats via SMS, as well as in face-to-face everyday neighbor-to-neighbor conversation. It was almost necessary to be aware of the case just to be able to engage in conversation with one's neighbors. The implication was and still is that social interaction, a strong theme behind millennial engagement, helped keep the public's attention focused on the issue. The public's desire for justice and displays of patriotism was significant. The public's antipathy toward the president's behavior thus accumulated until she was finally impeached, with citizens asking, "Is this a nation?" and wanting action on behalf of the country. Kim Gwi-Ok, a professor of sociology at Hansung University, explained it like this: "Citizens who are tired of politics and prosecutors have begun

to press on with diverse actions," and "Netizens investigation teams chasing the problem and collecting data are the subjects that make civil revolution." Other citizens said: "I feel like I did good for my country," "You are a patriot," "Our gallery [community] protects our country," "Justice wins," and "The truth will be known."

### Implications

Exploring the factors that motivated otherwise ordinary citizens to take the initiative in e-participation in the case of this Korean political intrigue, we have identified several topics lacking from the scholarly literature by focusing instead on social media and individual citizen initiative. The implications reflect how social media and other ICT contribute to "true participation" in the digital age.[1,21,26]

Live-streaming services and digital archives allowing millions of Koreans to seek the truth and check political facts produced critical public opinion when their content was known to online communities. Various models of public-opinion formation, including in Katz and Lazarsfeld,[12] have verified that individuals are influenced more through interaction with one another than with mass media. Citizen-users of online communities constantly reinforce the impact of the political content they find in live-streaming services and digital archives by sharing them with other users who then post comments, helping generate a critical mass of public opinion. A government's live-streaming service and official activities of major politicians, through constant activity and updating over decades,[5] can thus stimulate a critical public response in online communities or social media. Such digital information is more meaningful in terms of political participation when it causes the exchange of opinions than when it flows in only one direction, traditionally from government to the public.

Disclosure of the personal cellphone numbers of National Assembly members facilitated what had not previously been considered a means of political participation, enabling both individual and collective opinions in the form of what is often called "texting movement" or "texting democracy" in Korea. While SMS is often viewed as a conveyor of votes, texting democracy expands what it can convey, from votes alone to words, allegations, and opinions. In this sense, SMS in direct citizen-to-politician communication is not atypical but routine. Citizens have been engaged in political processes and events, including through ministers' communication via SMS. Using SMS motivates citizens to participate, even though they do not generally participate in other areas.[9] Moreover, SMS communication between citizens and politicians signals that technological development and its application in political participation have moved and could move further from what has historically been the closed sphere of politicians into the public sphere, as reflected in this citizen comment: "[National Assembly member] Kim Sung Tae's number is public goods." Our findings are in line with the political science literature advocating the virtues of mobile telephony in a healthy democracy when used for citizen-politician communication.[14]

**Although an individual citizen's participation can seem solitary, the wisdom of collective intelligence was now being exercised behind the scenes.**

Collective intelligence can be achieved in a digital-media environment characterized by openness, fluidity, and dynamic interaction to produce a new relationship, including in online communities.[15] In our example of citizen-politician communication, the civic participation using collective intelligence also engendered "common knowledge," or knowledge everyone knows, in terms of both opinion and fact.[6] As the original citizen's political tip-off alert in 2016 was covered extensively, millions of ordinary Korean citizens were able to see how their fellow citizens, as well as they themselves, are able to exert political influence through e-participation and collective intelligence. The now-available common knowledge is expected to influence the Korean public's way of thinking and behavior[6] by granting them greater confidence in their opinions, as well as helping politicians develop the personal attitudes needed to accept them. Common knowledge and a shared nationwide knowledge space can explain the increased participation of citizens using collective intelligence. Such changes consolidate new notions of political partici-

pation and promise to reform the nature of politics in Korea. Scholars and practitioners alike take note.

Along with the confidence netizens gain through their engagement in politically significant events and processes, we also identified disappointment over the lack of systematized channels for the public to report its personal investigations into political intrigue. Comments like "Would it be impossible to have the hearings in a format like YouTube Live where the politicians simultaneously read our comments and proceed? I am sure we [the users in the community] are way better at investigating" and reflect the public need to adjust the system so it facilitates direct civil participation. In this regard, a rigorous look at the needs of Korean citizens will be of both theoretical and practical value. Castells[4] suggested social and political networks in virtual and physical space bring political change. Establishing a systematized channel to link the people and political networks together will help millions of them create political change, as in some cases they already have, despite the deficiency of such channels.

## Conclusion

The true nature of citizens' political interest remains a chronic conundrum for political scientists. In our case, public interest in a political scandal was an underlying factor in Korean nationwide civic participation, emphasizing the importance of public interest in the political arena. While the subjects in our case, citizen-users and elected public officials, likely already had strong interest in political events before the country's political intrigue was so publicly revealed, it also shows how such interest can spread to the broader community of citizens. Users of other online communities learned about the case through shared posts, eventually leaving more than 3,300 replies about the initial netizen-informant's initiative, including, "I really wanted to praise it so I came up here," "I saw it in a different community! Cool!," and "I am not a member of this community, but I have heard about it. Thank you [users of the focal community]. So amazing!" Expanding a previous finding by Masip et al.[18] that reading shared posts by friends leads to more interest and trust than reading posts directly on the original website, people can be expected to be more affected by reading the posts shared by other members of the same community. Accessing shared posts across online communities and social media, more citizens are more likely to become politically active.

We identified five enablers of citizen-led e-participation by analyzing a hearing in the National Assembly, December 7, 2016, of the Parliamentary Inspection committee concerning the Park Geun-hye scandal. In a broader context, the enablers adapted existing communication technology to serve as the technological foundation for citizen-led e-participation. Technological advancement, including social media, may spur e-participation but is less likely to be citizen-led, thus yielding an unbalanced system. This is why further research on e-participation is crucial, stressing its effect on citizens and social media.[1,8,16,21,22,26] Such coproduction between citizens and government "increase the citizens' sense of well-being as a result of greater participation."[19] Only through targeted study can the citizens of Korea and of other countries achieve meaningful civic participation. C


**References**

1. Alarabiat, A., Soares, D.S., and Estevez, E. Electronic participation with a special reference to social media: A literature review. In *Proceedings of the Eighth International Conference on Electronic Participation* (Guimarães, Portugal, Sept. 5–8). Springer, Cham, Switzerland. 2016, 41–52.
2. Ardichvili, A., Page, V., and Wentling, T. Motivation and barriers to participation in virtual knowledge-sharing communities of practice. *Journal of Knowledge Management 7*, 1 (Feb. 2003), 64–77.
3. Bandura, A. *Self-Efficacy: The Exercise of Control.* Freeman, New York, 1997.
4. Castells, M. *Networks of Outrage and Hope: Social Movements in the Internet Age.* Polity Press, Cambridge, U.K., 2012.
5. Chadwick, A. Britain's first live televised party leaders' debate: From the news cycle to the political information cycle. *Parliamentary Affairs 64*, 1 (Jan. 2011), 24–44.
6. Chwe, M.S.-Y. *Rational Ritual: Culture, Coordination, and Common Knowledge.* Princeton University Press, Princeton, NJ, 2013.
7. Cogan, A. and Sharpe, S. The theory of citizen involvement. In *Planning Analysis: The Theory of Citizen Participation.* University of Oregon, Eugene, OR, 1986; http://pages.uoregon.edu/rgp/PPPM613/class10theory.htm
8. Dini, A.A. and Sæbø, Ø. The current state of social media research for e-participation in developing countries: A literature review. In *Proceedings of the 49th Hawaii International Conference on System Sciences* (Koloa, HI, Jan. 5–8). IEEE Press, New York, 2016, 2698–2707.
9. Hellström, J. and Karefelt, A. Participation through mobile phones: A s tudy of SMS use during the Ugandan general elections 2011. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development* (Atlanta, GA, Mar. 12–15). ACM Press, New York, 2012, 249–258.
10. Her, K. The netizen's decisive tip-off has knocked down Kim Ki-choon, *Hankook Ilbo*, Dec, 8. 2016; http://www.hankookilbo.com/News/Read/201612082038581427
11. Hoffman, L.H. Participation or communication? An explication of political activity in the Internet age. *Journal of Information Technology & Politics 9*, 3 (Sept. 2012), 217–233.
12. Katz, E. and Lazarsfeld, P.F. *Personal Influence.* Free Press, New York, 1955.
13. Komito, L. e-Participation and governance: Widening the net. *The Electronic Journal of e-Government 3*, 1 (July 2005), 39–48.
14. Lee, H., Kwak, N., Campbell, S.W., and Ling, R. Mobile communication and political participation in South Korea: Examining the intersections between informational and relational uses. *Computers in Human Behavior 38* (Sept. 2014), 85–92.
15. Lévy, P. *Cyberculture.* Editions Odile Jacob, Paris, France, 1997.
16. Linders, D. From e-government to we-government: Defining a typology for citizen coproduction in the age of social media. *Government Information Quarterly 29*, 4 (Oct. 2012), 446–454.
17. Macintosh, A. and Whyte, A. Towards an evaluation framework for e-participation. *Transforming Government: People, Process and Policy 2*, 1 (Mar. 2008), 16–30.
18. Masip, P., Suau-Martínez, J., and Ruiz-Caballero, C. Questioning the selective exposure to news: Understanding the impact of social networks on political news consumption. *American Behavioral Scientist 62*, 3 (Mar. 2018), 300–319.
19. Mattson, G.A. The promise of citizen coproduction: Some persistent issues. *Public Productivity Review 10*, 2 (Winter 1986), 51–56.
20. Phang, C.W. and Kankanhalli, A. A framework of ICT exploitation for e-participation initiatives. *Commun. ACM 51*, 12 (Dec. 2008), 128–132.
21. Porwol, L., Ojo, A., and Breslin, J.G. An ontology for next generation e-participation initiatives. *Government Information Quarterly 33*, 3 (July 2016), 583–594.
22. Porwol, L. and Ojo, A. Barriers and desired affordances of social media-based e-participation: Politicians' perspectives. In *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance* (New Delhi, India, Mar. 7–9). ACM Press, New York, 2017, 78–86.
23. Preece, J. and Shneiderman, B. The reader-to-leader framework: Motivating technology-mediated social participation. *AIS Transactions on Human-Computer Interaction 1*, 1 (Mar. 2009), 13–32.
24. Sæbø, Ø., Rose, J., and Flak, L.S. The shape of e-participation: Characterizing an emerging research area. *Government Information Quarterly 25*, 3 (July 2008), 400–428.
25. Smith, B.G. Socially distributing public relations: Twitter, Haiti, and interactivity in social media. *Public Relations Review 36*, 4 (Nov. 2010), 329–335.
26. Susha, I. and Grönlund, Å. e-Participation research: Systematizing the field. *Government Information Quarterly 29*, 3 (July 2012), 373–382.
27. Yu, G. 'Stock Galley user did it' ... Main contributor to overturn testimony of Kim Ki-choon by finding perjury video,. *Joongang Ilbo* (Dec. 8, 2016); https://news.joins.com/article/20976733



**Junyeong Lee** (junyeonglee@ustc.edu.cn), the corresponding author, is an assistant professor in the School of Management of the University of Science and Technology of China, Hefei, Anhui, China.

**Jaylyn Jeonghyun Oh** (jaylynoh@purdue.edu) is a Ph.D. student and instructor in the Brian Lamb School of Communication at Purdue University, West Lafayette, IN, USA.

Both authors contributed equally to this work, and the names are listed alphabetically.

DOI:10.1145/3199201

**Expect inherent uncertainties in health-wearables data to complicate future decision making concerning user health.**

BY BRAN KNOWLES, ALISON SMITH-RENNER, FOROUGH POURSABZI-SANGDEH, DI LU, AND HALIMAT ALABI

# Uncertainty in Current and Future Health Wearables

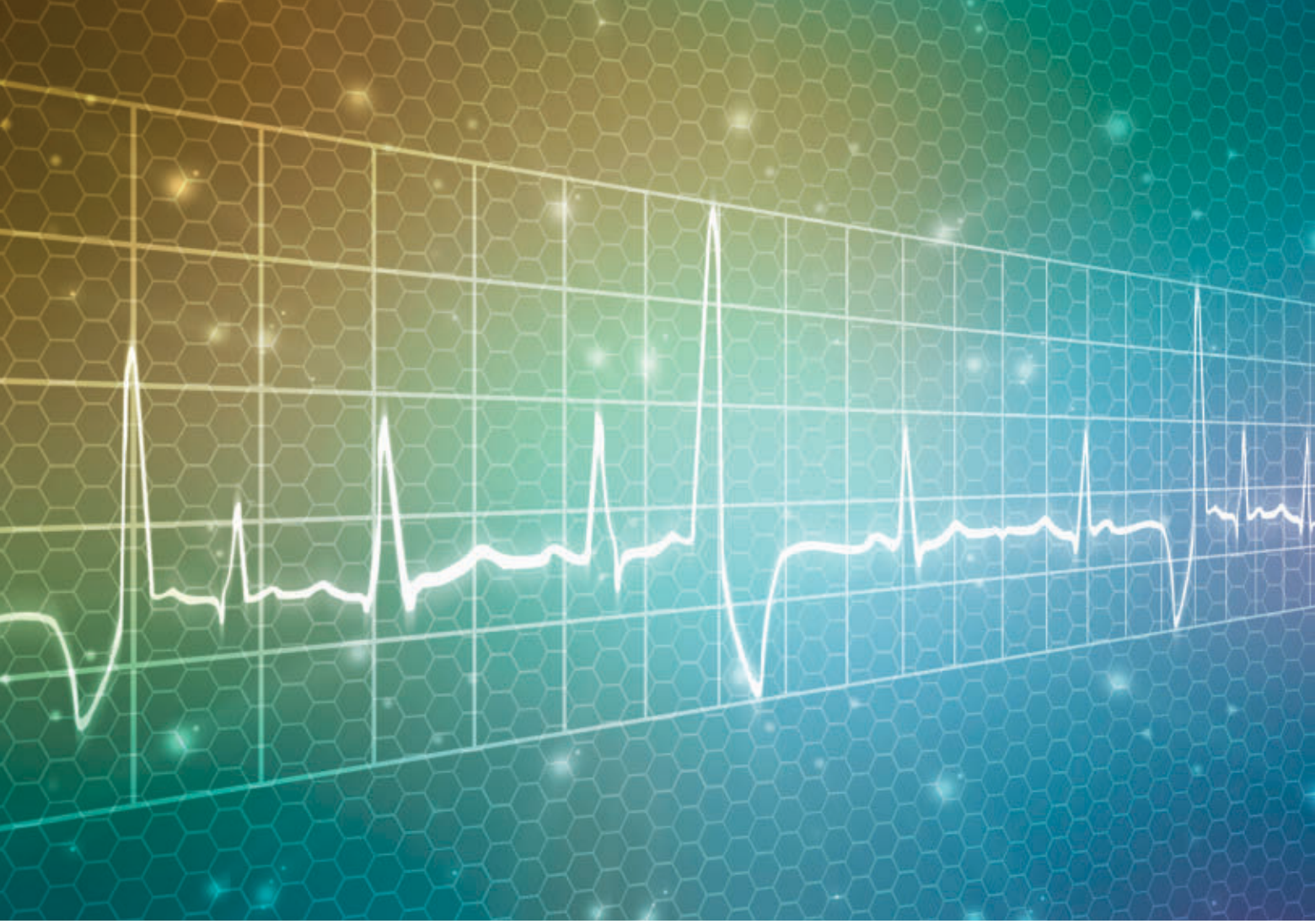
THERE IS DEMONSTRABLE appeal in consumer-wearable devices like activity trackers, having now been used by approximately 10% of American adults to track measures of their fitness or well-being.[4] Because activity trackers are most commonly used for motivating a change in behavior toward modest personal fitness goals or healthy activity levels over time,[8] it is easy to forget they are also used to inform more critical decision making and serious investigations of self, including tracking ongoing health conditions and disease progression;[24] tracking mood, with potential implications for mental-health treatment;[4] and self-diagnosing problems involving health or other concerns.[22]

These popular uses expose the potential variability of "uncertainty tolerance" among multiple users.[12] Those undertaking a serious investigation of self require a certain level of precision and data accuracy, as well as details regarding correlations between variables, whereas salient information for those with a casual interest in their fitness may simply want to know whether they have met some target or are generally improving over time. Technological advances, both recent and on the horizon for health wearables, are predicted by some experts to enable breakthroughs in disease prevention, prediction, and management, areas for which uncertainty tolerance differs significantly from that of the wearable consumer.[10] In addition to existing health wearables that claim to measure blood pressure, breathing rate, and mood or emotions and stress through galvanic skin response, wearables may soon be able to measure or infer health indicators like blood glucose, calories consumed, hydration, and heart strain (for details, see https://www.wareable.com/fitness-trackers).

Here, we explore the implications of, and difficulties in designing for, uncertainties regarding health wearables. We begin with the relatively minimal negative impact of uncertainty in current consumer uses of these gadgets as a way to demonstrate the known-but-as-yet-unresolved challenges in communicating health data to users. We next argue that seemingly innocuous uncertainties emerging in the present use of wearables need attending to, as they are likely to pro-

## » key insights

- **Unlike popular use scenarios involving today's consumers, more ambitious future use of health wearable data can be expected to involve even less tolerance for data uncertainty.**
- **Here, we explore the effect of uncertainty on several forward-looking scenarios in which health-wearable data is used to guide critical health decision making.**
- **Areas in need of further research include how to provide users access to confirmatory evidence of reliability, preserve provenance of uncertainties, and tailor communication about uncertainties to users.**



duce important consequences in the future. We raise three concerns in particular: First, advances in wearable technology will enable measurement of physiological data of which the user has little or no access to verifiable evidence (see the section in this article on emergency medical intervention and disease prevention). Second, low-level uncertainties are compounded by the interdependency between various data systems and their implications (such as for disease prevention, prediction, and management) (see the section on life coaching). And third, near-future scenarios involving external use of personal health data introduce new stakeholders whose tolerance for and ability to understand uncertainties will vary, requiring deeper research into ways to deal with uncertainties (see the section on patient compliance monitoring).

## Known Uncertainties of Consumer Wearables

For this purpose, we use the term "uncertainty" to mean a lack of understanding about the reliability of a particular input, output, or function of a system that could affect its trustworthiness. With wearable activity trackers, uncertainties arise in various forms and affect user trust to varying degrees. The consequences, while not always apparent to the user, also differ. Here, we explore some of the salient uncertainties that will be relevant to the discussion later in the article.

The old engineering principle says, "garbage in, garbage out," but it can be difficult to know whether the data coming into a system is sufficiently accurate to produce meaningful output—where "meaningful" is defined in relation to the user's needs; we call it "input uncertainty." Inaccuracies in data can be introduced by wearable users in various ways. For example, diagnostic tracking,[20] may require users to manually record instances of symptoms, food they have eaten, or medications they have taken. In such cases, the reliability of system outputs depends on users' ability to correctly infer what data their tracker is capable of automatically collecting[23] and their vigilance in manually collecting the rest, as well as the degree they are able to understand the standards for entering data and the importance of the precision of their input. Users often lack knowledge of how algorithms process their data and may thus fail to appreciate how imprecision in a single input could affect the overall system's ability to make appropriate recommendations. Supporting users' understanding of these impacts is difficult,[18] as few people have the requisite knowledge or interest in interrogating an algorithm. However, we suggest that supporting understanding and reducing input inaccuracies may be helped by following three practical guidelines: enable users to engage in a trial-interaction phase, where they are able to play around with different inputs to see the effects on calculated outputs; provide simple tips on the inputs that explain data-collection standards and the importance of precision; and/or provide some window into the underlying model and calculations.

IMAGE BY GREENTECH

# Understanding Health Wearables Data

**The virtually limitless opportunities for passive data collection through wearables mean any user has potentially large amounts of multidimensional data with which to make health-related decisions. To do so effectively, they must make sense of patterns within that data. This challenge is endemic to personal informatics, or "lived informatics,"[22] more generally, the goal being to "help people collect personally relevant information for the purpose of self-reflection and self-knowledge."[15] In the context of health wearables specifically, systems are typically designed to help users understand the effect of a range of contextual factors on a desired health outcome (such as well-being).[2]**

**Enabling user health revelations poses a significant information-presentation challenge. For example, users demonstrate poor graph literacy,[2] yet commercial-brand wearables interfaces are predominantly graph-based. These interfaces also tend to prioritize time-based views of data, smoothing out peaks and troughs and obscuring the most salient contexts around which they occur—information that would ostensibly lead to greatest user insight.[2,15] It is also often not readily apparent to users how the complexities of interactions between factors is negotiated by the system's algorithms,[2] nor whether such decision making is rooted in robust science. Complicating matters further, users generally have poor conceptual grounding for such concepts as "health," "well-being," and "fitness"; for example, Kay et al.[11] showed users are poorly equipped to determine the clinical relevance of weight-fluctuation data.**

**A growing body of work in HCI explores strategies for supporting intelligibility of data collected by health wearables (such as Bentley et al.,[2] Consolvo et al.,[6] Kay et al.,[11] Kay et al.,[12] Li et al.,[15] and Liu et al.[16]). This work is fundamental to attending to the challenges of uncertainty for health wearables, as it is indeed the basis for providing users insight into both when inaccuracies occur and the effect of inaccuracies in a reading or output relative to their intended use of the device.**

Input uncertainties also arise through onboard sensors. Notably, while guidelines for effective sensor placement are typically provided to users, user estimation of sensor accuracy is not. The reliability of fitness-tracker data has long been a source of concern in human-computer Interaction (HCI), and comparative evaluations of activity tracker brands reveal minimal though potentially significant differences in reliability.[3] While users of these tools are highly cognizant of their lack of reliability (such as with step counting[6] and sleep monitoring[16]), attempts to test devices for inaccuracies and calibrate use accordingly often fail.[18] Prevailing advice from designers is to enable users to annotate or amend their data if deemed inaccurate,[6,20] but users' ability to correct sensor errors is limited only to readings they are able to verify independently. As wearables begin to measure physiological data (such as heart strain) not otherwise accessible to the user, new design solutions will be needed to address input uncertainties.

Another type of uncertainty we call "output uncertainty" is apparent when users are unable to determine the significance of the inferences or recommendations produced by a system (see the sidebar "Understanding Health Wearables Data"). For example, many users of activity trackers struggle to understand how they compare with others (such as whether their readings are normal, exceptional, or worrying)[16] or whether they can claim to be "fit."[14] Even if users are able to determine their readings are outside what would be considered by medical doctors in the normal range, they routinely ask for guidance about what to do with the information.[14,15] Current tools do not provide the support users need to understand the significance of their data[3] and without it cannot determine the significance of uncertainties in that data.

While some evidence suggests providing users information about why a system behaved a certain way can increase trust[17] and not doing so (such as not providing uncertainty information) can lead to reduced trust,[11] a recent study found algorithm and system transparency does not necessarily yield more trust[21] and greater intelligibility tends to reduce trust when there are significant output uncertainties.[17] These points suggest questions that deserve further research; for example, when—or indeed for what users—is it appropriate to communicate how the systems collect and process data and how confident the systems are in their outputs? And, moreover, how should these uncertainties be communicated to maximize user trust?

A final notable concern is what we call "functional uncertainty" that emerges when users are unable to understand how, why, and by whom their data is being used. Concerns about privacy and security are manifestations of this uncertainty. It is not always apparent to users exactly what data is being collected from their devices, as well as the duration, location, or security level of their storage. For example, Epstein et al.[7] found that nearly half of the participants in their study turned off location tracking, fearing friends might be able to see where they were at all times or their location information might be sold to companies to better target ads. In certain contexts, a lack of location information might reduce the precision of other calculated metrics that depend on it. Further, consent terms and conditions being notoriously verbose and inaccessible, consumers may not fully understand the implications of the consent given when signing up with their devices.[1] This, in turn, can influence user compliance with recommended usage, introducing further input uncertainties.

We argue that for general fitness and well-being, the effect of the uncertainties we have just outlined are limited. They may contribute to loss of trust and high rates of device abandonment,[5] but while these consequences may be a concern for companies producing the gadgets, it is not especially problematic otherwise. However, our interest throughout the rest of this article is how the effect of these uncertainties could intensify in more ambitious uses of health-wearables data.

## Uncertainties in Future Uses

Here, we introduce three areas where we anticipate increased use of commercial activity-tracker data and explore how they may further affect uncertainty tolerance and thus implications in designing for uncertainty. We focus on these scenarios as a way to draw out three distinct concerns that require attending to in future research:

**Emergency medical intervention and disease prevention.** Health wear-

ables allow users to make sense of past events—what activities they have done and what effect they are likely to have on their well-being—to prompt positive behavior change, as discussed by Fritz et al.[8] The next stage of development might be for health wearables to predict health crises; examples include alerting a hospital of early signs of a heart attack or warning users of how likely it is they will develop, say, breast cancer.

The scenario involving predictive emergency medical intervention raises the question of who ought to have access to personal health data. While it would be helpful to link one's health data directly to the closest hospital in order to set the long chain of care in motion as early as possible in an emergency, there would be highly sensible consumer pushback around the access various parties might want to have to personal health data and that functional uncertainty in this arena would likely not be tolerated. Alternatively, if a wearable device alerted a user to hurry to a hospital at the start of a possible medical crisis, how certain does the device have to be? Should gadgets err on the side of caution, possibly provoking a false alarm? While not alerting a user due to insufficient certainty may lead to preventable deaths, so might causing alarm when alarm is not absolutely necessary, thus leading users to ignore or even reject subsequent alerts, with the gadget turning the user into "the boy who cried wolf."

The very notion of a health wearable alerting a user to an otherwise imperceptible impending crisis demonstrates the insufficiency of solutions for addressing uncertainty that rely on manual data correction by the user, as suggested by Consolvo et al.[6] and Packer et al.[20] Explaining the data collected and the ways it is processed by the algorithm may be more appropriate for assisting a user determining whether the device output is certain enough to warrant seeking medical attention. At the same time, this information must be delivered in ways that can be evaluated rationally by a person who just received an anxiety-provoking output (see the sidebar "Communicating Uncertainty"). Both parts of this solution are non-trivial and require further research.

**Life coaching.** Tracking data points through one's personal history is of limited value for individuals seeking improvements in and maintenance of their well-being, in contrast to information about dependencies and correlations among multiple variables[5] (such as the effect of certain foods on an individual's blood sugars). Given that users are often not rational data scientists[22] and consistent in asking for greater analytical capabilities than their devices are capable of providing, it seems inevitable that device manufacturers will introduce systems that purport to provide more definitive answers for users. The danger would be doing so without properly attending to the uncertainties highlighted earlier.

Users' inability to appreciate uncertainty is made especially clear in the case of wearables that claim to identify correlations between mood and activities (such as ZENTA, https://www.indiegogo.com/projects/zenta-stress-emotion-management-on-your-wrist). It is conceivable that wearable life coaches may soon draw from other pervasive technologies to provide indications of, say, toxic relationships between the user and other individuals and encouraging them to cut unhealthy social ties. While such revelations could have benefits, the implications of inaccuracies of one's data or of the data being drawn from other sources to determine correlations would begin to extend beyond the individual user, affecting others in the user's social circle who did not necessarily consent to such analysis. Additionally, the consequences to individuals deciding to cut a person out of their lives are not necessarily knowable to a system (such as how cutting ties might introduce undue financial instability into their lives). How certain would one have to be of the toxicity of a relationship to be willing to end it? It might indeed be the case that people would more readily accept diagnoses of their problems in the form of a scapegoat than that their unhappiness is a result of their own behaviors they find difficult to change. This is all the more reason why tools that claim deep insight into users' lives must be very clear about the uncertainties they are juggling in their algorithms.

For advanced diagnostic tracking in the form of life coaching, new techniques are needed to identify potential triggers from relevant contextual information; and to the extent that doing so entails drawing data from other pervasive devices, such a filter might introduce further uncertainties that need to be reflected in overall measures of uncertainty. Additional research is needed to understand how best to communicate these uncertainties to users. In particular, tools are needed for capturing users' cognitive and affective responses to these uncertainties (as covered in the sidebar "Communicating

## Communicating Uncertainty

**Experimental psychology studies, including those undertaken by Susan Joslyn and her colleagues at the University of Washington in Seattle (http://depts.washington.edu/forecast/), have shown that providing information about uncertainty can lead to greater trust in system models and better decision making. These benefits are far from assured, however, as studies have also shown that non-expert end users have great difficulty interpreting information about uncertainty.[12]**

**One presumed reason for this difficulty is that uncertainty increases the cognitive load individuals need to manage while making any kind of decision. It requires that individuals engage in slow and methodical thinking, as opposed to more quick and heuristic thinking. Verbal and numerical expressions of uncertainty create potential complications for decision makers—the verbal expressions being open to more subjective interpretative variability, and the numerical expressions often being more difficult to decipher.[9] Both forms are potentially subject to framing effects that influence people's processing of the information. Research has also uncovered "deterministic construal errors,"[9] or the tendency to interpret uncertain information as deterministic; for example, people frequently interpret—incorrectly—the "cone of uncertainty" in hurricane forecasts as the extent of the wind field, while, in fact, it represents the extent of all possible hurricane trajectories. All such factors require careful consideration when designing representations of uncertainty information.**

Uncertainty") and for capturing information regarding subsequent actions taken by users in order to improve uncertainty feedback visualizations and interfaces, as in Morris and Klentz[11] and Kay et al.[12]

**Patient compliance monitoring.** It has been argued by some technology experts that the commercial appeal of activity trackers for relatively affluent and active individuals has obscured the true potential of the devices for helping manage chronic illnesses—given that those with a true health need are significantly less likely to abandon their gadgets when the novelty has worn off.[10] If the degree of certainty in the reliability of activity-tracking data were to be better understood, such devices might be more readily accepted in the doctor's office as a way of inferring compliance with exercise plans and dietary advice, as discussed by Swan.[24] With this end in mind, we anticipate commercial wearables will advance to the point of being able to determine whether and when a patient is taking prescribed medication and at what dosage; the effects of such medication on their physiologies; and what other behavioral factors might be affecting symptoms.

Doctors could thus disambiguate factors that are affecting a patient's health. This is important information for determining the accuracy of patient self-reports, which can be flawed for any number of reasons, ranging from innocuous memory failings to subjective interpretation of one's experiences to intentional misrepresentation or deception. To the extent that patients understand noncompliance is detectable by their doctors, this may indeed promote greater compliance. On the other hand, the use of wearables as an objective (certain) measure may result in greater emphasis being placed on quantitative data than on the patient's own anecdotal reports. Inconsistencies between the two accounts that arise as a result of uncertainties surrounding the wearable data or input uncertainties relating to sensor-error rates and the device having been used incorrectly by the patient could have negative implications for the dynamic of patient-doctor trust if the uncertainties are not clarified by both parties.

Just as it is not always clear how to communicate uncertainties to the average consumer, it is also not clear how to communicate uncertainties to doctors. Effective communication of uncertainties may take different forms between these two groups. For example, doctors may be more comfortable interpreting raw data or graphs or need data in a certain form to be compatible and comparable with their existing patient records. It may take new training to be able to interpret results from commercial wearables within the standard assessment frameworks (these practices may need to evolve), as well as further training for dealing with patients who may have drawn their own (possibly false or irrelevant[11]) conclusions from their personal devices.

**Seemingly innocuous uncertainties emerging in the present use of wearables need attending to, as they are likely to produce important consequences in the future.**

Entering the consumer health-wearables market also raises potential ethical questions, including whether patients want their doctors to know everything they do. If not, research is needed to determine how to strike the appropriate balance in data that supports serious medical decision making while preserving plausible deniability for the patient.

## Conclusion

Future research is needed to address these questions and the trade-offs they imply:

*How to provide access to confirmatory evidence of reliability.* An inherent problem of many pervasive sensor technologies is the data recipient (whether doctor or patient) has little or no way to verify the data's accuracy.[13] In the case of health wearables, users might have some general sense of whether they are, say, dehydrated or have low blood sugar but are unlikely to be able to put an exact number on the measurements. So how might future health wearables provide access to confirmatory evidence of their precision? Doing so would be especially useful for enabling users to help with device calibration, as discussed by Mackinley,[18] to help mitigate at least some potential input uncertainties.

*How to preserve provenance of uncertainties.* Due to the trend toward greater interdependence between data systems, with outputs from one system being churned through the algorithms of others,[13] it is conceivable

that data from individuals' self-tracking devices will be used as bedrock data in other systems from which a range of inferences are made. Ensuring uncertainties are preserved and communicated throughout a long chain of systems whose developers and interpreters might have different readings of these uncertainties and tolerances for them is challenging but necessary if the systems are to be interpretable at scale, as discussed by Meyer et al.[19] This requires development of mechanisms for ensuring important context is not lost, including, say, both the uncertainties and uncertainty tolerances at different points along the chain.

*How to tailor communication of uncertainties.* Designs must be flexible and/or customizable, presenting uncertainty information in ways that are understandable by the full range of end users with differing needs in data granularity and information presentation. Given that much of the value of health wearables for lay consumers comes from data being available at-a-glance, there is a need to balance important nuance with the interface usability, as discussed by Liu et al.[16] Still, there are moments when even lay consumers could require access to uncertainty information, with systems perhaps allowing them to delve more deeply, as required. Contextual information (such as users' intended and ongoing use of their wearable data) might also be useful for determining what kinds of uncertainty information the system ought to communicate. At the same time, designers must be cognizant of user variability in cognitive and affective responses to uncertainty-related information to design systems that can identify, learn from, and adapt to these responses to inform health-related decision making most effectively.

These design implications can be considered an open challenge to the health-wearables community without suggesting precise mechanisms for realizing them through design.

## Acknowledgments

This article resulted from group work that was part of the CHI 2017 workshop "Designing for Uncertainty in HCI: When Does Uncertainty Help?"; http://visualization.ischool.uw.edu/hci_uncertainty/

We wish to thank our fellow workshop participants and keynote speaker, Susan Joslyn, for their feedback in developing these ideas. And in particular we thank the workshop's organizers—Miriam Greis, Jessica Hullman, Michael Correll, Matthew Kay, and Orit Shaer—for providing a forum for discussing these ideas and bringing this team of authors together for ongoing collaboration post workshop.

Finally, we also thank the anonymous reviewers for their help shaping and improving this article.

**References**

1. Barcena, M.B., Wueest, C., and Lau, H. *How safe is your quantified self?* Symantec, Inc., 2014; https://www.symantec.com/content/dam/symantec/docs/white-papers/how-safe-is-your-quantified-self-en.pdf
2. Bentley, F., Tollmar, K., Stephenson, P., Levy, L., Jones, B., Robertson, S., Price, E., Catrambone, R., and Wilson, J. Health mashups: Presenting statistical patterns between well-being data and context in natural language to promote behavior change. *ACM Transactions on Computer-Human Interactions 20*, 5 (Nov. 2013), 1–27.
3. Case, M.A., Burwick, H.A., Volpp, K.G., and Patel, M.S. Accuracy of smartphone applications and wearable devices for tracking physical activity data. *Journal of the American Medical Association 313*, 6 (Feb. 2015), 625–626.
4. Choe, E.K., Lee, N.B., Lee, B., Pratt, W., and Kientz, J.A. Understanding quantified-selfers' practices in collecting and exploring personal data. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems* (Toronto, ON, Canada, Apr. 26–May 1). ACM Press, New York, 2014, 1143–1152.
5. Clawson, J., Pater, J.A., Miller, A.D., Mynatt, E.D., and Mamykina, L. No longer wearing: Investigating the abandonment of personal health-tracking technologies on Craigslist. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Osaka, Japan, Sept. 7–11). ACM Press, New York, 2015, 647–658.
6. Consolvo, S., McDonald, D.W., Toscos, T., Chen, M.Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., Smith, I., and Landay, J. Activity sensing in the wild: A field trial of UbiFit Garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, Apr. 5–10). ACM Press, New York, 2008, 1797–1806.
7. Epstein, D.A., Caraway, M., Johnston, C., Ping, A., Fogarty, J., and Munson, S. A. Beyond abandonment to next steps: Understanding and designing for life after personal informatics tool use. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, CA, May 7–12). ACM Press, New York, 2016, 1109–1113.
8. Fritz, T., Huang, E.M., Murphy, G.C., and Zimmermann, T. Persuasive technology in the real world: A study of long-term use of activity sensing devices for fitness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, ON, Canada, Apr. 26–May 1). ACM Press, New York, 2014, 487–496.
9. Grounds, M.A., Joslyn, S., and Otsuka, K. Probabilistic interval forecasts: An individual differences approach to understanding forecast communication. *Advances in Meteorology* (2017).
10. Herz, J. Wearables are totally failing the people who need them most. *Wired* (Nov. 6, 2014); https://www.wired.com/2014/11/where-fitness-trackers-fail/
11. Kay, M., Morris, D., and Kientz, J.A. There's no such thing as gaining a pound: Reconsidering the bathroom scale user interface. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Zurich, Switzerland, Sept. 8–12). ACM Press, New York, 2013, 401–410.
12. Kay, M., Patel, S.N., and Kientz, J.A. How good is 85%? A survey tool to connect classifier evaluation to acceptability of accuracy. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea, Apr. 18–23). ACM Press, New York, 2015, 347–356.
13. Knowles, B. Emerging trust implications of data-rich systems. *IEEE Pervasive Computing 15*, 4 (Oct. 2016), 76–84.
14. Lazar, A., Koehler, C., Tanenbaum, J., and Nguyen, D.H. Why we use and abandon smart devices. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Osaka, Japan, Sept. 7–11). ACM Press, New York, 2015, 635–646.
15. Li, I., Dey, A., and Forlizzi, J. A stage-based model of personal informatics systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, GA, Apr. 10–15). ACM Press, New York, 2010, 557–566.
16. Liu, W., Ploderer, B., and Hoang, T. In bed with technology: Challenges and opportunities for sleep tracking. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction* (Parkville, VIC, Australia, Dec. 7–10). ACM Press, New York, 2015, 142–151.
17. Lim, B.Y., Dey, A.K., and Avrahami, D. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, Apr. 4–9). ACM Press, New York, 2009, 2119–2128.
18. Mackinlay, M.Z. Phases of accuracy diagnosis: (In) visibility of system status in the FitBit. *Intersect: The Stanford Journal of Science, Technology and Society 6*, 2 (June 2013).
19. Meyer, J., Wasmann, M., Heuten, W., El Ali, A., and Boll, S.C. Identification and classification of usage patterns in long-term activity tracking. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, CO, May 6–11). ACM Press, New York, 2017, 667–678.
20. Packer, H.S., Buzogany, G., Smith, D.A., Dragan, L., Van Kleek, M., and Shadbolt, N.R. The editable self: A workbench for personal activity data. In *Proceedings of CHI 2014 Extended Abstracts on Human Factors in Computing Systems* (Toronto, ON, Canada, Apr. 26–May 1). ACM Press, New York, 2014, 2185–2190.
21. Poursabzi-Sangdeh, F., Goldstein D.G., Hofman J.M., Wortman Vaughan, J., and Wallach H. Manipulating and measuring model interpretability. arXiv preprint, 2018; https://arxiv.org/pdf/1802.07810
22. Rooksby, J., Rost, M., Morrison, A., and Chalmers, M.C. Personal tracking as lived informatics. In *Proceedings of the 32nd annual ACM Conference on Human Factors in Computing Systems* (Toronto, ON, Canada, Apr. 26–May 1). ACM Press, New York, 2014, 1163–1172.
23. Shih, P.C., Han, K., Poole, E.S., Rosson, M.B., and Carroll, J. M. Use and adoption challenges of wearable activity trackers. *IConference Proceedings* (2015); https://www.ideals.illinois.edu/bitstream/handle/2142/73649/164_ready.pdf
24. Swan, M. Emerging patient-driven health care models: An examination of health social networks, consumer personalized medicine and quantified self-tracking. *International Journal of Environmental Research and Public Health 6*, 2 (Feb. 2009), 492–525.

**Bran Knowles** (b.h.knowles1@lancaster.ac.uk) is a lecturer in data science at Lancaster University, Lancaster, U.K.

**Alison Smith-Renner** (alison.smith@dac.us) leads the Machine Learning Visualization Lab at Decisive Analytics Corporation, Arlington, VA, USA, and is a Ph.D. candidate in computer science at the University of Maryland, College Park, MD, USA.

**Forough Poursabzi-Sangdeh** (forough.poursabzi@microsoft.com) is a post-doctoral researcher at Microsoft Research NYC, USA.

**Di Lu** (di.lu@pitt.edu) is a Ph.D. student in the School of Information Sciences at the University of Pittsburgh, Pittsburgh, PA, USA.

**Halimat Alabi** (halabi@sfu.ca) is an adjunct in the Art Institute Online and a Ph.D. candidate in the School of Interactive Art and Technology at Simon Fraser University, Vancouver, BC, Canada.

DOI:10.1145/3198470

**A series of reports promises the general public a technologically accurate view of the state of AI and its societal implications.**

BY BARBARA J. GROSZ AND PETER STONE

# A Century-Long Commitment to Assessing Artificial Intelligence and Its Impact on Society

The *Stanford One Hundred Year Study on Artificial Intelligence*, a project that launched in December 2014, is designed to be a century-long periodic assessment of the field of artificial intelligence (AI) and its influences on people, their communities, and society. Colloquially referred to as "AI100," the project issued its first report in September 2016. A standing committee of AI scientists and scholars in the humanities and social sciences working with the Stanford faculty director of AI100 oversees the project and the design of its activities. A little more than two years after the first report appeared, we reflect on the decisions made in shaping it, the process that produced it, its major conclusions, and reactions subsequent to its release.

The inaugural AI100 report,[6] called *Artificial Intelligence and Life in 2030*, examined eight domains of human activity in which AI technologies are already beginning to affect urban life. In scope, it encompasses domains with emerging products enabled by AI methods and domains, raising concerns about technological impact generated by potential AI-enabled systems. The study panel members who wrote the report and the AI100 standing committee, the body that directs the AI100 project, intend for it to be a catalyst, spurring conversations on how we as a society might shape and share the potentially powerful technologies AI could deliver. In addition to influencing researchers and guiding decisions in industry and governments, the report aims to provide the general public with a scientifically and technologically accurate portrayal of the current state of AI, along with that potential. It aspires to replace conceptions rooted in science fiction novels and movies with a realistic foundation for these deliberations.

The report focuses on AI research and "specialized AI technologies," or methods developed for and tailored to particular applications, that are increasingly prevalent in daily activities rather than deliberating about generalized intelligence, which is often mentioned in

## » key insights

- **Here, we describe the first report of the *100 Year Study on Artificial Intelligence*, which will regularly, over a century or more, assess past accomplishments, current status, and future potential of AI science and technologies and their possible effects on society.**
- **The 2016 inaugural report presents the consensus perspective of active AI researchers and scholars in related areas of social sciences, striving to be neither overly optimistic nor overly pessimistic.**
- **While drawing on common research and technologies, AI systems are specialized to accomplish particular tasks, with each application requiring years of focused research and unique construction.**

IMAGE BY TREETY

the media and is much further from realization. It anticipates that AI-enabled systems have great potential to improve people's daily lives worldwide and positive impact on economies worldwide but also create profound societal and ethical challenges. It thus argues that deliberations involving the broadest possible spectrum of expertise about AI technologies and the design, ethical, and policy challenges they raise should begin now to ensure the benefits of AI are broadly shared, as well as that systems are safe, reliable, and trustworthy.

In the rest of this article, we provide background on AI100 and the framing of its first report, then discuss some of its findings. Along the way, we address several questions posed to us during the years since the report first appeared and catalog some of its uses.

## Influences and Origins

The impetus for the AI100 study came from the many positive responses to a 2008–2009 Association for the Advancement of Artificial Intelligence Presidential Panel on Long-Term AI Futures that was commissioned by then-AAAI President Eric Horvitz (Microsoft Research) and co-chaired by him and Bart Selman (Cornell University). Intending a largely field-internal reflection on the state of AI, Horvitz charged the panel with exploring "the potential long-term societal influences of AI advances." In particular, he asked them to consider AI successes and the societal opportunities and challenges they raised; the socioeconomic, ethical, and legal issues raised by AI technologies; proactive steps those in the field could take to enhance long-term societal outcomes; and the kinds of policies and guidelines needed for autonomous systems. The findings of the panel (http://www.aaai.org/Organization/presidential-panel.php) and reactions to it led Horvitz to design AI100, a long-horizon study of how AI advances influence people and society. It is intended to pursue periodic studies of developments, trends, futures, and potential disruptions associated with developments in machine intelligence and formulate assessments, recommendations, and guidance on proactive efforts. The new project was to be balanced in its inward (within the AI field) and outward-looking (other disciplines and society at large) faces. The long-term nature of the project is its most novel aspect, as it is intended to periodically (typically every five years) assemble a study panel to reassess the state of AI and its impact on society. A "framing memo" (https://ai100.stanford.edu/reflections-and-framing) laid out Horvitz's aspirations for the project, along with the reasons for situating it at Stanford University.

## First Step

**Assemble a study panel.** As the AI100 project was launched in December 2014, the standing committee anticipated that several years would be available for shaping the project, engaging people with expertise across the social sciences and humanities, as well as AI, identifying a focal topic, and recruiting a study panel. Within a few months, however, it was clear that AI was entering daily life and garnering intense public interest at a rate that did not allow such a leisurely pace. The standing committee thus defined a compressed schedule and recruited Peter Stone of The University of Texas at Austin (co-author of this article) as the chair of the report's study panel. Together they assembled a 17-member study panel comprising experts in AI from academia, corporate laboratories, and industry, and AI-savvy scholars in law, political science, policy, and economics. Although their goal was a panel diverse in specialty and expertise, geographic region, gender, and career stages, the shortened time frame led it to be less geographically and field diverse than ideal, a point noted by several report readers. In recognition of these shortcomings, and as it considers the design of future studies, the steering committee, which has increased its membership to include more representation from the social sciences and humanities, has developed a more inclusive planning and reporting process.

**Design the charge.** The standing committee considered various possible themes and scopes for the inaugural AI100 report, ranging from a general survey of the status of research and applications in subfields to an in-depth examination of a particular technology (such as machine learning and natural language processing) or an application area (such as healthcare and transportation). Its final choice of topical focus reflects a desire to ground the report's assessments in a context that would bring to the fore societal settings and a broad array of technological developments. The focus on *AI and Life in 2030* arose from recognition of the central role cities have played throughout most of human history, as well as a venue in which many AI technologies are likely to come together in the lives of individuals and communities. The further focus on North American cities followed from recognition that within the short time frame allowed by the panel's work, it was not possible to adequately consider the great variability of urban settings and cultures around the world. Although the standing committee expects the projections, assessments, and proactive guidance stemming from the study to have broader global relevance, it intends for future studies to have greater international involvement and scope.

The charge the standing committee communicated to the study panel asked it to identify possible advances in AI over 15 years and their potential influences on daily life in urban settings (with a focus on North American cities), to specify scientific, engineering, and policy and legal efforts to realize these developments, consider actions to shape outcomes for societal good, and deliberate on the design, ethical, and policy challenges the developments raise. It further stipulated that the study panel ground its examination of AI technologies in a context that highlights interdependencies and interactions among AI subfields and these technologies and their potential influences on a variety of activities.

**Create the first report.** In the absence of precedent and with a short time horizon for its work, the study panel engaged in a sequence of virtually convened brainstorming sessions in which it successively refined the topics to consider in the report, with the aim of identifying domains, or economic sectors, in which AI seemed most likely to have impact within urban settings between publication of the report and 2030. Then, during a full-day intensive writing session during an in-person meeting at the 2016 AAAI conference in February, they drafted several report sections. They then iteratively revised these drafts with the goal of producing a report that would be accessible to the general public and convey the study panel's key messages. At a final in-person meeting in July at the

2017 International Joint Conferences on Artificial Intelligence, the study panel identified the main messages of the report, to appear as callouts in the margins of the report.

### The Report

The report aims to address multiple audiences, ranging from general public to AI researchers and practitioners, and thus to be both accessible and provide depth. As a result, it has a three-part hierarchical structure: executive summary, more expansive five-page overview summarizing the core of the report, and a core with further details. The core examines eight "domains" of typical urban settings on which AI is likely to have impact over the coming years: transportation, home and service robots, healthcare, education, public safety and security, low-resource communities, employment and workplace, and entertainment. The authors deliberately did not give much weight to positions they considered excessively optimistic or pessimistic, despite the prevalence of such positions in the popular press, as they intended the report to provide a sober assessment by the people at the heart of technological developments in AI.

For each domain the study panel investigated, the report looks back to 2000 to summarize the AI-enabled changes that have already occurred and then project forward through 2030. It identifies the availability of large amounts of data, including speech and geospatial data, as well as cloud computing resources and progress in hardware technology for sensing and perception, as contributing to recent advances in AI research and to the success of deployed AI-enabled systems. Advances in machine learning, fueled in part by these resources, as well as by development of "deep" artificial neural nets, have played a key role in enabling these achievements. The goal of the study panel's forward-looking assessment, which we summarize briefly, was to call attention to the opportunities the study panel anticipated for AI technologies to improve societal conditions, lower the barriers to realizing this potential, and address the realistic risks it likewise anticipated in applying AI technologies in the domains it studied.

The projected time horizons for AI-enabled systems to enter daily life vary across these domains, as do the opportunities for transforming people's lives and the challenges posed in each domain. Moreover, the challenges so identified ranged across the full spectrum of computer science, from hardware to human-computer interaction. For instance, improvements in safe, reliable hardware were determined to be essential for progress in transportation and home-service robots. Autonomous transportation, which the report projects, may "soon be commonplace," is among today's most visible AI applications; in addition to changing individuals' driving needs, it is expected to affect transportation infrastructure, urban organization, and jobs. Experience with home-service robots illustrates the key role of hardware. Although robotic vacuum cleaners have been in home use for years, technical constraints and the high cost of reliable mechanical devices has limited commercial opportunities to narrowly defined applications; the report projects they will do so for the foreseeable future. For healthcare, the challenges so highlighted include developing mechanisms for sharing data, removing policy, regulatory, and commercial obstacles, and enhancing the ability of systems to work naturally with care providers, patients, and patients' families. The report also identifies capabilities for fluent interactions and effective partnering with people as key to achieving the promise of AI technologies for enhancing education. Major challenges toward realizing the potential of AI to address the needs of low-resource communities include design of methods to cooperate with agencies and organizations working in those communities and the development of trust of AI technologies by these groups and by the communities they serve. Such challenges also arise in public safety and security. In the domain of employment and the workplace, while noting that AI-capable systems will replace people in some kinds of jobs, the report also predicts AI capabilities are more likely to *change* jobs by replacing tasks than by eliminating jobs. It highlights the role of social and political decisions in approaching a range of societal challenges that will arise as work evolves in response to AI technologies and argues these challenges should be addressed immediately.

**It aspires to replace conceptions rooted in science fiction novels and movies with a realistic foundation for these deliberations.**

In assessing "What's next?" in AI research, the report says: " ... as it becomes a central force in society, the field of AI is shifting toward building intelligent systems that can collaborate effectively with people, and that are more generally human-aware." It also identifies several "hot areas" of AI research and applications. For example, in the area of machine learning, it describes efforts in scaling to work with very large datasets, deep learning, and reinforcement learning. Robotics, computer vision, and natural language processing (including spoken language systems) already incorporated into a variety of applications have made great strides recently and are poised for further advances. Research in two relatively newer areas—collaborative systems and crowdsourcing/human computation—are developing methods, respectively, for AI systems to work effectively with people and for people to assist AI systems in computations that are more difficult for machines than for people. Other research areas the report highlights are algorithmic game theory and computational social choice, the Internet of Things, and neuromorphic computing.

The report concludes with a section on policy and legal issues, summarizing the study panel's views on the state of regulatory statutes relevant to AI technologies and includes its recommendations to policymakers. It notes that "The measure of success for AI applications is the value they create for human lives. In that light, they should be designed to enable people to understand AI systems, participate in their use, and build their trust." The report encourages "vigorous and informed debate" about AI capabilities and limitations, recommending that much broader and deeper understanding of AI is needed in government at all levels to enable expert assessments of AI technologies, programmatic objectives, and overall societal values. It argues that industry needs to formulate and deploy best practices, and that AI systems should be open or amenable to reverse engineering so they can be evaluated adequately with respect to such crucial issues as fairness, security, privacy, and social impacts by disinterested academics, government experts, and journalists. It also notes the importance of expertise across a variety of disciplinary areas being brought to bear assessing societal impact and thus the need for increased public and private funding for interdisciplinary studies of the societal impacts of AI.

**While noting that AI-capable systems will replace people in some kinds of jobs, the report predicts AI capabilities are more likely to *change* jobs by replacing tasks than by eliminating jobs.**

Here, we list several of the report's most important takeaways and findings. We hope it provides a sense of the scope of the report and encourages reading the report, at least at one of the levels of detail provided:

*General observations.* Like other technologies, AI has the potential to be used for good or for nefarious purposes. A vigorous and informed debate about how to best steer AI in ways that enrich our lives and our society is an urgent and vital need. As a society, we are today *underinvesting* resources in research on the societal implications of AI technologies. Private and public dollars should be directed toward interdisciplinary teams capable of analyzing AI from multiple angles. Misunderstandings about what AI is and is not could fuel opposition to technologies with the potential to benefit everyone. Poorly informed regulation that stifles innovation would be a tragic mistake.

*Potential near-term applications and design constraints.* While many AI-based systems draw on common research and technologies, all such existing systems are specialized to accomplish particular tasks. Each application requires years of focused research and unique construction. AI-based applications could improve health outcomes and quality of life for millions of people in the coming years but only if they win the trust of doctors, nurses, and patients. Though quality education will always require active engagement by human teachers, AI promises to enhance education at all levels, especially through personalization at scale. With targeted incentives and funding priorities, AI technologies could help address the needs of low-resource communities. Budding efforts (such as those reported in recent workshops on AI and social good[2,3]) are promising.

*Societal concerns.* As highlighted in the movie *Minority Report* and subsequently reported by ProPublica,[1] predictive-policing tools raise the specter of innocent people being unjustifiably targeted. But well-designed and appropriately deployed AI prediction tools have potential to remove or at least reduce human bias. AI will likely replace tasks

rather than jobs in the near term and also create new kinds of jobs. But imagining what new jobs will emerge is more difficult in advance than is identifying the existing jobs that will likely be lost. As AI applications engage in behavior that, if done by a human, would constitute a crime, courts and other legal actors will have to puzzle through whom to hold accountable and on what theory.

## Reactions and Uses

Even more than when the AI100 project was first planned in 2014, we are at a crucial juncture in determining how to deploy AI-based technologies in ways that support societal needs and promote rather than hinder democratic values of freedom, equality, and transparency. The philosopher J.H. Moor wrote[5] that in ethical arguments, most often people agree on values but not on the facts of the matter. This first AI100 report aims to bring AI expertise to the forefront so the challenges, as well as the promise, of technologies that incorporate AI methods can be understood and assessed properly.

Although the report's impact over time remains to be seen, we hope it will establish a strong precedent for future AI100 study panels. We are gratified to have seen that since the report first appeared, it seems to have succeeded in this aim, along with the larger AI 100 goals, in several ways. For instance, shortly after it was released, September 1, 2016, it was covered widely in the press, including in the *New York Times*, *Christian Science Monitor*, NPR, BBC, and CBC radio. It helped shape a series of workshops sponsored by the White House Office of Science and Technology Policy and the reports that emanated from them.[4] Requests for permission to translate the report into several languages demonstrate worldwide interest. Various members of the AI 100 standing committee and the inaugural study panel have been asked to organize workshops for various governmental and scientific organizations and give talks in many settings. The study panel chair (and co-author of this article) was invited to speak by the Prime Minister of Finland, Juha Sipilä, on the occasion of his announcement of a new "AI strategy" for Finland, in February 2017; http://valtioneuvosto.fi/live?v=/vnk/events-seminars/professori-peter-stonen-puhe-tekoalyseminaarissa. The report is also being used in AI classes in various ways.

## Looking Forward

AI technologies are becoming ever more prevalent, and opinions on their impact on individuals and societies vary widely, from those the (inaugural) study panel considered overly optimistic to others it considered overly pessimistic. The need for the general public, government, and industry to have reliable information is of increasing importance. The AI100 project aims to fill that need. This first report is an important initial step, launching a long-term project. It crucially illuminates the enormous technical differences between AI technologies that are developed and targeted toward specific application domains and a "general-purpose AI" capability that can be incorporated into any device to make it more intelligent. The former is the focus of much research and business development, while the latter remains science fiction. It is quite tempting to think that if AI technologies can help drive our cars, they ought to also be able to fold our laundry, but these two activities make very different types of demands on reasoning. They require very different algorithms and capabilities. People do both, along with a full range of equally distinct activities requiring intelligence of various sorts. However, current AI applications are based on specialized domain-specific methods, and the normal human inclination to generalize from one intelligent behavior to seemingly related ones leads some people astray when assessing machine capabilities. This first AI100 report aims to provide insights to its readers, enabling them to better assess the implications of any AI success for other open challenges, as well as alert them to the societal and ethical issues that must be addressed as AI pervades ever more areas of daily life.

Since publishing the inaugural study panel's report, the AI100 project has begun a complementary effort, the Artificial Intelligence Index (AI Index), an ongoing tracking activity led by a steering committee of Yoav Shoham, Ray Perrault, Erik Brynjolfsson, Jack Clark, John Etchemendy, Terah Lyons, and James Maniyka. It complements the major studies originally envisioned for AI100 by providing annual reports and, in the future, an ongoing blog to augment the periodic AI100 studies to be produced by future study panels. The AI Index follows various facets of AI, including those related to volume of activity, technological progress, and societal impact, as determined by a broad advisory panel with advice from the AI100 standing committee. As with the study panel reports, the AI Index aims to provide information on the status of AI that is useful for those both outside the field and those engaged in developing AI technologies, as well as those actively involved in AI research and applications, policymakers and business executives, and the general public. This nascent effort issued its first report in December 2017.

The AI100 project (https://ai100.stanford.edu/) welcomes advice as it plans its next report, as does the AI Index (http://aiindex.org/). We look forward to following, and continuing to help shape, the AI100 trajectory over the coming years. C


**References**

1. Angwin, J. et al. Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks. *ProPublica* (May 23, 2016); https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
2. Association for the Advancement of Artificial Intelligence. AAAI Workshop on AI and OR [Operations Research] for Social Good (San Francisco, CA, Feb. 2017); https://www.aaai.org/Library/Workshops/ws17-01.php
3. Computing Community Consortium Workshop on AI and Social Good (Washington, D.C., June 7, 2016); https://cra.org/ccc/events/ai-social-good/
4. Felten, E. and Lyons, T. *The Administration's Report on the Future of Artificial Intelligence.* The White House, Oct. 12, 2016; https://obamawhitehouse.archives.gov/blog/2016/10/12/administrations-report-future-artificial-intelligence
5. Moor, J.H. What is computer ethics? *Metaphilosophy 16*, 4 (Mar. 1985), 266–275.
6. Stone, P., Brooks, R., Brynjolfsson, E., Calo, R., Etzioni, O., Hager, G., Hirschberg, J., Kalyanakrishnan, S., Kamar, E., Kraus, S., Leyton-Brown, K., Parkes, D., Press, W., Saxenian, A., Shah, J., Tambe, M., and Teller, A. *Artificial Intelligence and Life in 2030. One Hundred Year Study on Artificial Intelligence (A100): Report of the 2015-2016 Study Panel.* Stanford University, Stanford, CA, Sept. 2016; http://ai100.stanford.edu/2016-report



**Barbara J. Grosz** (grosz@eecs.harvard.edu) is Higgins Professor of Natural Sciences on the Computer Science faculty of the John A. Paulson School of Engineering and Applied Sciences at Harvard University, Cambridge, MA, USA, and a member of the External Faculty of Santa Fe Institute, Santa Fe, NM, USA. She was Inaugural Chair of the standing committee for the One Hundred Year Study on Artificial Intelligence.

**Peter Stone** (pstone@cs.utexas.edu) is the David Bruton, Jr. Centennial Professor in the Department of Computer Science at The University of Texas at Austin, Austin, TX, USA, and President and COO of Cogitai, Inc. He was Chair of the inaugural study panel of the One Hundred Year Study on Artificial Intelligence.

DOI:10.1145/3186591

**Emotionally sentient systems will enable computers to perform complex tasks more effectively, making better decisions and offering more productive services.**

BY DANIEL MCDUFF AND MARY CZERWINSKI

# Designing Emotionally Sentient Agents

TODAY, PEOPLE INCREASINGLY rely on computer agents in their lives, from searching for information, to chatting with a bot, to performing everyday tasks. These agent-based systems are our first forays into a world in which machines will assist, teach, counsel, care for, and entertain us. While one could imagine purely rational agents in these roles, this prospect is not attractive for several reasons, which we will outline in this article. The field of affective computing concerns the design and development of computer systems that sense, interpret, adapt, and potentially respond appropriately to human emotions. Here, we specifically focus on the design of affective agents and assistants. Emotions play a significant role in our decisions, memory, and well-being. Furthermore, they are critical for facilitating effective communication and social interactions. So, it makes sense that the emotional component surrounding the design of computer agents should be at the forefront of this design discussion.

Consider the following examples: Personal assistants (PAs) have become ubiquitous in our everyday computing lives. From well-known services like Amazon's Alexa, Apple's Siri, Microsoft's Cortana, or Google Assistant, to chat bots for areas such as customer service and training, consumers are familiar with the concept of a computerized PA. We argue that for a PA to truly become valuable to the user, it must be natural to interact with and engaging. How do we design a PA that is liked, fun, and easy to work with, and most importantly, trustworthy? Several researchers have shown that an assistant that can sense a user's social cues and affective signals along with her context, and respond appropriately, is valued more, considered more intelligent, and creates a greater desire by the user to interact with it.[4,17]

As they move into the digital era, healthcare and mental healthcare are seeing vast benefits from the influx of technology and machine learning. However, few systems effectively track the emotional health of their users—most of the time this is done via

## » key insights

- **Systems that respond to social and emotional cues can be more engaging and trusted, enabling computers to perform complex tasks in a more socially acceptable manner.**
- **Emotionally sentient agents present the exciting potential for large-scale, in-situ experimentation and user experience testing. Large-scale analysis of affective data from everyday contexts is important for improving affective computing systems, and helping us learn more about human expression and well-being.**
- **It is important for designers to consider the specifications of emotionally aware systems. Learning purely from human-human behavior may not always be the most effective approach and an affective agent may raise users' expectations of competence that the system may not possess.**

ALL IMAGES BY MDI DIGITAL


SAMSUNG

paper forms filled out before a doctor or therapy visit. The problem is that memory limits render these methods less effective over extended periods of time and are associated with demand effects (changes in behavior resulting from cues as to what constitutes appropriate behavior.) Computer programs can now track consumer and patient health, allowing for mining of that data for ideal intervention timing and personal reflection by the individual user of what makes them feel positive or not.[24] Recent efforts have successfully used conversational agents to automate the assessment and evaluation of psychology treatments.[25] Conversational agents could help with social support, wellness counseling, task completion, and safety, if they are designed with the ability to sense and manage affect and social interaction. This promising new direction could, for example, stave off rampant problems of loneliness in the elderly.[31]

Researchers have argued the relationship between a tutor and a learner plays an important role in improving educational results.[39] New educational platforms (for example, EdX and Coursera) are asynchronous and distributed. Automated tutoring systems designed with the ability to understand students' affective responses are very promising.[11] There is also growing literature on using affective agents in training simulations, (for example, by the military), to improve realism, evoke empathy, and even stir fear.[15] These simulations are critical for preparing soldiers, medical staff, and other personnel for the realities of combat zones and environmental catastrophes.

Affective computing brings newfound realism and immersion to entertainment applications, such as games, interactive media exhibits, and shows. In fact, companies have recently tracked their audience's affective response as it was presented with variants of commercials and other kinds of entertainment during sporting events (for example, Affectiva, Inc. and Emotient, Inc.). This practice is becoming increasingly common in the areas of marketing and advertising to drive decision-making about marketing content (for example, what content works best, when and where to air advertisements).

Beyond these examples, emotionally intelligent systems are likely to impact retail, transportation, communications, governance, and policing. Computers are likely to replace human service professionals in many settings and emotion will play a role in these interactions. This wealth of examples illustrates the impact this technology might have on society. Careful design is therefore critical. Many people currently say they would not trust a machine with important decision-making (that is, money or health management), even when given evidence that machines can perform many tasks, such as data collection, numerical analysis, and planning more effectively than humans.[a] This further reinforces the need for research around systems that engender trust and personalized, emotional intelligence, so they might be considered more trustworthy, empathetic, socially appropriate, and persuasive. However, it will not always be appropriate to make affective systems. For instance, a PA could be considered valued if it performs essential functions, regardless of how natural it is to interact with. Consider human air traffic controllers and the highly analytical and symbolic way that they interact with airline pilots, as one example. Therefore, it is important to consider when it is appropriate to make technology emotion-aware.

As the basis of our position, we turn to a recent article written by Byron Reeves[29] about interactive, online characters that might have several advantages over alternative system instantiations. Reeves claims that since the interactions humans have with media are fundamentally social, it is important for embodied agents to employ social intelligence to be successful. He makes the point that socially intelligent interfaces increase memory and learning and explicitly ground the social interaction. He argues that people implicitly react to these online characters (agents) as social actors. The agents could also increase trust in their interactions, which could be ever more important moving forward, as we incorporate the human-appropriate design aspects.

a https://hbr.org/2017/02/the-rise-of-ai-makes-emotional-intelligence-more-important?utm_campaign=hbr&utm_source=linkedin&utm_medium=social

It has been 20 years since Rosalind Picard published her seminal book on the subject of affective computing.[28] As with other areas of artificial intelligence (AI), however, progress toward her vision has ebbed and flowed. Smaller electronics have transformed wearable computing, enabling signals to be captured and analyzed on comfortable wrist-worn devices. Many consumer-grade smart watches now contain miniaturized physiological sensors that could be used for affect detection. Machine learning, including deep learning, has significantly improved computer-based speech and visual understanding algorithms, such as speech-to-text, facial expression recognition, and scene understanding.

As is the case with other forms of computer technology, there is danger of overhyping the capabilities of affective computing systems. Many of the compelling applications of affective computing have yet to be realized, in part because designing emotionally sentient systems is much more complex than simply sensing affective signals. Understanding and adapting to emotional cues is highly context dependent and relies on tacit knowledge. Compounding this, large, interpersonal variability exists in how people express emotions. Humans also have diverse preferences for how an agent responds to them. Personalization is very important to enable more compelling systems. The most successful affective agent is likely one that can learn about a person's nuanced expressions and responses and adapt to different situations and contexts.

To do all of this, we must develop models of emotion that are amenable to computation. This is challenging, as emotions are difficult define, and the relationship between observed signals and states often requires a many-to-many mapping. Furthermore, human knowledge of emotion is predominantly implicit, defined by unwritten, learned social rules. These rules are also culturally dependent[13] and not universal. Scientists have proposed numerous models of emotion, each with their own strengths and weaknesses. Nevertheless, the choice of defining emotions has significant implications for the design of a sentient system.

In this article, we describe the numerous benefits that emotion-aware

systems can deliver for society. However, it would be negligent to downplay the significant ethical challenges and public concerns that surround the development of this technology. Practitioners should consider our proposals for safeguarding people and maintaining their trust.

To summarize, systems that respond to social and emotional cues are more engaging,[9] build rapport better,[16] and are more trustworthy.[4,17] Unsurprisingly, researchers have also found them to be rated as more human-like and intelligent.[33] However, as with physical appearance, there may not be a linear relationship between an agent's emotional response and how likable it is. Specifically, an "uncanny valley"[21] may exist for emotional expression. Humans are very adept at detecting behaviors that appear to be "off."

Despite the number of challenges associated with building emotionally sentient systems, it remains a highl y motivating goal. For anything other than simple tasks, emotionally intelligent agents have the potential to improve our health and quality of life. For just one example, these systems could help deliver mental health therapy to people struggling to access traditional care,[b] an area of increasing importance.

Here, we address the key design challenges in developing emotionally sentient systems, namely affect sensing, interpretation, and adaptation. While it would not be possible to survey each challenge in depth, we highlight the state-of-the-art research and discuss the most pressing opportunities facing researchers and practitioners.

**We argue that for a PA to truly become valuable to the user, it must be natural to interact with and engaging.**

## Emotion Sensing

Affect sensing and tracking in and of itself has benefits. For example, one could track how the emotions of an individual change over time to understand his emotional triggers.[24] In most cases, however, users would want a system that adapts and responds to affective cues in an intelligent way, such as a computer game designed to change in difficulty based on the players' emotions (for example, Nevermind by Flying Mollusk, Inc.). Furthermore, it is likely that people will desire systems that respond with the appropriate emotion if interaction is required.[27]

Sensing affective states is an integral part of designing emotionally sentient systems. For more than 25 years, computer science methods have been applied to visual, audio, and language data to infer emotion. In many cases, this involves detecting subtle signals amongst high-dimensional data. While verbal and nonverbal cues both contain rich information about a person's emotional state, researchers have found significant improvement in the automated understanding of nonverbal behavior by combining signals from numerous modalities (such as speech, gestures and language).[10] Though the aim of this article is not to survey affect sensing methods, it is important to discuss them, as they influence many practical design considerations. For example, how should designers choose the appropriate types of sensor signals to measure emotions? What is the best way to fuse signals from different modalities? How can you tell if sensor measurements are sufficiently accurate for a given use case? How can a system distinguish between emotional expression and other social cues? In this section, we will discuss the detection of signals. In the next, we discuss how they might be modeled and interpreted.

**Verbal.** Linguistic patterns and word choice can tell us a lot about a user's affective state. Linguistic style matching occurs between people in natural social interactions.[26] Typically, style matching is a sign of rapport or bonding between individuals. People may even alter their speaking style without being consciously aware of it over the passage of time with an interactant.

The LIWC software is a package that enables automatic extraction of linguistic style features[26] by capturing the frequency of use of words from different categories. For example, positive, negative, and functional words turn out to be especially important. Matching a person's linguistic style (for example, through word choice) is perhaps one of the simplest ways an agent can be designed to emotionally bond with a person. For unembodied chatbots, this is one of a small set of techniques that could be used. There are numerous packages available for

b https://woebot.io/

text and speech sentiment analysis, and they are simple to apply. One can design a system that analyzes speech or text for verbal sentiment with a speech-to-text engine. Designers should be aware that these systems might not capture the full complexity of human language. Though many of these systems are trained on large-scale corpora that are available to researchers (for example, Tweets), they may not always generalize well to other domains (like email messages).

**Nonverbal.** Facial expressions, body gestures, and posture are some of the richest sources of affective information. We use automated facial action coding and expression recognition systems to measure these signals in videos. Automated facial action coding can be performed using highly scalable frameworks,[23] allowing analysis of extremely large datasets (for example, millions of individuals). These analyses have revealed observational evidence of cross-cultural and gender differences in emotional expression[23] that for the first time can actually be quantified. Depth-sensing devices like the Kinect sensor significantly advanced pose, gesture, posture and gait analysis making it possible to design systems that used off-the-shelf low-cost hardware. Designers now have access to software SDKs for automated facial and gesture coding that are relatively simple to integrate into other applications. These can even be run on resource-constrained devices enabling mobile applications of facial expression analysis, such as mobile agents that respond to visual cues.

Acknowledging expressions of confusion or frustration from a user's face is one practical way that an agent could make use of facial cues to the benefit of the interaction. Within a known context (that is, an information-seeking task) it is possible to detect these types of negative expressions when they occur. Generally, responses to incorrectly detected affective states will not frustrate the user if they are able to understand the reasoning that the agent used.[24]

The use of a camera or microphone for measuring affective signals (whether in public or private settings) is a particularly sensitive topic, especially if subjects are not aware the sensors are present and active. Designers need to carefully consider how their applications may ultimately influence social norms about where and when video and audio analysis and recording is acceptable.

*Speech prosody.* With the rise of conversational interfaces (such as Cortana and Siri) nonverbal speech signals present an increasingly valuable source of affective information. As with facial coding, there is a strong focus on designing systems that work outside of lab-based settings. Numerous companies have related software development kits (SDK) and application programming interfaces (API) (for example, BeyondVerbal, audEERING, Affectiva) that provide prosodic feature extraction and affect prediction. As with facial expressions, there is likely to be some level of universality in the perception of emotion in speech (a similar set of "basic" emotions) but a great amount of variability will exist across languages and cultures. Many of these "non-basic" states will be of greater relevance in everyday interactions.

*Physiology and brain imaging.* While expressed affective signals are those that are most used in social interactions, physiology plays a significant role in emotional responses. Innervation of the autonomic nervous system has an impact on numerous organs in the body. Computer systems can measure many of these signals in a way that an unaided human could not. Brain activity (for example, electroencephalography (EEG), functional near infra-red (fNIR)), cardiopulmonary parameters (for example, heart and respiration rates and variability) and skin conductance all can be used for measuring aspects of nervous system activity. Although wearable devices have only had partial adoption, there are several compelling approaches for measuring cardiovascular (heart) and pulmonary (breathing) signals using more ubiquitous hardware. The accelerometers and gyroscopes on a cellphone can be used to detect pulse and breathing signals, and almost any webcam is sufficient to remotely measure the same. While people are experienced at applying social controls to their facial expressions and voice tone, they do not have the same control over physiological responses, meaning measurements may feel more intimidating and intrusive to them. One should be cognizant of these concerns in the design of agents, as they are likely to influence how the agents are perceived, from how likable they are, to how trustworthy they are.

**Design challenges for adoption.** Despite the advances in sensing emotions, there remain many challenges in basic objective measurement. Many of

these measurement approaches have not been characterized, or simply fail in natural settings. For example, facial expression recognition may be reliable for videos with simple behaviors and when the face is frontal to the camera, but, in the case of out-of-plane head rotation and co-occurring facial actions, recognition can perform poorly. Physiological sensing approaches are seriously hampered during physical activities. As machine learning and affective computing research advance, objective measurement techniques will improve. In the meantime, practical systems can still be deployed based on automated facial and speech analysis. However, designers need to take these limitations into account.

One challenge with real-world systems that respond to emotions is that expressions of emotion are often very subtle or sparse. This may mean that it is challenging to develop automated detection systems with high recall (that is, the fraction of emotion responses detected) and low false positive (alarm) rates. In social interactions, many nonverbal behaviors (for example, smiling) will be more frequent than when people are alone. Thus, it may be more practical to design systems that respond to both social and emotional cues.

The sparsity and lack of specificity within unimodal cues (that is, a facial expression) are key reasons why multimodal affective computing systems have been found to be consistently better than unimodal ones.[10] In some settings (for example, call center analysis) the availability of visual cues might be limited. In others, various modalities might not available. The most effective systems will be those that leverage the most information, both about the individual and the context she is in.

Large interpersonal variability exists in nonverbal behaviors. Thus, person-specific models can bring many benefits. Longitudinal studies are needed for this type of modeling. To date, such studies have been few and far between. We need to design new mechanisms for incentivizing individuals to interact with a system or to be passively monitored for extended periods of time. Ultimately, the most successful affective computing technology will be able to build personalized models that leverage online learning to update over time.

## Emotion Labels

One of the most significant choices in designing an affective computing system is how to represent or classify emotional states. Emotion theorists have long debated the exact definition of emotion, and many models and taxonomies of emotion have been proposed. Common approaches include discrete, dimensional, and cognitive-appraisal models; other approaches include rational, communicative and anatomic representations of affect.[22]

**Discrete models.** Discrete categorizations of emotion posit that there are "affect" programs that drive a set of core basic emotions and the associated cognitive, physiological, and behavioral processes.[39] There are several categorizations that have been proposed, but by far the most commonly used set is the so-called "basic" list of emotions of anger, fear, sadness, disgust, surprise, and joy. These states can be represented as regions within a dimensional space. In practice, the challenge with discrete models of emotion arises from the state definitions. Even "basic" states do not occur frequently in many situations. Designers must a priori consider which states might be relevant and/or commonly observed in their context.

**Dimensional models.** The most commonly used dimensional model of affect is the circumplex—a circular, two-dimensional space in which points close to one another are highly correlated. Valence (pleasantness) and arousal (activation) are the most frequently selected descriptions used for the two axes of the circumplex, however, the appropriate principal axes are still debated. Another model uses "Positive Affect" (PA) and "Negative Affect" (NA) each with an activation component. Dimensional models are appealing, as they do not confine the output to a specific label but can be interpreted in more continuous ways. For example, in some applications, none of the "basic" emotions labels may apply to an observed emotional response, but that response will still lie somewhere within the dimensional space. Nevertheless, a designer will still need to carefully consider which axes are most appropriate for their use case.

**Appraisal models.** Cognitive-appraisal models consider the influence of emotions on decisions. Specifically, emotions are elicited and differentiated based on a person's evaluation of a stimulus (that is, an event or object). In this case, a person's appraisal of a situation will affect their emotional response to a stimulus. People in different contexts experiencing the same stimulus will not necessarily experience the same emotion.

Appraisal models employ a more formalized approach to context. This is very important, given that only a very small number of behaviors are universally interpretable (and even those theories have been vigorously debated). It is likely that a hybrid dimensional-appraisal model will be the most useful approach.

Although academics have been experimenting with computational models of emotion extensively, there are no commercially available software tools for recognizing emotion (either from verbal or nonverbal modalities) that use an appraisal based model of emotion. Incorporating context and personalization into assessment of the emotional state of an individual is arguably the next big technical and design challenge for commercial software systems that wish to recognize the emotion of a user.

## Emotional Agents

Several articles have been written on the benefits of conversational agents for more naturalistic human-computer interactions.[7,8] This research movement partly came from a belief that traditional WIMP (windows, icons, mouse, and pointer) user interfaces were too difficult to navigate and learn[14] and not natural enough. Here, we focus on the addition of emotional sentience to the agent to explore what additional benefits might be achieved with the addition of intelligent affect sensing and appropriate agent-based responses.

**Dialogue systems.** The first examples of affective agents were dialogue systems. In the 1960s, Eliza was an agent capable of limited natural language understanding[37] that simulated a psychotherapist. Recently, chat systems have become popular and are being used in many forms, from mental health therapy to customer support. The practical application of these dialogue systems has been made possible by advancements in natural language processing (NLP). The barrier to create

bots is now much lower, as illustrated by a 14-year-old boy who created his own homework reminder bot.[c] Many emotional cues are nonverbal, and therefore require an agent to have the ability to express nonverbal emotion. More recent dialogue systems, such as Xiaoiced[d]), leverage text-to-speech technology, allows for a greater range of expression through voice prosody. Yet, the effective synthesis of nonverbal cues is still a very challenging problem. Currently, realistic synthesis of voice tone requires thousands of lines of dialogue to be recorded. Generative machine learning methods may eventually help replace the need for this type of labor-intensive data collection and provide realistic voice synthesis.

**Virtual agents.** While most present day virtual, conversational personal assistants do not rely on emotional recognition or delivery (for example, Siri, Cortana, and others), there has been a large literature examining personality and other emotional components of conversational agents, as well as the social and personal benefits that accrue from their use. Starting with the work by Reeves and Nass[30] in their landmark book, The Media Equation, a communication theory was laid out that suggested humans treated computers and other forms of media as socially as they would another human during conversation. They also claimed that this response from humans is automatic (that is, without conscious effort). Reeves and Nass argued that people respond to what is present in new forms of media, and their *perception* of reality, as opposed to what they know to be true (for example, this is a computer). This allows users to be able to assign a personality to a conversational agent, among other things. Through a series of studies, Reeves, Nass and their colleagues showed that politeness, personality, emotion, social roles, and form all influence how humans treat and respond to all kinds of media, including computer systems. Researchers in the tutoring community[11] have shown that emotionally sentient systems enhance the effectiveness of human-computer interaction, and that the lack of emotional responsiveness can reduce performance. Kraemer[19] has provided ample evidence of the benefits of socio-emotional benefits of pedagogical conversational agents.

A further line of research emphasizes that *embodied* agents offer several advantages over non-embodied dialogue systems. An agent that has a physical presence means that the user can look at it. Cassell[8] has written a lot about this, including how the representation of the agent and its modalities have greater benefits than the early dreams of ubiquitous computing[36] and its goal of embedded (invisible) interaction. Central to her argument is that it is important to realize how humans interact with each other. The human body allows us to "locate" intelligence—both the typical domain knowledge required, but also the social and interactional information we need about conversational parameters such as turn-taking, taking the floor, interruptions, and more. In this vision, then, an embodied social agent who converses with the user requires less navigation and searching than traditional user interfaces (because you know where to find information). Multimodal gestures, such as deixis, eye gaze, speech patterns, and head nods and other, nonverbal gestures are external manifestations of social intelligence which support trustworthiness.[3] For instance, early research has shown that to attain conversation clarify, people rely more on gestural cues when their conversations are noisy.[32] From this perspective, embodied social agents might be a more natural way for people to interact with computation.

So, conversational agents provide a mental model for the user to start with. Well-designed or anthropomorphic features can then help to create a framework of understanding around how to work with these agents. Specifically, conversational agents can provide affordances for available interaction qualities, capabilities, and limits. Our argument is that if designers can tap into users' natural affinity for social interaction with an agent, this will also lead to higher levels of affinity for, and interaction with, that agent. This will eventually lead to trust. If we design agents to not only behave as we expect them to, but also to adhere to social norms and values, then we can amplify trust.[12]

Today, research focusing on virtual assistants, both embodied and not, has achieved positive results: improving users' task performance,[29] establishing trust and likeability in a real estate transaction context,[3] improving naturalness of interactions with appropriate emotions,[29] and in advancing tutorial systems.[11,39] This can largely be attributed to the findings that humans respond to these systems socially, even when they are not. Adding emotional intelligence should only enhance this natural, social response, but more research is needed.

The issue of "social caretaking"[6] (that is, using emotional agents to care for the young, infirmed, or elderly) is a new field under investigation. It has been found that proactive, affective agents can help elderly users feel more comfortable with the technology, and can even ease loneliness to some degree.[31] Also, work by Lucas et al.[20] shows the real promise in using conversational agents in clinical interviews. They obtained more honest responses from patients with increased willingness to disclose, since the patients felt more comfortable talking with an agent than a human in certain circumstances. While researchers in this line of work have shown the benefits of agents, they have also pointed out that humans will engage in racism, lie, feel envy, and more toward emotional agents. Thus, this is a key area to continue exploring, as we get better at designing emotional systems.

However, there have been concerns raised that the appearance of these embodied emotional agents lack naturalness, especially with nonverbal gestures and cues such as inaccurate eye gaze or emotional facial gestures.[2] If humans begin to mimic or model their interactions with an agent who doesn't emote appropriately, it could result in negative emotional learning. This issue is of most concern in the social caretaking scenarios mentioned above, and especially with children, who model behavior through social learning.[1] While the affective modeling community is making great strides in creating more natural, human-like embodied agents with real, human-like communication patterns,[39] we have a lot to do to allay these concerns.

**Robots.** Physical systems have advantages over virtual agents. The most obvious is that robotic systems can perform physical actions and tasks in the real world. They can put an arm around

c http://www.christopherbot.co/

d https://thestack.com/world/2016/02/05/microsoft-xiaoice-turing-test-china/

a person to comfort them or move an object or make a meal. Again, in this domain, research is revealing the benefits of robots that express affect appropriate to each situation, such as asking for something politely or apologizing after making a mistake. Researchers have found that robots showing human-like expressions and positive politeness are more able to get humans to assist them and that robots that show sorrow or sadness after making a mistake are viewed as more intimate, especially if the users thought the robots were acting autonomously.[17] Hammer et al.[18] report on several studies that look at the acceptability of social robots by older adults. They found that attributes like appearance, intellectuality, friendliness, and kind-heartedness are important for acceptability. In addition, robot companions may be viewed more positively if they emulate situationally appropriate social behavior.

Another well-known study also looked at users' reactions to interactions with a robot after good or mistaken task performance and whether or not the robot responded emotionally.[17] These researchers were interested in the question surrounding unexpected behaviors from robots during collaborative tasks, which are extremely likely to occur. There is currently very little research on the topic. These researchers thought an affective interaction might be more useful and trust-enabling than a more efficient, less human-like interaction. What they found was a humanoid robot that expresses emotions, for instance apologizing via speech and nonverbal gestures, is much preferred over one without these skills, despite taking more time on the task and making errors. They also found the robot that exhibited more human-like, emotional signals might make humans more likely to feel empathetic toward the robot, and not want to hurt its feelings. Most importantly, the humans trusted these robots more because of their increased transparency and feedback in communication and emotional expression. These findings suggest that robots that express human-like, polite, emotional signals can significantly mitigate dissatisfaction when errors or other problems occur during human-agent interaction. These findings could also result in good design guidelines for designers of human-robot or other kinds of human-agent conversational systems. These systems will always suffer from imperfect reliability and a superior design principle involves exposing transparency about the outcome and involving the human in the reparation. As the authors point out, however, juxtaposing reliability with expressiveness is challenging and the design of an error-free system is unlikely in the near term.

**These systems will always suffer from imperfect reliability and a superior design principle involves exposing transparency about the outcome and involving the human in the reparation.**

And of course, there is concern about the uncanny valley, as it has been shown that if robots look too human-like, but do not match social expectations in terms of behavior, then people do not like and might distrust these systems even more. Anti-robot sentiment, in addition, could be a real concern. People may feel threatened by the proliferation of robots and the appearance that robots will not care for humans, act morally or ethically.

### Future Affective Systems

The deployment of intelligent agents is widespread on mobile devices and desktops. However, most agents that have been designed with some emotion sentience have been limited to constrained experimental settings. While "cognitive" agents can often perform effectively with NLP alone, emotionally sentient agents require multimodal sensing capabilities and the ability to express emotion in more complex ways, which has been very challenging to achieve in real-world settings. However, given the review here, it is likely the next frontier on which these assistants/agents compete with one another will be their ability to emotionally connect with their users.

Social robotics that have basic facial expression recognition (for example, Pepper, Softbank Inc.) are now on the market. These devices are likely to elicit a richer set of emotions than the typical interaction with a cognitive agent designed for information retrieval. As such, they present the exciting potential for large-scale, in-situ experimentation and user experience testing. Large-scale collection and analysis of affective data is important for improving affective computing systems, and deployment of systems in everyday contexts is one way to achieve this, with the obvious caveats raised earlier.

Robots can express rich emotion, in

addition to having customized hardware for sensing affective signals. Leonardo[5] is an example of a robot with a face capable of near human-level expression. Commercially available robots, such as Cozmo (by Anki, Inc.), have engines for expressing limited physical emotional behaviors. However, robotics such as these are unlikely to be ubiquitous in the near-term. The most common emotional agents are still likely to be virtual. These agents need not have human appearance; abstracted representations of characters can still communicate significant amounts of emotional information. We can return to perhaps the most famous robot of all—R2-D2—that was scripted to successfully convey many emotions through colors and sounds. Agents such as Cortana could use similar abstractions to both convey emotions and elicit emotion from their users; physical motion is not a prerequisite for complex emotional expression.

It is also important for designers to understand that learning purely from human-human behavior may not always be the most effective approach.[35] Considering how to present and sense information is important when a user is trying to complete tasks that already require considerable cognitive processing.

Embodied social agents can help express and regulate emotion, which is important in every social interaction. We know that emotional intelligence is a key factor in intellect and can strongly influence behavior. Per Reeves,[29] research shows that negative experiences with technology are much more strongly remembered and actionable than are positive ones, so automated systems need to consider negative interactions in design, as ignoring these negative incidents could lead to the same bad feelings, or worse, rejection of an automated system. Embodied social agents are a preferred way to deal with these kinds of experiences. Facial expressions, for example, can signal what responses are appropriate, or when more information is needed. This can be much faster than just using words or text alone. Likewise, intelligent social agents can be used to display important social and cultural manners, whose influence should not be ignored in design as well. Reeves' overall point, much like that of Cassell's,[3,9] is that embodied, social agents that respect human-to-human interaction protocols, simply can make user interfaces easier to use, if designed appropriately.

In the near-term machines are unlikely to understand all of the complex social norms that humans typically follow or detect the emotional states of people with high precision and recall. Therefore, agents will, on occasion, exhibit socially inappropriate behavior. Ideally, an intelligent system should be designed so that it can learn from these mistakes, or at the very least apologize when a mistake is detected. In a week-long study, we found that people were generally delighted when the computer accurately reflected their mood and quite forgiving when it did not.[24] However, for a commercial system that will be used for more than two weeks, a user's patience could be tested by a system which regularly makes mistakes and cannot be corrected or learn online.

Designing systems that can measure, often passively, and log affective signals presents ethical challenges. As with any technology, there is the possibility that it will be abused. Much of the hardware used for sensing affective signals is small and ubiquitous (for example, microphones or webcams). Even measurement of physiological signals can be performed using these devices and does not require contact with the body. Thus, people may not be aware that an agent is measuring and responding to their emotional state.

As described in Becker et al.,[2] our ability to render emotional expressiveness in agents is extremely limited today, though this is improving quickly. Still, it should be cautioned that embodied agents and robots will never experience the physiological reactions nor the actual emotions that they project (for example, a racing heart or relaxation). The question then becomes one of how humans react to this limited display of emotionality and our obvious understanding that these agents are not human. Much more experimentation must be done to identify the uncanny valley and find design sweet spots, where more natural expression abilities and ease of use don't crossover into negative experiences.

There is a danger that a person could be manipulated by agents that can interpret their emotional state. People tend to trust agents that appear more attractive, for example, even when they are not reliable.[38] Deception of this kind must be avoided. If we are to be interacting with computer agents more and more, there is a likelihood that we will change our behavior to mimic that of the system, much as humans do.[26] Other evidence supports this idea, such as data showing that people are changing how they think as a result of using Internet search engines. Specifically, children, who have extensive interactions with an agent that cannot accu-

rately mimic human emotional cues and understanding, may end up "imprinting" these social agents' behaviors and styles of interaction. Another undesirable outcome would be that children grow-up treating agents rudely and that these behaviors leak into human interactions. Designers should study and consider how to minimize the chance of these negative scenarios.

Finally, an affective agent may raise the users' expectations of competence or common sense that the system may not possess. In circumstances where this could lead to frustration, or other negative outcomes, it might not be appropriate to make a system respond to affective signals.

## Conclusion

While research and development of emotionally sentient computer systems is already 50 years old, only recently have these systems been adopted for real-world applications. Agents that sense, interpret, and adapt to human emotions are impacting healthcare, education, media and communications, entertainment, and transportation. However, there remain fundamental questions about the design principles that should govern such systems. From the types of signals that are measured, to the model of emotions that is employed, to the types of tasks they perform and the emotions they express, there are fundamental research questions that still need to be answered.

Agents can take many forms, from dialogue systems to physically expressive humanoid robots. While intelligent agents are widespread on mobile devices and desktops, those that have been designed with emotional sentience have been limited to constrained experimental settings. However, one could argue the deployment of emotionally sentient systems is at a tipping point. The next major advancement in development will be spurred by large-scale and longitudinal testing of these systems in real-world settings. This will in part be made possible by the increasing adoption of intelligent assistants (for example, Apple's Siri, Microsoft's Cortana, Amazon's Alexa, or Google Assistant) and in part by the availability of social robots.

We have highlighted current design challenges that are limiting adoption of these systems, including, how to account for large interpersonal variability, sparsity, many-to-many mappings between behaviors and emotions, and how to create a system that avoids social faux pas. There are ethical issues raised by emotionally sentient systems and this needs very serious, careful design consideration. C


**References**

1. Bandura, A. Human agency in social cognitive theory. *American Psychologist 44*, 9 (1989), 1175.
2. Becker, B. Social robots-emotional agents: Some remarks on naturalizing man-machine interaction. Intern. *Review of Info. Ethics 6*,12 (2006), 37–45.
3. Bickmore, T. and Cassell, J. How about this weather? Social dialogue with embodied conversational agents. In *Proceedings of the AAAI Fall Symposium on Socially*, 2000.
4. Bickmore, T. and Cassell, J. Relational agents: a model and implementation of building user trust. In *Proceedings of 2001 SIGCHI Conf. Human Factors in Computing Systems*. ACM, 396–403.
5. Breazeal, C., Buchsbaum, D., Gray, J., and Gatenby, D. Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots. *Artificial Life 11*, 1–2 (2005), 31–62.
6. Breazeal, C. Designing sociable machines. *Socially Intelligent Agents*. Kluwer Academic Publishers, Boston, MA, 2002, 149–156.
7. Breese, J. and Ball, G. Modeling emotional state and personality for conversational agents. Rapport technique MSR-TR-98-41. Microsoft Research.
8. Cassell, J. Embodied conversational agents: representation and intelligence in user interfaces. *AI Magazine 22*, 4 (2001).
9. Cassell, J. and Thorisson, K.R. The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence 13*, 4–5 (1999), 519–538.
10. D'Mello, S. and Kory, J. Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies. In *Proceedings of the 14th ACM Conf. Intern. Multimodal Interaction*, 2012, 31–38.
11. D'Mello, S., Picard, R.W., and Graesser, A. Toward an affect-sensitive AutoTutor. *IEEE Intelligent Systems 22*, 4 (2007).
12. Drury, J.L., Scholtz, J., and Yanco, H.A. Awareness in human-robot interactions. *IEEE Intern. Conf. Systems, Man and Cybernetics*, 2003, 912–918.
13. Elfenbein, H. and Ambady, N. On the universality and cultural specificity of emotion recognition: A meta-analysis. *Psychological Bulletin 128*, 2 (2002), 203–235.
14. Flanagan, J., Huang, T., Jones, P., and Kasif, S. Human-Centered Systems: Information, Interactivity and Intelligence. NSF Report, 1997.
15. Gratch, J. and Marsella, S. Tears and fears: Modeling emotions and emotional behaviors in synthetic agents. In *Proceedings of the 5th Intern. Conf. Autonomous Agents*, 2001, 278–285. IEEE.
16. Gratch, J., Wang, N., Gerten, J., Fast, E., and Duffy, R. Creating rapport with virtual agents. In *Proceedings of the Intern. Conf. Intelligent Virtual Agents*, 2007, 125–138. Springer.
17. Hamacher, A. and Bianchi-Berthouze, N. Believing in BERT: Using expressive communication to enhance trust and counteract operational error in physical human-robot interaction. *Robot and Human Interactive Communication*, 2016, 493–500.
18. Hammer, S., Lugrin, B., Bogomolov, S., and Janowski, K. Investigating politeness strategies and their persuasiveness for a robotic elderly assistant. *PERSUASIVE*, (2016), 315–326.
19. Krämer, N. and Bente, G. Personalizing e-learning. The social effects of pedagogical agents. *Educational Psychology Rev. 22*, 1 (2010), 71–87.
20. Lucas, G.M., Gratch, J., King, A., and Morency, L.P. It's only a computer: virtual humans increase willingness to disclose. *Computers in Human Behavior 37* (2014), 94–100.
21. Mori, M. The uncanny valley. *Energy 7*, 4 (1970), 33–35.
22. Marsella, S., Gratch, J., and Petta, P. Computational models of emotion. *Blueprint for Affective Computing—A Sourcebook and Manual*. Oxford University Press, 2010, 21–46.
23. McDuff, D., Girard, J.M., and el Kaliouby, R. Large-scale observational evidence of cross-ultural differences in facial behavior. *J. Nonverbal Behavior* 1573 (2016), 1–19. Springer.
24. McDuff, D., Karlson, A., Kapoor, A., and Roseway, A. AffectAura: An intelligent system for emotional memory. In *Proceedings of the SIGCHI Conf. Human Factors in Computing Systems*, 2012, 849–858. ACM.
25. Miner, A., Chow, A., Adler, S., Zaitsev, I., and Tero, P. Conversational agents and mental health: Theory-informed assessment of language and affect. In *Proceedings of the 4th Intern. Conf. Human Agent Interaction*, 2016, 123–130. ACM.
26. Niederhoffer, K.G. and Pennebaker, J.W. Linguistic style matching in social interaction. *J. Language and Social 21*, 4 (2002), 337–360.
27. Paredes, P., Giald-Bachrach, R., Czerwinski, M., Roseway, A., Rowan, K., and Hernandez, J. PopTherapy: Coping with stress through pop-culture. In *Proceedings of the 8th Intern. Conf. Pervasive Computing Technologies for Healthcare*, 2014, 109–117.
28. Picard, R.W. *Affective Computing*. MIT Press, 1995, 1–16.
29. Reeves, B. The benefits of interactive, online characters. *The Madison Avenue J.*, 2010.
30. Reeves, B. and Nass, C. *The Media Equation: How people treat computers, television, and new media like real people and places*. CSLI Publications and Cambridge University Press, 1996.
31. Ring, L., Barry, B., and Totzke, K. Addressing loneliness and isolation in older adults: Proactive affective agents provide better support. In *Proceedings of the 2013 Humaine Assoc. Conf. Affective Computing and Intelligent Interaction*. IEEE, 61–66.
32. Rogers, W.T. The contribution of kinesic illustrators toward the comprehension of verbal behavior within utterances. *Human Communication Research 5*, 1 (1978), 54–62.
33. Shamekhi, A., Czerwinski, M., Mark, G., and Novotny, M. An exploratory study toward the preferred conversational style for compatible virtual agents. In *Proceedings of the Intern. Conf. Intelligent Virtual Agents*. Springer, 2016, 40–50.
34. Srinivasan, V. and Takayama, L. Help me please: Robot politeness strategies for soliciting help from humans. In *Proceedings of the ACM SIGCHI Conf. Human Factors in Computing Systems*, 2016, 494–4955.
35. Shneiderman, B. The limits of speech recognition. *Commun. ACM 43*, 9 (Sept. 2000), 63–65.
36. Walters, M.L., Syrdal, D.S., and Dautenhahn, K. Avoiding the uncanny valley: Robot appearance, personality and consistency of behavior in an attention-seeking home scenario for a robot companion. *Autonomous Robots 24*, 2 (2008), 159–178.
37. Weiser, M. The computer for the 21st century. *Scientific American 265*, 3 (1991), 94–104.
38. Weizenbaum, J. ELIZA—A computer program for the study of natural language communication between man and machine. *Commun. ACM 9*, 1 (Jan. 1966), 36–45.
39. Yuksel, B.F., Collisson, P, and Czerwinski, M. Brains or beauty: How to engender trust in user-agent interactions. *ACM Trans. Internet Techn. 17*, 1 (2017). ACM, 2.
40. Zhao, R., Sinha, T., Black, A.W., and Cassell, J. Socially-aware virtual agents: Automatically assessing dyadic rapport from temporal patterns of behavior. In *Proceedings of the International Conference on Intelligent Virtual Agent.*, Springer, 2016, 218–233.



**Daniel McDuff** (damcduff@microsoft.com) is a researcher at Microsoft Research, Redmond, WA, USA.

**Mary Czerwinski** (marycz@microsoft.com) is a research manager of the Visualization and Interaction (VIBE) Research Group at Microsoft Research, Redmond, WA, USA.




Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/designing-emotionally-sentient-agents

DOI:10.1145/3208071

**A promising, useful tool for future programming development environments.**

BY RAJEEV ALUR, RISHABH SINGH, DANA FISMAN, AND ARMANDO SOLAR-LEZAMA

# Search-based Program Synthesis

Writing programs that are both correct and efficient is challenging. A potential solution lies in *program synthesis* aimed at automatic derivation of an executable implementation (the "how") from a high-level logical specification of the desired input-to-output behavior (the "what"). A mature synthesis technology can have a transformative impact on programmer productivity by liberating the programmer from low-level coding details. For instance, for the classical computational problem of sorting a list of numbers, the programmer has to simply specify that given an input array $A$ of $n$ numbers, compute an output array $B$ consisting of exactly the same numbers as $A$ such that $B[i] \leq B[i+1]$ for $1 \leq i < n$, leaving it to the synthesizer to figure out the sequence of steps needed for the desired computation.

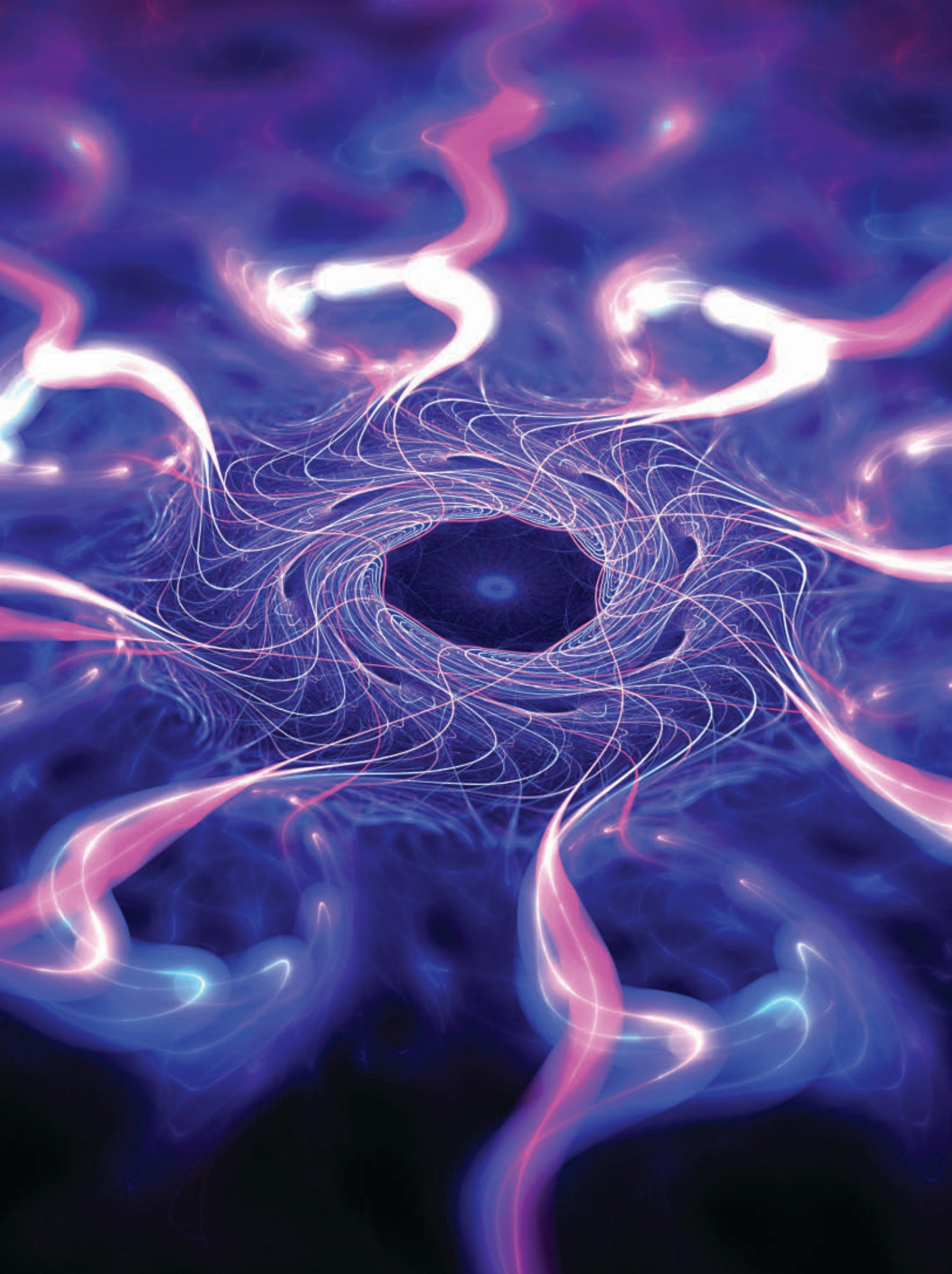
Traditionally, program synthesis is formalized as a problem in deductive theorem proving:[17] A program is derived from the constructive proof of the theorem that states that for all inputs, there exists an output, such that the desired correctness specification holds. Building automated and scalable tools to solve this problem has proved to be difficult. A recent alternative to formalizing synthesis allows the programmer to supplement the logical specification with a syntactic template that constrains the space of allowed implementations and the solution strategies focus on search algorithms for efficiently exploring this space. The resulting *search-based program synthesis* paradigm is emerging as an enabling technology for both designing more intuitive programming notations and aggressive program optimizations.

As an instance of making programming easier and accessible to end users, consider the *programming-by-examples* (PBE) systems that allow a user to specify the desired functionality using representative input-to-output examples. Such a programming environment is feasible in domain-specific applications such as data manipulation as illustrated by the success of the FlashFill feature in Microsoft Excel spreadsheet software.[10, 11] This feature can automatically fill in a column in the spreadsheet by examining a few examples provided by the user. The underlying computational problem is search-based synthesis, namely, finding a program that is consistent with all the user-provided examples and fits the syntactic template of the native language.

## » key insights

- **Syntex-guided synthesis formalizes the computational problem of searching for a program expression that meets both syntactic and logical constraints.**
- **A wide variety of problems, such as programming by examples, program superoptimization, and program repair, naturally map to syntax-guided synthesis.**
- **Standardization, benchmark collection, and solver competition have led to significant advances in solution strategies and new applications.**

IMAGE FROM SHUTTERSTOCK.COM



A traditional optimizing compiler transforms the input program by applying a sequence of transformations where each transformation makes a local change to the program that is guaranteed to preserve semantic equivalence. An alternative, based on search-based synthesis, is to explore the space of syntactically correct programs for a program that is semantically equivalent to the input program and meets desired performance criteria (for example, uses an expensive operation only a limited number of times). This approach offers the possibility of more aggressive optimization—sometimes called *superoptimization*, as it can lead to a resulting program that is structurally quite dissimilar to the original one.[18,26]

Since the number of syntactically correct programs grows exponentially with the size, searching through the space of programs is computationally intractable. Our attempt to tackle this seemingly hopeless research challenge is rooted in two lessons learned from the progress on two analogous, computationally intractable, problems in formal analysis: *model checking* that requires exploration of reachable states of finite-state models of protocols[4] and *constraint solving* to find a satisfying assignment to variables in a logical formula with Boolean connectives.[5,16] First, a sustained focus on battling the computational bottlenecks via algorithmic innovations, data structures, and performance tuning can result in impressive advances in tools. Second, even when the tools have scalability limits, they can still prove invaluable in practice when applied to carefully chosen real-world problems.

While search-based synthesis is the computational problem at the core of a number of synthesis projects dating back to the system SKETCH for program completion,[28,29] the precise formulation we focus on is called *syntax-guided synthesis* (SyGuS):[2] Given a set *Exp* of expressions specified by a context-free grammar that captures the set of candidate implementations of an unknown function *f*, and a logical formula *Spec* that captures the desired functionality of *f*, find an expression *e* in *Exp* such that replacing *f* by *e* in *Spec* results in a valid formula. The input format for this problem has been standardized, hundreds of benchmarks from different application domains have been collected, and a competition of solvers has been held annually starting 2014 (see www.sygus.org). This community effort has led to innovations in both computational techniques for solvers and practical applications.

In this article, we introduce the SyGuS problem using four applications: synthesis from logical specifications, programming by examples, program transformation, and automatic inference of program invariants. Next, we discuss a generic architecture for solving the SyGuS problem using the iterative counterexample-guided inductive synthesis (CEGIS) strategy[29] that combines a search strategy with a verification oracle. As an instance of the learning algorithm, we explain the enumerative technique that generates the candidate expressions of increasing size relying on the input examples for pruning.[32] We then describe the standardized input format, the benchmarks, and the annual competition of solvers. This infrastructure effort was supported by NSF Expeditions in Computing project ExCAPE (Expeditions in Computer-Augmented Program Engineering) focused on advancing tools and applications of program synthesis. We close by examining the resulting progress. In particular, we explain how the best performing solver in the 2017 competition integrates decision trees in the enumerative search algorithm to boost its performance,[3] and discuss a new application of SyGuS to automatically make cryptographic circuits resilient to timing attacks.[7]

It should be noted that search-based program synthesis is an active area of research with tools and applications beyond the specific formalization we focus on. While space does not permit a detailed discussion of the related work, let us mention a few relevant trends: type-based approaches to code completion,[14,21,22] use of statistical models learnt from code repositories for program synthesis,[24] and search-based program repair.[15,19]

## Syntax-Guided Synthesis

The SyGuS problem is to find a function *f* that meets the specified *syntactic* and *semantic* constraints.[2] The syntactic constraint is given as a grammar deriving a set *Exp* of expressions that captures the candidate implementations of *f*. The semantic constraint is a logical formula *Spec* that captures the desired functionality of *f*. We introduce the problem using a series of illustrative examples from different applications.

**Synthesis from logical specifications.** A logical specification of a function describes *what* needs to be computed. As a simple example, consider the following specification $Spec_1$ of a function *f* that takes two input arguments *x* and *y* of type *int* and returns an integer value that is the maximum of the input arguments:

$$Spec_1 : (f(x,y) \geq x) \wedge (f(x,y) \geq y) \wedge (f(x,y) \in \{x,y\}).$$

Finding a function *f* satisfying this logical specification can be viewed as establishing the truth of the quantified formula: $\exists f. \forall x, y. Spec_1$. A constructive proof of this formula can reveal an implementation of *f*.[17] Since automatic proofs in a logic that supports quantification over functions remains a challenge, the *syntax-guided* approach we advocate asks the user to specify additional structural constraint on the set of expressions that can be used as possible implementations of *f*. For example, the following grammar specifies the set $Exp_1$ of all linear expressions over input arguments *x* and *y* with positive coefficients:

$$Exp_1 : E := x \mid y \mid 0 \mid 1 \mid E + E.$$

Now the computational problem is to systematically search through the set $Exp_1$ of expressions to find an expression *e* such that the formula obtained by substituting *e* for $f(x, y)$ in $Spec_1$ is valid. Convince yourself that there is no solution in this case: no linear expression over two integers can correspond to the maximum of the two.

Since no linear expression satisfies the specification, we can enrich the set of candidate implementations by allowing conditionals. The following

grammar specifies this set $Exp_2$:

$$Exp_2: T := x \mid y \mid 0 \mid 1 \mid T + T \mid \texttt{ITE}(C,T,T)$$
$$C := (T \leq T) \mid \neg C \mid (C \wedge C).$$

Here the nonterminal T generates linear expressions, the nonterminal C generates tests used in conditionals, and for a test $t$ and expressions $e_1$ and $e_2$, ITE($t$, $e_1$, $e_2$) stands for *if t then* $e_1$ *else* $e_2$. Now $f(x, y)$ = ITE($(x \leq y)$, $y$, $x$) satisfies the logical specification $Spec_1$ and also belongs to the set $Exp_2$. Observe that this expression does not involve addition of terms and thus can also be generated by the following simpler grammar that specifies the set $Exp_3$ of expressions:

$$Exp_3: T := x \mid y \mid 0 \mid 1 \mid \texttt{ITE}(C,T,T)$$
$$C := (T \leq T) \mid \neg C \mid (C \wedge C).$$

Now suppose we change the logical requirement of the desired function $f$ from $Spec_1$ to $Spec_2$:

$$Spec_2: (f(x,y) > x) \wedge (f(x,y) > y).$$

Observe that this is an under-specification since multiple functions can satisfy this logical constraint. If we choose the set of expressions to be $Exp_2$, a possible solution is $f(x,y) = \texttt{ITE}((x \leq y), y, x) + 1$. However, this solution will no longer work if the set of expressions is $Exp_3$. This ability to change the specification of the desired function by revising either the logical formula or the set of expressions offers a significant convenience in encoding synthesis problems.

**Programming by examples.** An appealing application of synthesis is to learn a program from representative input-to-output examples. The Flash-Fill feature in Microsoft Excel is a recent success of such a programming methodology in practice.[10,11] It allows Excel users to perform string transformations using a small number of input-to-output examples. For example, consider the task of transforming names from one format to another as shown in Table 1. Formally, the semantic constraint on the desired function $f$ from strings to strings is given by the formula with a conjunct for each of the rows, where the conjunct for the first row is of the form $f$(Nancy FreeHafer) = FreeHafer, N.

**String transformation by examples.**

| Input | Output |
|---|---|
| Nancy FreeHafer | FreeHafer, N. |
| Andrew Cencini | Cencini, A. |
| Jan Kotas | Kotas, J. |

The set of string transformations supported by the domain specific language of Excel can be specified by the grammar below (simplified for exposition):

$$E := \texttt{Concat}(E,E) \mid \texttt{SubStr}(E,I,I) \mid \text{“.”} \mid \text{“,”} \mid \text{“ ”}$$
$$I := 0 \mid 1 \mid 2 \mid I + I \mid I - I \mid \texttt{Len}(E) \mid \texttt{IndexOf}(E,E,I).$$

In this grammar, the nonterminal $E$ generates string transformations, and the nonterminal $I$ generates integer-valued index expressions. The Concat($s_1$, $s_2$) function returns the concatenation of the strings $s_1$ and $s_2$, SubStr($s$, $i_1$, $i_2$) returns the substring of the string $s$ between the integer positions $i_1$ and $i_2$, Len($s$) returns the length of the string $s$, and IndexOf($s_1$, $s_2$, $i$) returns the index of the $i^{\text{th}}$ occurrence of the string $s_2$ in the string $s_1$.

A possible solution to the SyGuS problem is: Concat($s_1$, “,”, $s_2$, “.”), where $s_1$ is the expression SubStr($s$, IndexOf($s$, “ ”, 1) + 1, Len($s$)) and $s_2$ is SubStr($s$, 0, 1). This program concatenates the following four strings: the substring in the input string starting after the first whitespace; constant string “,”; the first character; and constant string “.”.

**Program optimization.** In automatic program optimization, we are given an original program $f$, and we want to find another program $g$ such that the program $g$ is functionally equivalent to $f$ and satisfies specified syntactic constraints so it has better performance compared to $f$. The syntactic constraint can be used to rule out, or restrict, the use of certain operations deemed expensive.

As an example, consider the problem of computing the average of two unsigned integer input numbers $x$ and $y$ represented as bitvectors. The obvious expression $(x + y)/2$ is not a correct implementation since the intermediate result $(x + y)$ can cause an overflow error. If the input numbers are bitvectors of length 32, an alternative correct formulation can first extend the given numbers to 64-bits to make sure that no overflow error will be introduced when they are summed up together, then divide by 2, and finally convert the result back to 32-bits. This is specified by the function:

$$f(x,y) = \text{BV}_{32}((\text{BV}_{64}(x) + \text{BV}_{64}(y))/2),$$

where the operator $\text{BV}_{64}$ converts a 32-bitvector to a 64-bitvector by concatenating 32 zeros to its left, and $\text{BV}_{32}$ converts a 64-bitvector to a 32-bitvector by taking the 32 rightmost bits.

Since the result does not require more than 32-bits, we want to know if there exists an equivalent solution that works without using an extension to 64-bits. We can pose this as a SyGuS question: does there exist an expression that is equivalent to $f(x, y)$ and is generated by the grammar:

$$E := x \mid y \mid E + E \mid E \,\&\, E \mid E \mid\mid E \mid E \,\hat{}\, E \mid \tilde{}\, E \mid E \gg N \mid E \ll N$$

where $+$ is addition, &, ||, $\hat{}$ are bitwise and, or, and xor, $\ll$ and $\gg$ are shift left and shift right, $N$ is the set of integer constants between 0 and 31. Note that the grammar explicitly rules out the use of bitvector conversion operators $\text{BV}_{64}$ and $\text{BV}_{32}$ used in the original program $f$. A correct solution to the synthesis problem is the program

$$g(x,y) = (x \,\&\, y) + (x \,\hat{}\, y) \gg 1.$$

**Template-based invariant synthesis.** To verify that a program satisfies its correctness specification, one needs to identify *loop invariants*—conditions over program variables that are preserved by an execution of the loop. As a simple example consider the following program, where $i$, $j$, $m$, and $n$ are integers:

```
1. i = m;
2. j = n;
3. while (i > 0) {
4.       i = i - 1;
5.       j = j + 1;
6. }.
```

**Figure 1. Architecture of SyGuS solver.**

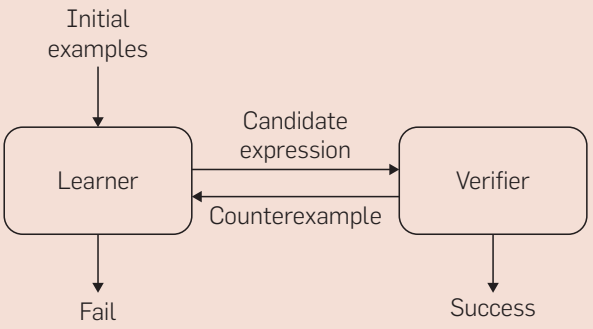



**Figure 2. Illustrative execution of CEGIS.**

| Iteration | Candidate expression | Counterexample |
|---|---|---|
| 1 | $x$ | $(x=0, y=1)$ |
| 2 | $y$ | $(x=1, y=0)$ |
| 3 | 1 | $(x=0, y=0)$ |
| 4 | $x+y$ | $(x=1, y=1)$ |
| 5 | $\texttt{ITE}((x \le y), y, x)$ | Success |

We want to prove that if $m$ is a non-negative integer then when the program terminates $j$ equals $m+n$; that is, assuming the pre-condition $m \ge 0$, the post-condition $j=m+n$ holds. To apply the standard verification technology, we need to first find a Boolean predicate $f$ over the variables $i, j, m$, and $n$, that must hold every time the program control is at line number 3. The desired predicate $f(i, j, m, n)$ should satisfy the following three logical requirements: (1) assuming the pre-condition, the first time the program control reaches the while loop, the desired predicate $f$ holds:

$$Pre\colon (m \ge 0) \rightarrow f(m, n, m, n),$$

(2) assuming that $f(i, j, m, n)$ holds, and the program enters the while loop (that is, the test $i > 0$ is satisfied), after executing the body of the loop once, the condition $f$ continues to hold for the updated variables:

$$Induct\colon (f(i,j,m,n) \wedge i > 0) \rightarrow f(i-1,\ j+1, m, n),$$

and (3) assuming that $f(i, j, m, n)$ holds, and the program exits the loop, the post-condition of the program holds:

$$Post\colon (f(i,j,m,n) \wedge \neg(i > 0)) \rightarrow j = m+n.$$

A predicate $f$ that satisfies all these conditions is an inductive invariant that is strong enough to prove the correctness of the program. A modern proof assistant for program verification asks a user to annotate a program with such loop invariants, and then automatically checks whether all these conditions are satisfied.

The more ambitious task of automatically synthesizing loop invariants that satisfy the desired conditions can be formalized as a SyGuS problem.[30] In the above example, the function $f$ to be synthesized takes four integer arguments and returns a Boolean value. The logical specification is the conjunction $Pre \wedge Induct \wedge Post$. As a syntactic specification for the set of potential candidates for invariants, we choose expressions that are conjunctions of linear inequalities over program variables. This set is expressed by the grammar:

$$C := C \wedge C \mid (T \le T);$$
$$T := x \mid y \mid m \mid n \mid 0 \mid 1 \mid T + T.$$

The following expression $f(i, j, m, n)$ then is a solution satisfying both syntactic and semantic constraints:

$$(i + j \le m + n) \wedge (m + n \le i + j) \wedge (0 \le i).$$

## Solving Sygus

Given a set *Exp* of expressions specified using a grammar, and a logical formula *Spec* that constrains the desired function $f$, the SyGuS problem is to find an expression $e$ in *Exp* such that the formula $Spec[f/e]$ obtained by replacing $f$ with $e$ in *Spec* is valid or report failure if no such expression exists. This search involves an alternation of quantifiers: *there exists* an expression $e$ in *Exp* such that *for all* inputs, $Spec[f/e]$ holds. The architecture underlying current solvers involves a cooperation between a *learning* module that searches for a candidate expression and a *verification* oracle that checks its validity as explained next.

**Counterexamples and inductive synthesis.** The architecture of a typical SyGuS solver is shown in Figure 1 (see Seshia[27] for alternative querying models). The set *Examples* contains *interesting* inputs that the learner uses to guide its search. This set can initially be empty. The learner is tasked with finding an expression $e$ in *Exp* such that $Spec[f/e]$ is satisfied at least for the inputs in *Examples*. If the learner fails in this task, then there is no solution to the synthesis problem. Otherwise, the candidate expression $e$ produced by the learner is given to the verifier that checks if $Spec[f/e]$ holds for all inputs. If so, the current expression $e$ is the desired answer to the synthesis problem. If not, the verifier produces a *counterexample*, that is, an input for which the specification does not hold, and this input now is added to the set *Examples* to reiterate the learning phase. The learning phase is an instance of the so-called *inductive synthesis* as the learner is attempting to generalize based on the current set *Examples* of inputs it considers significant. Since the inputs added to this set are counterexamples produced by the verifier, the overall solution strategy is called CEGIS.[28,29]

For illustrating this strategy, let us show a plausible sequence of iterations using the logical specification $Spec_1$ and the set $Exp_2$ of linear expressions with conditionals noted previously. Initially the set *Examples* is empty, and as a result, the learner has no constraints and can return any expression it wants. Suppose it

returns $x$. Now the verifier checks if setting $f(x, y) = x$ satisfies the logical specification; that is, it checks the validity of the formula $(x \geq x) \wedge (x \geq y) \wedge (x \in \{x, y\})$. This formula does not hold for all values of $x$ and $y$, and the verifier returns one such counterexample, say, ($x = 0$, $y = 1$). This input is added to the set *Examples*, and the learner now needs to find a solution for $f$ that satisfies the specification at least for this input. The learner can possibly return the expression $y$, and when the verifier checks the validity of this answer it returns ($x = 1$, $y = 0$) as a counterexample. Figure 2 shows the expressions learnt and the corresponding counterexamples produced by the verifier in successive iterations. For instance, in iteration 4, the learner attempts to find a candidate solution that satisfies the specification for the inputs (0, 1), (1, 0), and (0, 0), and $f(x, y) = x + y$ is indeed such a plausible answer. The verifier then checks the validity of $(x + y \geq x) \wedge (x + y \geq y) \wedge (x + y \in \{x, y\})$ and returns ($x = 1$, $y = 1$) as a counterexample. In the subsequent iteration, when the learner attempts to find a solution that fits all the four inputs currently in *Examples*, the shortest expression possible is `ITE`$((x \leq y), y, x)$, which the verifier finds to be a valid solution.

Given a candidate solution for the desired function, checking whether it satisfies the logical specification for all inputs is a standard verification problem, and we can rely upon a mature verification technology such as SMT solvers for this purpose.[5] Learning an expression from the set *Exp* of candidate expressions that satisfies the specification for the current inputs in *Examples* is a new challenge, and has been the focus of research in design and implementation of SyGuS solvers.

**Enumerative search.** Given a set *Exp* of candidate expressions specified by a (context-free) grammar, a finite set *Examples* of inputs, and a logical specification *Spec*, the learning problem is to find an expression $e$ in *Exp* such that *Spec*[$f/e$] is satisfied for all inputs in *Examples*. Furthermore, we want to find the simplest such expression.

The simplest solution to the learning problem is based on enumerating all expressions in *Exp* one by one in increasing order of size, and checking for each one if it satisfies *Spec* for all inputs in *Examples*. Since the number of expressions grows exponentially with the size, we need some heuristics to prune the search space. An optimization that turns out to be effective is based on a notion of equivalence among expressions with respect to the given set of inputs. Let us say that two expressions $e_1$ and $e_2$ are *Examples*-equivalent if for all inputs in *Examples*, $e_1$ and $e_2$ evaluate to the same value. Notice that if $e$ is an expression that contains $e_1$ as a subexpression, and if we obtain $e'$ by substituting $e_1$ by another expression $e_2$ that is *Examples*-equivalent to $e_1$, then $e'$ is guaranteed to be *Examples*-equivalent to $e$. As a result, the enumeration algorithm maintains a list of only inequivalent expressions. To construct the next expression, it uses only the expressions from this list as potential subexpressions, and as a new expression is constructed, it first checks if it is equivalent to one already in the list, and if so, discards it.

To illustrate the algorithm, suppose the logical specification is $Spec_2$, the set of expressions is $Exp_1$, and the current set of *Examples* contains a single input ($x = 0$, $y = 1$) (as noted earlier). The job of the learning algorithm is to find an expression $e$ that satisfies $Spec_2$ for $x = 0$ and $y = 1$, that is, $e(0, 1) > 1$. Two expressions are equivalent in this case if $e_1(0, 1) = e_2(0, 1)$. The enumerator starts by listing expressions of size 1 one by one. The first expression considered is $x$. It is added to the list, and since it does not satisfy the specification, the search continues. The next expression is $y$, which is inequivalent to $x$ and does not satisfy the specification, so is added to the list and the search continues. The next expression is 0, which turns out to be equivalent to $x$ (both evaluate to 0 for the input (0, 1)), and is hence discarded. The next expression is 1, which is also discarded as it is equivalent to $y$. Next the algorithm considers the expressions generated by the application of the rule $E + E$. The algorithm considers only $x$ and $y$ as potential subexpressions at this step, and thus, examines only $x + x$, $x + y$, $y + x$, and $y + y$, in that order. Of these, the first one is equivalent to $x$, and the next two are equivalent to $y$, and hence discarded. The expression $y + y$ is the only interesting example of size 3, and the algorithm checks if it satisfies the specification. Indeed that is the case, and the learner returns $y + y$. The verifier will discover that this solution does not satisfy the specification for all inputs, and will generate a counterexample, say, ($x = 1$, $y = 0$). Note that adding this input to *Examples* changes the notion of equivalence of expressions (for instance, the size 1 expressions $x$ and 0 are no longer equivalent), so in the next iteration, the learning algorithm needs to start enumeration from scratch.

We conclude the discussion of the enumerative search algorithm with a few observations. First, if the set *Exp* is unbounded, the algorithm may simply keep enumerating expressions of larger and larger size without ever finding one that satisfies the specification. Second, if we know (or impose) a bound $k$ on the depth of the expression we are looking for, the number of possible expressions is exponential in $k$. The equivalence-based pruning leads to significant savings, but the exponential dependence remains. Finally, to translate the idea described above to an actual algorithm that works for the set of expressions described by a context-free grammar some fine-tuning is needed. For example, consider the grammar for the set $Exp_2$ of linear expressions with conditionals, the algorithm needs to enumerate (inequivalent) expressions generated by both non-terminals T and C concurrently by employing a dynamic programming strategy (see Alur et al.[2] and Udupa et al.[32]).

## An Infrastructure for Solvers

In the world of constraint solving, the standardization of the input format, collection and categorization of a large number of benchmarks, access to open-source computational infrastructure, and organization of an annual competition of solvers, had a transformative impact on both the development of powerful computational techniques and the practical applications to diverse problems.[5]

This success inspired us to initiate a similar effort centered on the SyGuS problem (see www.sygus.org).

**Standardized input format.** To define a standardized input format for SyGuS, a natural starting point is the input format used by SMT solvers for two reasons. First, there is already a vibrant ecosystem of benchmarks, solvers, users, and researchers committed to the SMT format. Second, a typical SyGuS solution strategy (see Figure 1) needs to verify that a candidate solution satisfies the logical constraint, and the ability to use a standard SMT solver as a verifier is a big win.

The SyGuS input format SYNTH-LIB thus extends the format SMT-LIB2 for specifying logical constraints. This means that to define a problem, we must first choose one of the standard SMT logics. An example is LIA that can encode formulas in linear arithmetic with conditionals (essentially same as the set of expressions in the set $Exp_2$). Other commonly used logics for our purpose include BV for manipulating bit-vectors, LRA for linear arithmetic over reals, and SLIA for processing strings.

Once a logic is chosen, the problem definition next declares the name of the function to be synthesized along with the types of the input arguments and output value. These types must be from the underlying theory, for instance, boolean and integer types are possible in LIA. The function declaration also specifies the grammar for defining the set *Exp* of candidate expressions. This simply includes a list of typed nonterminals, including the special nonterminal `Start`, and a list of productions for each of them. The terminals in the grammar rules must be the symbols from the underlying logic used in a type-consistent manner. The unknown function itself cannot occur in the rules, meaning that we do not support synthesis of recursively defined functions in the current version.

The logical constraint *Spec* is specified as a formula that is built from the operations in the chosen logic and invocations of the function to be synthesized. In the current version, we require this formula to be free of quantifiers as is the case in examples mentioned previously. This means that once the learner returns a candidate expression *e*, the verifier needs to check the truth of *Spec*[*f*/*e*] with all the variables universally quantified. This amounts to checking the satisfiability of the quantifier-free formula ¬ *Spec*[*f*/*e*], a task for which contemporary SMT solvers are particularly effective.

The format allows specifying synthesis of multiple unknown functions simultaneously. It also allows the use of `let` expressions in the grammar rules. Such expressions can make synthesized solutions succinct, and are analogous to the use of auxiliary variables in imperative code.

**Benchmarks.** The benchmarks we have collected come from different domain areas, use different SMT logics, and different grammars. There are currently over 1,500 benchmarks. We give a few examples for benchmark categories.

The *hacker's delight* benchmarks are concerned with bit-manipulation problems from the book Hacker's Delight.[33] These benchmarks were among the first to be successfully tackled by synthesis technology.[12,13,29] Each such problem induces several benchmarks with varying grammars. The grammar in the easiest instances includes only the operators that are required to implement the desired transformations, whereas the grammar in the hardest instances is highly unconstrained, so the synthesizer must discover which operators to use in addition to how to compose them together.

SV-COMP is a competition of automated tools for software verification held annually in conjunction with ETAPS (European Joint Conferences on Theory and Practice of Software). In this competition, the verifier is tasked with checking correctness requirements (such as assertions) of C programs. When the program to be verified contains loops, this requires inference of an inductive loop invariant. Research on automated synthesis of invariants has used benchmarks of SV-COMP by converting fragments of C programs to logical formulas corresponding to verification conditions.[8] Augmenting these benchmarks with a syntactic template for the unknown invariant, typically using linear arithmetic with conditionals, leads to benchmarks for SyGuS solvers.

The *string* category of benchmarks consists of tasks that require learning programs to manipulate strings based on regular expressions and come from the public set of benchmarks of the FlashFill system (and its successors). These benchmarks are based on the newly supported theory SLIA in SMT-LIB, which supports string operations such as prefix, suffix, substring, length, and indexing.

The other sources of benchmarks include motion planning for robot movements, the 2013 ICFP Programming Competition[1] that included synthetic but challenging bitvector functions, program repair for introductory programming solutions and real-world programs,[15] compiler optimization, and synthesis of cryptographic circuits that are resilient to timing attacks[7] (as we will detail later).

**SyGuS-Comp: a competition of solvers.** In order to encourage the development of solvers for the SyGuS problem we initiated a competition of solvers called SyGuS-Comp. The solvers are compared on the basis of the number of benchmarks solved, the time taken to solve, and the size of the generated expressions. The first competition was held in 2014, and is now an annual event, co-located with the annual Computer Aided Verification Conference (CAV). The Star-Exec platform provides the computational infrastructure needed for the competition.[31]

The first competition consisted of a single track. The benchmarks in this track used the SMT logics LIA (conditional linear arithmetic) and BV (bit-vectors), and each benchmark provided its own context-free grammar to be used in the solution. The second competition consisted of three tracks: *the general track* that is same as the single track of the first competition, *the CLIA track* where logic is LIA and the grammar admits every LIA expression, and *the INV track* aimed at benchmarks for synthesis of loop invariants. This track is a restriction of the CLIA track, which consists of special syntactic sugaring of SyGuS problems, to allow direct encoding of inference of inductive invariants. The

third and fourth competitions consisted, in addition to these three tracks, *the PBE track* for *programming by examples*. The PBE track restricts semantic constraints to be based upon only input–output examples. This track is divided into two: benchmarks using the BV logic, and benchmarks manipulating strings expressed in the SLIA logic.

The ESolver based on the enumerative search strategy described previously won the first competition. Since then a number of researchers have proposed new solution strategies. For instance, the ICE-DT solver is specialized to learning invariants based on a novel idea of generalizing from implication counterexamples (as opposed just positive and negative examples common in classical learning), and won the INV track in recent competitions,[8] and the strategy to solve alternation of quantifiers within the SMT solver CVC4 was modified to produce witness functions that match the syntactic template leading to a SyGuS solver that is the most effective current solver for the CLIA and PBE-String tracks.[25] The winner of the general track in the 2017 competition is EUSolver, which we will explore.

## State of the Art

The formalization of the SyGuS problem and organization of the annual competition of solvers has been a catalyst for research in search-based program synthesis. We first give an overview of the progress in solver technology, then describe the solution strategy employed by the current winner, and explain a novel application of SyGuS to synthesis of cryptographic circuits resistant to timing attacks.

**Evolution of SyGuS solvers.** The capabilities of SyGuS solvers are improving from competition to competition. For instance, all instances of the ICFP benchmarks were solved in 2017 competition, most in less than 10 seconds. In contrast, none of these were solved in the first competition, and in the original ICFP competition, some of these benchmarks were solved by the participants using enormously large compute clusters.

As another example, recall the example given earlier of synthesizing the maximum of two numbers using the grammar $Exp_3$. For any number $n$, we can similarly write a specification for computing the maximum of $n$ input arguments. In the first competition all solvers were able to solve for $n = 2$, only one solver was able to solve for $n = 3$ and none could solve for $n = 4$. The 2017 solvers are capable of solving for $n \leq 21$. Note that the size of the minimal expression grows quadratically with $n$. The size of the expression generated for $n = 21$ is 1621. With regard to the time to solve these benchmarks, instances with $n \leq 10$ are solved within 5s, whereas the solution for $n = 21$ required 2100 s.

**The formalization of the SyGuS problem and organization of the annual competition of solvers has been a catalyst for research in search-based program synthesis.**

Trying to understand which SyGuS instances are easily solved by current SyGuS solvers, we recall different aspects of a SyGuS instance: (i) The grammar can be very *general* or very *restrictive*, depending on the size of the set of syntactically allowed expressions. (ii) The specification can require a *single* or *multiple* functions to be simultaneously synthesized. (iii) The specification can be *complete* or *partial* depending on whether the set of semantic solutions is a singleton or not. (iv) The grammar may or may not allow the use of `let` for specifying auxiliary variables. (v) When the specification has several invocations of the function to be synthesized, all invocations may be exactly the same (in the sense that the sequence of parameters is the same in all) or there may be different ways in which the function is invoked. We refer to the former as *single invocation* and to the latter as *multiple invocation*.[25] The categories of benchmarks in which state-of-the-art solvers excel are those with a single function invocation, a single function to synthesize, a complete specification, no use of `let`, and a restricted grammar. Benchmarks of the CLIA and invariant generation tracks are also easily handled by current solvers, in spite of their grammar being general.

**Enumerative search with decision trees.** When the grammar specifying the set of allowed expressions includes conditionals, the desired solution is typically a tree whose internal nodes are labeled with tests used in conditionals and leaves are labeled with test-free expressions. The key

idea behind the optimization to the enumerative search employed by the 2017 winning solver, EUSolver, is to find expressions suitable as labels in the desired tree by enumeration, and construct the desired tree using the well-studied heuristic for decision tree learning from machine learning literature.[20,23] We will illustrate the mechanics of this algorithm, and why its performance is superior to the enumerative search, using the logical specification $Spec_1$ and the set $Exp_2$ of conditional linear expressions. Recall that the correct solution to this synthesis problem is the expression `ITE`$((x \leq y), y, x)$, which corresponds to an expression tree of size 6. The enumerative search algorithm thus has to process all expressions of size 5 or less that are inequivalent with respect to the current set *Examples*.

To understand the divide-and-conquer strategy of EUSolver, let us ignore the pruning based on *Examples*-equivalence for now. Suppose the algorithm starts enumerating expressions in $Exp_2$ in increasing order of size, and checks for each one if it satisfies $Spec_1$ for all inputs in the current set *Examples*. The expressions of size 1, namely, 0, 1, $x$, and $y$, are considered first. Suppose none of them satisfies the specification for all inputs in *Examples* (this will be the case, for instance, if it contains both $(0, 1)$ and $(1, 0)$). However, no matter what inputs belong to *Examples*, one of the terms $x$ or $y$ satisfies the specification for every input in *Examples*. In other words, the terms $x$ and $y$ *cover* the current set, and can be viewed as *partial solutions*. If such partial solutions can be combined using conditional tests, then this can already yield a solution that satisfies all inputs in *Examples* without enumerating terms of larger sizes. The EUSolver consists of a module that enumerates predicates (that is, tests used in conditionals) concurrently in increasing size. The test $(x \leq y)$ is a predicate of smallest possible size.

Given such a test, the set of inputs divides naturally into two sets, $Examples^1$ for which the test is true and $Examples^0$ for which the test is false. Observe that the partial solution $y$ works for all inputs in $Examples^1$ while the partial solution $x$ works for all inputs in $Examples^0$. Thus, the learner can return `ITE`$((x \leq y), y, x)$ as a candidate expression.

**The categories of benchmarks in which state-of-the-art solvers excel are those with a single function invocation, a single function to synthesize, a complete specification, no use of `let`, and a restricted grammar.**

In general, consider a set *Examples* of inputs, a set $L$ of terms enumerated so far, and a set $P$ of conditional tests enumerated so far. Suppose the terms in $L$ cover all inputs, that is, for each input in *Examples*, there is at least one term in $L$, which satisfies the specification for this input. The computational problem is now to construct a conditional expression with tests in $P$ and leaf expressions in $L$. A natural recursive algorithm to construct the decision tree is to first choose a test $p$ in $P$, learn a conditional expression $e_1$ for the subset $Examples^1$ of inputs for which the test $p$ is true, learn a conditional expression $e_0$ for the subset $Examples^0$ of inputs for which the test is false, and return `ITE`$(p, e_1, e_0)$. The effectiveness of this algorithm, that is, how many tests the final expression contains, depends on how the splitting predicate $p$ is chosen. The construction of the desired tree can be formalized as a decision tree learning problem: one can think of the inputs in *Examples* as instances, terms in $L$ as labels where a data point is labeled with a term if the term satisfies the specification for that input, and tests in $P$ as attributes. The greedy heuristic for constructing a small decision tree selects a test $p$ as the first decision attribute if it leads to the maximum information gain among all possible tests, where the gain is calculated from the so-called *entropy* of the sets $Examples^1$ and $Examples^0$ of data points as split by the attribute $p$.

The idea of considering only those terms and tests that are inequivalent with respect to current inputs is orthogonal to the above divide-and-conquer strategy, and can be integrated in it.

**Repairing cryptographic circuits.** Consider a circuit $C$ with a set $I_0$ of *private* inputs and a set $I_1$ of *public* inputs such that if an attacker changes the values of the public inputs and observes the corresponding output, she is unable to infer the values of the private inputs (under standard assumptions about computational resources in cryptography). The private inputs can correspond to a user's

secret key, the public inputs can correspond to a message, and the output can be the encryption of the message using the secret key. Such cryptographic circuits are commonplace in encryption systems used in practice. One possible attack, that is, a strategy for the attacker to gain information about the private inputs despite the established logical correctness of the circuit, is based on measuring the time it takes for the circuit to compute. For instance, when a public input bit changes from 1 to 0, a specific output bit is guaranteed to change from 1 to 0 independent of whether a particular private input bit is 0 or 1, but may change faster when this private input is 0, thus leaking information. Such vulnerabilities do occur in practice, and in fact, Ghalaty et al.[9] reports such an attack on a circuit used in the AES encryption standard. The timing attack is not possible if the circuit meets the so-called structural property of being *constant time*: A constant-time circuit is the one in which the length of all input-to-output paths measured in terms of number of gates are the same. After identifying the attack, Ghalaty et al.[9] shows how to convert the given circuit to an equivalent constant-time circuit by introducing delay elements on shorter paths.

As noted in the work of Eldib et al.,[7] being constant-time is a syntactic constraint on the logical representation of a circuit, that is, it depends on the structure of the expression and not on the operators used in the construction. As a result, given a circuit $C$, synthesizing another circuit $C'$ such that $C'$ is a constant-time circuit and is functionally equivalent to $C$ can be formalized as a SyGuS problem. The set of all constant-time circuits with all input-to-output path lengths within a given bound can be expressed using a context-free grammar and being logically equivalent to the original circuit can be expressed as a Boolean formula involving the unknown circuit. Eldib et al.[7] then use the EUSolver to automatically synthesize constant-time circuits that are logically equivalent to given cryptographic circuits, and in particular, report a solution that is smaller in terms of overall size as well as path lengths compared to the manually constructed one in the work of Ghalaty et al.[9]

## Conclusion

Search-based program synthesis promises to be a useful tool for future program development environments. Programming by examples in domain-specific applications and semantics-preserving optimization of program fragments to satisfy performance goals expressed via syntactic criteria are already proving to be its interesting applications. Our experience shows that investing in the infrastructure—standardized input formats, collection of benchmarks, open-source prototype solvers, and a competition of solvers—has been vital in advancing the state of the of art. Finally, improving the scalability of SyGuS solvers is an active area of current research, and in particular, a promising research direction is to explore how these solvers can benefit from modern machine learning technology (see, for example, Devlin et al.[6] for the use of neural networks for learning programs from input-to-output examples). C


**References**

1. Akiba, T., Imajo, K., Iwami, H., Iwata, Y., Kataoka, T., Takahashi, N., Mmoskal, M., Swamy, N. Calibrating research in program synthesis using 72,000 hours of programmer time. Technical Report, MSR, 2013.
2. Alur, R., Bodík, R., Juniwal, G., Martin, M.M.K., Raghothaman, M., Seshia, S.A., Singh, R., Solar-Lezama, A., Torlak, E., Udupa, A. Syntax-guided synthesis. In *Proc. FMCAD*, 2013, 1–17.
3. Alur, R., Radhakrishna, A., Udupa, A. Scaling enumerative program synthesis via divide and conquer. In *Proc. TACAS*, LNCS 10205, 2017, 319–336.
4. Clarke, E., Grumberg, O., Peled, D. *Model Checking*. MIT Press, 2000.
5. de Moura, L., Bjørner, N. Satisfiability Modulo Theories: Introduction and applications. *Commun. ACM* 54, 9 (2011), 69–77.
6. Devlin, J., Uesato, J., Bhupatiraju, S., Singh, R., Mohamed, A., Kohli, P. Robustfill: Neural program learning under noisy I/O. In *Proc. ICML*, 2017, 990–998.
7. Eldib, H., Wu, M., Wang, C. Synthesis of fault-attack countermeasures for cryptographic circuits. In *Proc. CAV*, LNCS 9780, 2016, 343–363.
8. Garg, P., Löding, C., Madhusudan, P., Neider, D. ICE: A robust framework for learning invariants. In *Proc. CAV*, LNCS 8559, 2014, 69–87.
9. Ghalaty, N., Aysu, A., Schaumont, P. Analyzing and eliminating the causes of fault sensitivity analysis. In *Proc. DATE*, 2014, 1–6.
10. Gulwani, S. Automating string processing in spreadsheets using input–output examples. In *Proc. POPL*, 2011, 317–330.
11. Gulwani, S., Harris, W.R., Singh, R. Spreadsheet data manipulation using examples. *Commun. ACM*, 55, 8 (2012), 97–105.
12. Gulwani, S., Jha, S., Tiwari, A., Venkatesan, R. Synthesis of loop-free programs. In *Proc. PLDI*, 2011, 62–73.
13. Jha, S., Gulwani, S., Seshia, S.A., Tiwari, A. Oracle-guided component-based program synthesis. In *Proc. ICSE*, 2010, 215–224.
14. Kuncak, V., Mayer, M., Piskac, R., Suter, P. Software synthesis procedures. *Commun. ACM*, 55, 2.
15. Le, X.D., Chu, D., Lo, D., Le Goues, C., Visser, W. S3: Syntax- and semantic-guided repair synthesis via programming by examples. In *Proc. FSE*, 2017, 593–604.
16. Malik, S., Zhang, L. Boolean satisfiability: From theoretical hardness to practical success. *Commun. ACM*, 52, 8 (2009), 76–82.
17. Manna, Z., Waldinger, R. Fundamentals of deductive program synthesis. *IEEE Trans. Softw. Eng.* 18, 8 (1992), 674–704.
18. Massalin, H. Superoptimizer – A look at the smallest program. In *Proc. ASPLOS*, 1987, 122–126.
19. Mechtaev, S., Yi, J., Roychoudhury, A. Angelix: Scalable multiline program patch synthesis via symbolic analysis. In *Proc. ICSE*, 2016, 691–701.
20. Mitchell, T. *Machine Learning*. McGraw-Hill, 1997.
21. Osera, P., Zdancewic, S. Type-and-example-directed program synthesis. In *Proc. PLDI*, 2015, 619–630.
22. Polikarpova, N., Kuraj, I., Solar-Lezama, A. Program synthesis from polymorphic refinement types. In *Proc. PLDI*, 2016, 522–538.
23. Quinlan, J. Introduction to decision trees. *Mach. Learn.* 1, 1 (1986), 81–106.
24. Raychev, V., Vechev, M.T., Yahav, E. Code completion with statistical language models. In *Proc. PLDI*, 2014, 419–428.
25. Reynolds, A., Deters, M., Kuncak, V., Tinelli, C., Barrett, C.W. Counterexample-guided quantifier instantiation for synthesis in SMT. In *Proc. CAV*, 2015, 198–216.
26. Schkufza, E., Sharma, R., Aiken, A. Stochastic program optimization. *Commun. ACM* 59, 2 (2016), 114–122.
27. Seshia, S.A. Combining induction, deduction, and structure for verification and synthesis. *Proc. IEEE* 103, 11 (2015), 2036–2051.
28. Solar-Lezama, A. Program sketching. *STTT* 15, 5–6 (2013), 475–495.
29. Solar-Lezama, A., Rabbah, R., Bodík, R., Ebcioglu, K. Programming by sketching for bit-streaming programs. In *Proc. PLDI*, 2005, 281–294.
30. Srivastava, S., Gulwani, S., Foster, J.S. Template-based program verification and program synthesis. *STTT* 15, 5–6 (2013), 497–518.
31. Stump, A., Sutcliffe, G., Tinelli, C. Starexec: A cross-community infrastructure for logic solving. In *Proc. IJCAR*, 2014, 367–373.
32. Udupa, A., Raghavan, A., Deshmukh, J., Mador-Haim, S., Martin, M., Alur, R. TRANSIT: Specifying protocols with concolic snippets. In *Proc. PLDI*, 2013, 287–296.
33. Warren, H.S. *Hacker's Delight*. Addison-Wesley, 2002.



**Rajeev Alur** is the Zisman Family Professor in the Department of Computer and Information Sciences at the University of Pennsylvania, Philadelphia, PA, USA.

**Rishabh Singh** is a research scientist at Google Brain, Mountain View, CA, USA.

**Dana Fisman** is a senior lecturerat Ben Gurion University, Be'er Shera, Israel.

**Armando Solar-Lezama** is an associate professor and leader of the Computer Assisted Programming Group at MIT, Cambridge, MA, USA.

# Inviting Young Scientists

HEIDELBERG LAUREATE FORUM

acm Association for Computing Machinery

## Meet Great Minds in Computer Science and Mathematics

As one of the founding organizations of the Heidelberg Laureate Forum **http://www.heidelberg-laureate-forum.org/**, ACM invites young computer science and mathematics researchers to meet some of the preeminent scientists in their field. These may be the very pioneering researchers who sparked your passion for research in computer science and/or mathematics.

These laureates include recipients of the ACM A.M. Turing Award, the Abel Prize, the Fields Medal, and the Nevanlinna Prize.

The 7th Heidelberg Laureate Forum will take place **September 22–27, 2019** in Heidelberg, Germany.

This week-long event features presentations, workshops, panel discussions, and social events focusing on scientific inspiration and exchange among laureates and young scientists.

### Who can participate?

New and recent Ph.Ds, doctoral candidates, other graduate students pursuing research, and undergraduate students with solid research experience and a commitment to computing research

### How to apply:

Online: **https://application.heidelberg-laureate-forum.org/**
Materials to complete applications are listed on the site.

### What is the schedule?

The application deadline is **February 15, 2019.**

We reserve the right to close the application website early depending on the volume

Successful applicants will be notified by **mid April 2019**.

*More information available on Heidelberg social media*

PHOTOS: ©HLFF / B. Kreutzer (top); ©HLFF / C. Flemming (bottom)

# research highlights

DOI:10.1145/3282504

To view the accompanying paper, visit doi.acm.org/10.1145/3282506 rh

# Technical Perspective
# Node Replication Divides to Conquer

By Tim Harris

SHARED-MEMORY CONCURRENT DATA structures are pervasive. They are used explicitly in server software such as in-memory databases and key-value stores. Even when software is built with a programing model based around shared-nothing computation or side-effect-free functional programming, we find concurrent data structures in the heart of the implementation in the operating system, the language runtime system, or the garbage collector.

Designing these high-performance concurrent data structures has long been recognized as a difficult task. Not only is it challenging to develop an implementation that is correct, but the underlying hardware is a moving target; techniques that work well on one system may work poorly on another, and techniques that work on today's systems may work poorly on tomorrow's.

What better way to simplify this task than to have an automatic technique to generate a concurrent data structure from existing sequential code? That is the goal set by Calciu et al. in their work on Node Replication (NR). In the following article, they show that not only *can* a concurrent data structure be built automatically, but that performance is actually competitive with state-of-the-art designs for a series of important workloads.

NR achieves this good performance by recognizing that the bottleneck for many concurrent data structures is the memory accesses that are made—particularly when threads on different sockets are "fighting" over the same cache lines, or when threads on one socket are accessing data stored on a different socket. NR reduces these costs by maintaining per-socket synchronized replicas of a data structure, and routing a thread's requests to its own local replica.

This is a surprising and inspiring result, particularly given that building "universal" constructions for concurrent data structures has been an active research field for over 30 years. Prior work has made important theoretical contributions, not the least of which around the importance of instruction set support for atomic operations such as compare-and-swap. However, NR takes that exploration further both in reaching the point where practical implementations can perform well in some workloads, and also in illustrating the benefits of "mechanical sympathy" between the techniques in the implementation and the physical structure of the underlying machine.

An exciting implication of this paper is it provides a division of responsibility: The implementer of a data structure is responsible for its correctness and for making it efficient in the absence of concurrency. The implementer of NR is responsible for building the replication and routing mechanisms efficiently for a particular machine; improvements to these mechanisms will help any data structure using them. For example, if a new machine has specialized hardware for sending messages between sockets then that could be used within NR without needing to change data structures.

In the longer term, it is interesting to think about broader applications of the ideas used in NR. One is to support shared data structures on machines without hardware cache coherence—both at the scale of multiple processors combined in a system-on-chip, and also in a distributed context between separate machines with a high-performance interconnect. C

**Tim Harris**, Cambridge, U.K.

DOI:10.1145/3282506

# How to Implement Any Concurrent Data Structure

By Irina Calciu, Siddhartha Sen, Mahesh Balakrishnan, and Marcos K. Aguilera

**Abstract**

**We propose a method called Node Replication (NR) to implement any concurrent data structure. The method takes a single-threaded implementation of a data structure and automatically transforms it into a concurrent (thread-safe) implementation. The result is designed to work well with and harness the power of modern servers, which are complex Non-Uniform Memory Access (NUMA) machines with many processor sockets and subtle performance characteristics. Using NR requires no expertise in concurrent data structure design, and the result is free of concurrency bugs. NR represents a paradigm shift of how concurrent algorithms are developed: rather than designing for a data structure, we design for the architecture.**

## 1. INTRODUCTION

Concurrent data structures are everywhere in the software stack, from the kernel (e.g., priority queues for scheduling), to application libraries (e.g., tries for memory allocation), to applications (e.g., balanced trees for indexing). These data structures, when inefficient, can cripple the performance of the system.

Due to recent architectural changes, high-performance servers today are Non-Uniform Memory Access (NUMA) machines. Such machines have multiple processor sockets, herein called *nodes*, each with some local cache and memory. Although cores in a node can access the memory in other nodes, it is faster to access local memory and to share cache lines within a node than across nodes. To fully harness the power of NUMA, data structures must take this asymmetry into consideration: they must be NUMA-aware to reduce cross-node communication and minimize accesses to remote caches and memory.

Unfortunately, there are few NUMA-aware concurrent data structures, and designing new ones is hard. The key challenge is how to deal with contention on the data structure, where simple techniques limit concurrency and scale poorly, while efficient techniques are complex, error-prone, and rigid (Section 2).

We propose a new technique, called Node Replication (NR), to obtain NUMA-aware data structures, by automatically transforming any single-threaded data structure into a corresponding concurrent (thread-safe) NUMA-aware structure. NR is general and *black-box:* it requires no inner knowledge of the structure and no expertise in NUMA software design. The resulting concurrent structure provides strong consistency in the form of linearizability.[8]

Node Replication combines ideas from two disciplines: distributed systems and shared-memory algorithms. NR maintains per-node replicas of an arbitrary data structure and synchronizes them via a shared log (an idea from distributed systems[1]). The shared log is realized by a hierarchical, NUMA-aware design that uses flat combining[5] within nodes and lock-free appending across nodes (ideas from shared-memory algorithms). With this interdisciplinary approach, only a handful of threads need to synchronize across nodes, so most synchronization occurs efficiently within each node.

Node Replication represents a paradigm shift of how concurrent algorithms are designed. Currently, each new concurrent data structure requires its own design, and our community of experts has spent decades writing papers and developing algorithms for all kinds of structures (skip lists, queues, priority queues, and hash tables, etc). However, computer architectures are now fluid with the introduction of new memory features (non-volatility, in-memory processing), new memory models (NUMA, non-coherent caches), new processing elements (GPU, FPGA, TPU), new processor features (transactional memory, SGX), and more. Unfortunately, the old algorithms do not work well in the new architectures, so the community has to redesign the algorithms for each new architecture.

Node Replication shows there is a better way to design algorithms, by using a black-box approach that is independent of the data structure. Thus, rather than designing for a data structure, we design for the architecture. This approach significantly reduces the design effort to a few architectures, instead of the product of the number of architectures and the number of data structures. While we demonstrate the black-box approach for NUMA here, we envision its general applicability to other new architectures as they emerge.

Node Replication cannot always outperform algorithms that specialize for a single data structure and architecture. However, perhaps surprisingly, NR performs well in many cases, particularly when there is contention, where an operation often affects the output of other operations. On a contended priority queue and a dictionary, NR can outperform lock-free algorithms by up to 2.4x and 3.1x with 112 threads; and NR can outperform a lock-based solution by 8x and 30x on the same data structures. To demonstrate the benefits to a real application, we apply NR to the data structures of the Redis storage server. Many systems have shown how servers can scale the handling of network requests and minimize Remote Procedure Calls (RPC) bottlenecks.[10] There is less research on how to scale the servicing of the requests. These

The original version of this paper, titled "*Black-box Concurrent Data Structures for NUMA Architectures*," was published in ASPLOS 2017. For more information, please check https://research.vmware.com/projects/nodereplication.

systems either implement a simple service (e.g., get/put) that can partition requests across cores;[10] or they develop sophisticated concurrent data structures from scratch to support more complex operations,[11] and doing this requires expertise in concurrent algorithms. This is where our black-box approach comes handy: NR provides these concurrent data structures automatically from single-threaded implementations. For Redis, we were able to convert a single-threaded sorted set into a concurrent one with just 20 new lines of wrapper code. The result outperforms data structures obtained from other methods by up to 14x.

Although NR is powerful, easy to use, and efficient, it has three limitations. First, it incurs space overhead due to replication: it consumes $n$ times more memory, where $n$ is the number of nodes. Thus, NR is best suited for smaller structures that occupy just a fraction of the available memory (e.g., up to hundreds of MB). Second, NR is *blocking*: a thread that stops executing operations can block the progress of other threads; in practice, we did not find that to be a problem as long as threads keep executing operations on the data structure. Finding a non-blocking variant of NR is an interesting research direction. Finally, NR may be outperformed by non-black-box algorithms crafted for a given data structure—For example, a lock-free skip list running on low-contention workloads, or a NUMA-aware stack.[2] Thus, the generality of black-box methods has some cost. However, in some cases NR outperforms even the crafted algorithms; we observe this for the same lock-free skip list running instead on high-contention workloads.

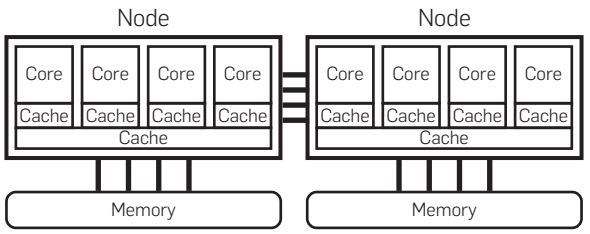
We plan to make the source code for NR available in our project page at https://research.vmware.com/projects/nodereplication.

## 2. BACKGROUND

### 2.1. NUMA architectures

Our work is motivated by recent trends in computer architecture. To support a large number of cores, data center servers have adopted a NUMA architecture with many processor sockets or *nodes* (see Figure 1). Each node has many processor cores and a shared cache, while individual cores have private caches. Sharing a cache line within a node is more efficient than across nodes because the cache coherence protocol operates more efficiently within a node. Each node has some local memory, and a core can access local memory faster than memory in a remote node. A similar architecture—Non-Uniform Cache Access (NUCA)—has a single shared memory but nodes have local caches as in NUMA. Our ideas are applicable to NUCA too. NUMA is everywhere now. A high-performance Intel server might have eight processors (nodes), each with 28 cores, while a typical server might have two processors, each with 8–16 cores. AMD and Oracle have similar machines. To best use these cores, we need appropriate concurrent data structures.

**Figure 1. NUMA architecture of a modern server in a data center. The server has many processor sockets, herein called nodes. Each node has many processor cores and some local memory. Nodes are connected by an interconnect, so that cores in one node can access the remote memory of another node, but these accesses come at a cost. Typically, cores have local caches, and cores on a node share a last level cache.**




### 2.2. Concurrent data structures

Concurrent data structures permit many threads to operate on common data using a high-level interface. When a data structure is accessed concurrently by many threads, its semantics are typically defined by a property called linearizability,[8] which provides strong consistency. Linearizability requires that each operation appear to take effect instantly at some point between the operation's invocation and response.

The key challenge in designing concurrent data structures is dealing with *operation contention*, which occurs when an operation often affects the output of another operation. More precisely, given an execution, we say that an operation affects another if the removal of the first causes the second to return a different result. For example, a write of a new value affects a subsequent read. A workload has operation contention if a large fraction of operations affect a large fraction of operations occurring soon after them. Examples include a storage system where users read and write a popular object, a priority queue where threads often remove the minimum element, a stack where threads push and pop data, and a bounded queue where threads enqueue and dequeue data. Non-examples include read-only workloads and write-only workloads where writes do not return a result. Operation contention is challenging because operations must observe each other across cores.

Much work has been devoted to designing and implementing efficient concurrent data structures; we provide a broad overview in Calciu, Sen et al.[3] Unfortunately, each data structure requires its own algorithm with novel techniques, which involve considerable work from experts in the field. To get a sense, a new concurrent data structure often leads to a scientific publication just for its algorithm.

Unfortunately, most existing concurrent data structures and techniques are for Uniform Memory Access (UMA), including some prior black-box methods.[5, 6, 16] These algorithms are not sensitive to the asymmetry and limitations of NUMA, which hinders their performance.[9] There are some recent NUMA-aware algorithms,[2, 12, 14] but they cover few data structures. Moreover, these solutions are not applicable when applications compose data structures and wish to modify several of them with a single composed operation (e.g., remove an item from a hash table and a skip list simultaneously). This is the case in the Redis application, which we describe later in the paper.

## 3. NODE REPLICATION (NR)

Node Replication is a NUMA-aware algorithm for concurrent data structures. Unlike traditional algorithms, which target a specific data structure, NR implements *all* data structures at once. Furthermore, NR is designed to work well under operation contention. Specifically, under update-heavy contended workloads, some algorithms drop performance as we add more cores; in contrast, NR can avoid the drops, so that the parallelizable parts of the application can benefit from more cores without being hindered by the data structures. NR cannot always outperform specialized data structures with tailored optimizations, but it can be competitive in a broad class of workloads.

While NR can provide any concurrent data structures, it does not automatically convert entire single-threaded applications to multiple threads. Applications have a broad interface, unlike data structures, so they are less amenable to black-box methods.

### 3.1. API

To work with an arbitrary data structure, NR expects a single-threaded implementation of the data structure provided as four generic methods:

$Create() \rightarrow ptr$
$Execute(ptr, op, args) \rightarrow result$
$IsReadOnly(ptr, op) \rightarrow Boolean$
$Destroy()$

The *Create* method creates an instance of the data structure, returning its pointer. The *Execute* method takes a data structure pointer, an operation, and its arguments; it executes the operation on the data structure, returning the result. The method must produce side effects only on the data structure and it must not block. Operation results must be deterministic, but we allow nondeterminism inside the operation execution and the data structure (e.g., levels of nodes in a skip list). Similarly, operations can use randomization internally, but results should not be random (results *can* be pseudorandom with a fixed initial seed). The *IsReadOnly* method indicates if an operation is read-only; we use this information for read-only optimizations in NR. The *Destroy* method deallocates the replicas and the log. NR provides a new method *ExecuteConcurrent* that can be called concurrently from different threads.

For example, to implement a hash table, a developer provides a Create method that creates an empty hash table; an Execute method that recognizes three *op* parameters (insert, lookup, remove) with the *args* parameter being a key-value pair or a key; and a IsReadOnly method that returns true for op=lookup and false otherwise. The Execute method implements the three operations of a hash table in a single-threaded setting (not thread-safe). NR then provides a concurrent (thread-safe) implementation of the hash table via a new method *ExecuteConcurrent*. For convenience, the developer may subsequently write three simple wrappers (insert, lookup, remove) that invoke *ExecuteConcurrent* with the appropriate *op* parameter.

### 3.2. Basic idea

Node Replication replicates the data structure on each NUMA node, so that threads can execute operations on a replica that is local to their node. Replication brings two benefits. First, an operation can access the data structure on memory that is local to the node. Second, operations can execute concurrently across nodes on different replicas. Replication, however, raises the question of how threads coordinate access to the replicas and maintain them in sync.

For efficiency, NR uses different mechanisms to coordinate threads within nodes and across nodes. At the highest level, NR leverages the fact that coordination within a node is cheaper than across nodes.

Within each node, NR uses flat combining (a technique from concurrent computing[5]). Flat combining batches operations from multiple threads and then executes the batch using a single thread, called the *combiner*. The combiner is analogous to a leader in distributed systems. In NR, we batch operations from threads in the same node, using one combiner per node. The combiner of a node is responsible for checking if threads within the node have any outstanding update operations, and then it executes all such operations on behalf of the other threads. Which thread is the combiner? The choice is made dynamically among threads within a node that have outstanding operations. The combiner changes over time: it abdicates when it finishes executing the outstanding updates, up to a maximum number. Batching can gather many operations, because there are many threads per node (e.g., 28 in our machine). Batching in NR is advantageous because it localizes synchronization within a node.

Across nodes, threads coordinate through a shared log (a technique from distributed systems[1]). The combiner of each node reserves entries in the log, writes the outstanding update operations to the log, brings the local replica up-to-date by replaying the log if necessary, and executes the local outstanding update operations.

Node Replication applies an optimization to *read-only* operations (operations that do not change the state of the data structure). Such operations execute without going through the log, by reading directly the local replica. To ensure consistency (linearizability[8]), the operation must ensure that the local replica is fresh: the log must be replayed at least until the last operation that completed before the read started.

We have considered an additional optimization, which dedicates a thread to run the combiner for each node; this thread replays the log proactively. This optimization is sensible for systems that have many threads per node, which is an ongoing trend in processor architecture. However, we have not employed this optimization in the results we present here.

The techniques above provide a number of benefits:

- *Reduce Cross-Node Synchronization and Contention:* NR appends to the log without acquiring locks; instead, it uses the atomic Compare-And-Swap (CAS) instruction on the log tail to reserve new entries in the log. The CAS instruction incurs little cross-node synchronization

because only the combiners execute the CAS, and there is at most one combiner per node—hence synchronization required for the CAS involves only a few threads (typically 2–8). In addition, the cost of a CAS is amortized over many operations due to batching.

- *Read and Write to the Log in Parallel:* Combiners can concurrently read the log to update their local replicas. Moreover, combiners can also concurrently write to the log: after combiners have reserved new entries using CAS, combiners can fill their entries concurrently.
- *Read Locally in Parallel:* Read-only operations in the data structure execute against the local replica, and so they can proceed in parallel if the replica is fresh. Checking for freshness might fetch a cache line across nodes, but this fetch populates the local cache and benefits many local readers. Readers execute in parallel with combiners on different nodes, and with the local combiner when it is filling entries in the log.
- *Use Compact Representation of Shared Data:* Operations often have a shorter description than the effects they produce, and thus communicating the operation via the log incurs less communication across cores than sharing the modifications to the data structure. For example, clearing a dictionary might modify many parts of the data structure, but we only communicate the operation description across nodes.

A complication that must be addressed is how to recycle the log. This must be done without much coordination, for performance, but must also ensure that a log entry is recycled only after it has been applied at all the replicas. Roughly speaking, NR uses a lightweight lazy mechanism that reduces synchronization by delegating responsibility of recycling to one of the threads.

In what follows, we describe these ideas in more detail.

### 3.3. Intra-node coordination: combining

To execute an operation, a thread posts its operation in a reserved slot[a] and tries to become the combiner by acquiring the *combiner lock*. The combiner reads the slots of the threads in the node and forms a batch $B$ of operations to execute. The combiner then proceeds to write the operations in $B$ to the log, and to update the local replica with the entries from the log.

To avoid small inefficient batches, the combiner in NR waits if the batch size is smaller than a parameter *min_batch*. Rather than idle waiting, the combiner refreshes the local replica from the log, though it might need to refresh it again after finally adding the batch to the log. Figure 2 depicts the general ideas.

### 3.4. Inter-node coordination: circular buffer

Node Replication replicates the data structure across nodes using a log realized as a shared circular buffer that stores update operations on the data structure. This buffer can be allocated from the memory of one of the NUMA nodes, or it could be spread across nodes. The log is accessed by at most one thread per node, and it provides coordination and consistency across nodes.

A variable *logTail* contains the index of the next available entry. Each node has a replica of the data structure and a variable *localTail* indicating how far in the log the replica has been updated. A node elects a temporary leader thread called a *combiner* to write to the buffer (Section 3.3).

The combiner writes many operations (a batch) to the log at a time. To do so, it first allocates space by using a CAS to advance *logTail* by the batch size. Then, it writes the buffer entries with the operations and arguments. Next, it updates the local replica by replaying the entries from

**Figure 2. NR replicates the data structure across nodes. A shared log stores updates that are later applied to each replica. Here, there are two nodes and hence two replicas of a tree. The replicas are not in sync, because the right replica has incorporated more updates from the shared log. Threads in the same node share the replica in that node; they coordinate access to the replica using a lock and a technique called flat combining (Section 3.3). Flat combining is particularly efficient in UMA systems. Effectively, NR treats each node as a separate UMA system.**

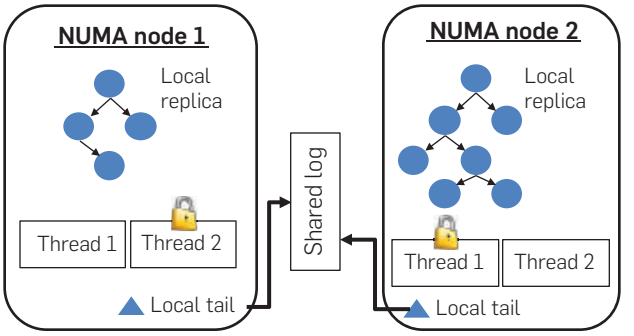
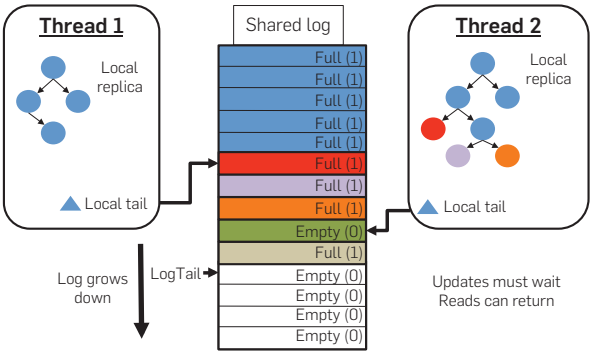



**Figure 3. The shared log in NR is realized as a circular buffer, shown here as an array for simplicity. There is a global *log-Tail* variable that indicates the first unreserved entry in the log. Each node has a *localTail* variable that indicates the next operation in the log to be executed on each local replica. The figure shows only one thread for each node—the thread that is currently chosen as the combiner for that node—but there are other threads. Thread 1's replica executed 5 operations from the log. Thread 2's replica executed 3 more operations and found an "empty" reserved entry that is not yet filled. A combiner must wait for all empty entries preceding its batch in the log. Readers can return when they find an empty entry (Section 3.6).**




[a] We call *slots* the locations where threads post operations for the combiners; we call *entries* the locations in the shared log.

*localTail* to right before the entries it allocated. In doing so the combiner may find empty entries allocated by other threads; in that case, it waits until the entry is filled (identified by a bit in the entry). Figure 3 shows two combiners accessing the log to update their local replicas, which they do in parallel.

### 3.5. Recycling log entries

Each log entry has a bit that alternates when the log wraps around to indicate empty entries. An index *logMin* stores the last known safe location to write; for efficiency, this index is updated only when a thread reaches a *low mark* in the log, which is *max_batch* entries before *logMin*. The thread that reserves the low mark entry updates *logMin* to the smallest *localTail* of all nodes; meanwhile, other threads wait for *logMin* to change. This scheme is efficient: it incurs no synchronization and reads *localTail* rarely if the log is large. A drawback is that a slow node becomes a bottleneck if no thread on that node updates the *localTail*. This problem is avoided using a larger log size.

### 3.6. Read-only operations

Threads performing read-only operations (*readers*) do not reserve space in the log, because their operations do not affect the other replicas. Moreover, a reader that is updating from the log can return and proceed with the read if it encounters an empty entry. Unlike flat combining, NR optimizes read-only operations by executing them directly on the local replica using a readers-writer lock for each node. The combiner acquires the lock in write mode when it wishes to modify the local replica, while *reader* threads acquire the lock in read mode. To avoid stale reads that violate linearizability, a reader must ensure the local replica is fresh. However, the replica need not reflect all operations up to *logTail*, only to the last operation that had completed before the reader started. To do this, we keep a *completedTail* variable, which is an index $\leq$ *logTail* that points to a log entry after which there are no completed operations. After a combiner refreshes its local replica, it updates *completedTail* using a CAS to its last batch entry if it is smaller. A reader reads *completedTail* when it starts, storing it in a local variable *readTail*. If the reader sees that a combiner exists, it just waits until *localTail* $\geq$ *readTail*; otherwise, the reader acquires the readers-writer lock in writer mode and refreshes the replica itself.

### 3.7. Readers-combiner parallelism

Node Replication's algorithm is designed for readers to execute in parallel with combiners in the same node. In early versions of the algorithm, the combiner lock also protected the local replica against readers, but this prevented the desired parallelism. By separating the combiner lock and the readers-writer lock (Section 3.6), readers can access the replica while a combiner is reading the slots or writing the log, before it refreshes the replica. Furthermore, to enable parallelism, readers must wait for *completedTail* as described, not *logTail* because otherwise readers block on the hole created by the local combiner, despite the readers lock being available.

### 3.8. Better readers-writer lock

Vyukov's distributed readers-writer lock uses a per-reader lock to reduce reader overhead; the writer must acquire the locks from all readers. We modify this algorithm to reduce writer overhead as well, by adding an additional writer lock. To enter the critical section, the writer must acquire the writer lock and wait for all the readers locks to be released, without acquiring them; to exit, it releases its lock. A reader waits if the writer lock is taken, then acquires its local lock, and checks the writer lock again; if this lock is taken, the reader releases its local lock and restarts; otherwise, it enters the critical section; to exit, it releases the local lock. With this scheme, the writer and readers incur just one atomic write each on distinct cache lines to enter the critical section. Readers may starve if writers keep coming, but this is unlikely with NR, as often only one thread wishes to be a writer at a time (the combiner) and that thread has significant work outside the critical section.

### 3.9. Practical considerations

We now discuss some important practical considerations that arised when we implemented NR.

**Software and hardware threads.** So far, we have assumed that software threads correspond one-to-one with hardware threads, and we have used the term *thread* indistinguishably to refer to either of them. However, in practice applications may have many more software threads than available hardware threads. To handle this situation, we can have more combiner slots than hardware threads, and then assign each software thread to a combiner slot. Beyond a certain number of software threads, they can share combiner slots using CAS to insert requests. When a software thread waits for the local combiner, it yields instead of spinning, so that the underlying hardware thread can run other software threads to generate larger combiner batches and increase efficiency.

**Log length.** NR uses a circular array for its log; if the array gets full, threads pause until older entries are consumed. This is undesirable, so one should use a large log, but how large? The solution is to dynamically resize the log if it gets full. This is done by writing a special log entry that indicates that the log has grown so that all replicas agree on the new size after consuming the special entry. This scheme gradually adjusts the log size until it is large enough.

**Memory allocation.** Memory allocation can become a performance bottleneck. We need an allocator that (1) avoids too much coordination across threads, and (2) allocates memory local to each node. We use a simple allocator in which threads get buffers from local pools. The allocator incurs coordination only if a buffer is allocated in one thread and freed in another; this requires returning the buffer to the allocating thread's pool. This is done in batches to reduce coordination.

**Inactive replica.** If threads in a node execute no operation on the data structure, the replica of that node stops replaying entries from the log, causing the log to fill up. This problem is solved by periodically running a thread per node that refreshes the local replica if the node has no operations to execute.

**Coupled data structures.** In some applications, data structures are read or updated together. For example, Redis implements sorted sets using a hash table and a skip list, which are updated atomically by each request. NR can provide these atomic updates, by treating the data structures as a single larger data structure with combined operations.

**Fake update operations.** Some update operations become readonly during execution (e.g., remove of a nonexistent key). Black-box methods must know about read-only operations at invocation time. If updates become read-only often, one can first attempt to execute them as read-only and, if not possible, then execute them as updates (e.g., remove(key) first tries to look up the key). This requires a simple wrapper around remove(). We did not implement this.

## 4. EVALUATION

We have evaluated NR to answer five broad questions: How does NR scale with the number of cores for different data structures and workloads? How does NR compare with other concurrent data structures? What is the benefit of NR to real applications? How does NR behave on different NUMA architectures? What are the benefits of NR's techniques? What are the costs of NR? Here, we highlight the most representative results and focus on the first three questions; the complete set of results are available in Calciu, Sen et al.[3] We report on two classes of experiments:

- *Real Data Structures (Section 4.1):* We run micro-benchmarks on real data structures: a skip list priority queue and a skip list dictionary.
- *Real Application (Section 4.2):* We run macro-benchmarks on the data structures of a real application: the Redis storage server modified to use many threads.

We compare NR against other methods (baselines) shown in Figure 4. Single Lock (SL) and Readers-Writer Lock (RWL) are methods often used in practice; they work by protecting the data structure with a SL or a single RWL. For RWL we use the same readers-writer lock as NR (Section 3.8). FC consists of flat combining used to implement the entire data structure. FC can be used as a black-box method, but it can also use data-structure-specific optimizations to combine operations for faster execution (hence its name); we use these optimizations whenever possible. FC+ is an improvement of FC by using a readers-writer lock to execute read-only operations more efficiently. Lock-Free Baseline (LF) is a lock-free algorithm specialized for a specific data structure; this baseline is available only for some data structures. In the real application (Redis), threads must atomically update multiple data structures but existing lock-free algorithms do not support that. LF requires a mechanism to garbage collect memory, such as hazard pointers[13] or epoch reclamation;[4] these mechanisms can reduce performance by 5x. We do not use these mechanisms, so the reported numbers for LF are better than in reality.

**Figure 4. Other methods for comparison (baselines).**

| Baseline | Description |
|---|---|
| SL | One big lock (spinlock) |
| RWL | One big readers-writer lock |
| FC | Flat combining |
| FC+ | Flat combining with readers-writer lock |
| LF | Lock-free algorithm |

**Summary of results.** On the real data structures (Section 4.1), we find that NR outperforms other methods at many threads under high operation contention, with the exception of NUMA-aware algorithms tailored to the data structure. The other methods, including lock-free algorithms, tend to lose significant performance beyond a NUMA node. We also find that NR consumes more memory than other methods. On a real application's data structures (Section 4.2), NR outperforms alternatives by 2.6x–14x on workloads with 10% updates, or by 1.1x–4.4x on 100% updates.

**Testbed.** We use a Dell server with 512GB RAM and 56 cores on four Intel Xeon E7-4850v3 processors at 2.2GHz. Each processor is a NUMA node with 14 cores, a 35MB shared L3 cache, and a private L2/L1 cache of size 256KB/64KB per core. Each core has 2 hyper-threads for a total of 112 hyper-threads. Cache lines have 64B.

### 4.1. Real data structures

These experiments use two real data structures: a skip list priority queue and a skip list dictionary. (Additional results using two other data structures are given in Calciu, Sen et al.,[3] but these results are qualitatively similar to the ones we present here.) A priority queue provides two update operations and one read-only operation: *insert(i)* inserts element *i*, *deleteMin()* removes and returns the smallest element, and *findMin()* returns the smallest element without removing it. We implement these operations using a skip list to order the elements and keep the minimum at the beginning of the list. A dictionary provides operations to insert, lookup, and remove elements, and we use a skip list to provide the dictionary. NR, FC, and FC+ internally use the same single-threaded implementation of a skip list;[15] FC uses the flat combining implementation from Hendler et al.[5] For the LF, we use the skip-list-based priority queue and skip list dictionary from Herlihy and Shavit.[7]

We use the benchmark from the flat combining paper,[5] which runs a mix of generic *add*, *remove*, and *read* operations. We map these operations to each data structure as shown here.

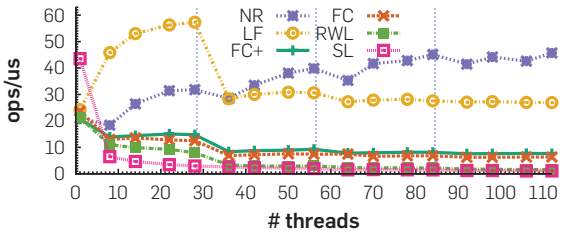
| generic | priority queue | dictionary |
|---|---|---|
| *add* | *insert*(*rnd*, *v*) | *insert*(*rnd*, *v*) |
| *remove* | *deleteMin*() | *delete*(*rnd*) |
| *read* | *findMin*() | *lookup*(*rnd*) |

Here, *rnd* indicates a key chosen at random and *v* is an arbitrary value. We use the same ratio of *add* and *remove* to keep the data structure size approximately constant over time, and the results aggregate these two operations as "update operations." We consider two ratios of update-to-read operations: 10%, 100% updates (90%, 0% reads). For the priority queue, we choose random keys from a uniform distribution. For the dictionary, we vary the operation contention by drawing the keys from two distributions: uniform (low contention) and zipf with parameter 1.5 (high contention).

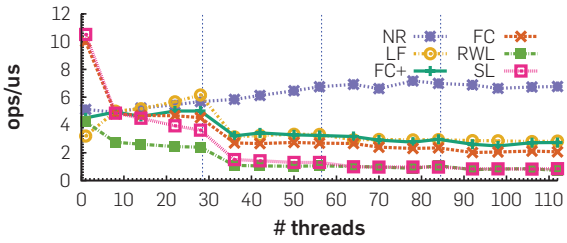
Between operations the benchmark optionally does work by writing $e$ random locations external to the data structure. This work causes cache pollution and reduces the arrival rate of operations. We first populate the data structure with 200,000 items, and then measure the performance of the methods for various workload mixes. In each experiment, we fix a method, a ratio of update-to-read operations, an external work amount $e$, and a number of threads.

**Results for priority queue.** For the priority queue, we see the following results (see Figure 5).

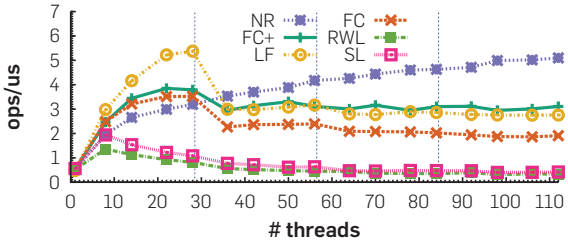
**Figure 5. Performance of priority queue made concurrent using different methods. Vertical lines show the boundaries between NUMA nodes.**



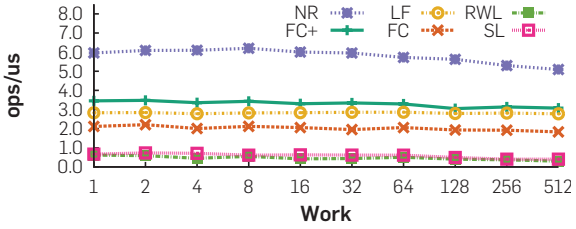

*(a)* 10% *update rate, e*=0




*(b)* 100% *update rate, e*=0




*(c)* 100% *update rate, e*=512




*(d)* 100% *update rate, max threads*

(e)

| | NR | others |
|---|---|---|
| **memory at max threads (MB)** | 148 | 34 |

(a) For 10% updates, all methods drop in performance at the NUMA node boundaries due to the cross-node overheads; but NR drops little, making it the best after one NUMA node. At max threads, NR is better than LF, FC+, FC, RWL, SL by 1.7x, 6x, 7x, 27x, 41x. Checking the CPU performance counters, NR had the fewest L3 cache misses and L3 cache misses served from remote caches, indicating lower cross-node traffic.

(b) For 100% updates, LF loses its advantage due to higher operation contention: even within a NUMA node, NR is close to LF. After one node, NR is best as before. At max threads, NR is better than LF, FC+, FC, SL, RWL by 2.4x, 2.5x, 3.3x, 8x, 9.4x.

(c) In some methods, one thread outperforms many threads, but not when there is work outside the data structure, as in many real applications. In such applications, we need more threads to scale the application and we want the shared data structure to not become a bottleneck.

(d) Node Replication remains the best method even as we vary the amount of external work $e$ and cache pollution. With $e$=512, NR is better than FC+, LF, FC, SL, RWL by 1.7x, 1.8x, 2.8x, 12.6x, 16.9x.

(e) The cost of NR is that it consumes more memory, namely, 148MB of memory at 112 threads (4.4x the other methods): 12MB for the log and 34MB for each of the four replicas. Technically, NR has another cost: it executes an operation many times, one per replica. However, this cost is relatively small as NR makes up for it with better overall performance.

**Results for dictionary.** For the dictionary, we see the following results (see Figure 6). When there are updates, performance depends on the level of contention. With low contention (uniform keys), LF outperforms other methods (it is off the charts): at maximum threads, it is 7x and 14x better than NR for 10% and 100% updates, respectively. This is due to the parallelism of the skip list unhindered by contention. Excluding LF, NR outperforms the other methods (with 100% updates, it does so after threads grow beyond a node).

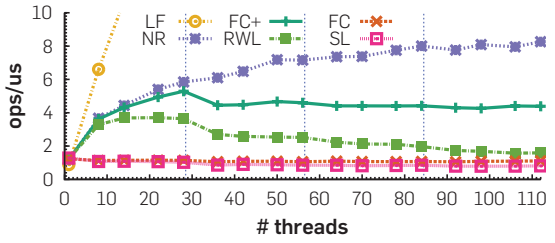
However, with high contention (zipf keys), LF loses its benefit, becoming the worst method for 100% updates. There is a high probability of collisions in the vicinity of the hot keys and the skip list starts to suffer from many failed CASs: with uniform keys, the skip list has ≈300$K$ failed CASs, but with the zipf keys this number increases to >7$M$. NR is the best method after 8 threads. Contention in the data structure does not disrupt the NR log. On the contrary, data structure contention improves cache locality with NR. With maximum threads and 10% updates, NR is better than LF, FC+, FC, RWL, SL by 3.1x, 4.0x, 6.8x, 16x, 30x. With 100% updates, NR is better by 2.8x, 1.8x, 2.4x, 5.7x, 4.3x.

## 4.2. Redis

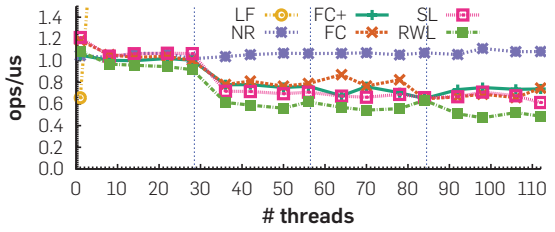
We now consider the data structures of the Redis server, made concurrent using various black-box methods, including NR.

We evaluate the sorted set data structure in Redis, which sorts items based on a score. In Redis, sorted sets are

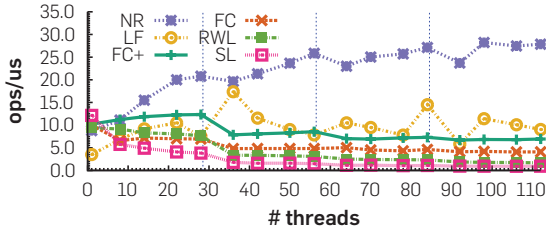
**Figure 6. Performance of skip list dictionary.**



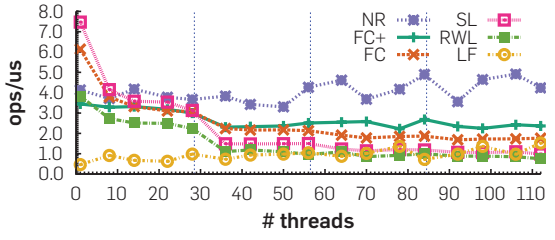

*(a) uniform keys, 10% update rate*




*(b) uniform keys, 100% update rate*




*(c) zipf keys, 10% update rate*

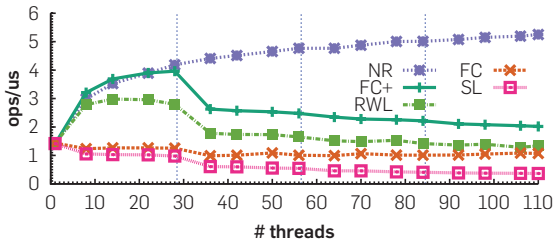



*(d) zipf keys, 100% update rate*

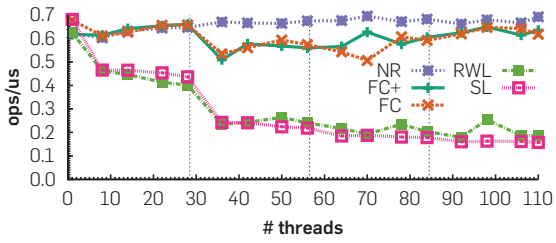
(e)

| | NR | others |
|---|---|---|
| memory at max threads (MB) | 148 | 34 |

**Figure 7. Performance of Redis application.**




*(a) 10% update rate*




*(b) 100% update rate*

implemented by a composed data structure that combines a hash table (for fast lookup) and a skip list (for fast rank/range queries). Every element in the sorted set is kept in both hash table and skip list. These underlying data structures must be updated atomically without the possibility that a user observes an update reflected in the hash table without it being reflected in the skip list, and vice versa.

For read operations, we use the ZRANK command, which returns the rank of an item in the sorted order. ZRANK finds the item in the hash table; if present, it finds its rank in the skip list. For update operations we use ZINCRBY, which increases the score of an item by a chosen value. ZINCRBY finds the item in the hash table; if present, it updates its score, and deletes and reinserts it into the skip list.

We used the redis-benchmark utility provided in the distribution to generate client load. We modified the benchmark to support hybrid read/write workloads using the update-read mix of the YCSB benchmark (0%, 10% updates) in addition to 100% updates.

To overcome the significant overheads of the Redis RPC and approximate a high-performance RPC,[10] we invoke Redis's operations directly at the server after the RPC layer, instead of generating requests from remote clients.

In each experiment, we create a single sorted set with 10,000 items. We launch multiple threads that repeatedly read or update a uniformly distributed random item using ZRANK or ZINCRBY, respectively. In each experiment, we fix an update ratio, a method, and a number of cores, and we measure the aggregate throughput.

**Results.** We see the following results (see Figure 7). For 10% updates, we see that all methods except NR drop after threads grow beyond a single node, making NR the best method for maximum threads. NR is better than FC+, RWL, FC, SL by 2.6x, 3.9x, 4.9x, 14x, respectively. For 100% updates, NR is better by 1.1x, 3.7x, 1.1x, 4.4x, respectively. For 0% updates, RWL, NR and FC+ scale well and have almost identical performance, while FC and SL do not scale (the graph is omitted).

While its scalability is not perfect, NR is the best method here. As discussed, the goal is to reduce data structure bottlenecks so that the rest of the application benefits from adding cores.

## 5. CONCLUSION

Node Replication is a general black-box method to transform single-threaded data structures into NUMA-aware concurrent data structures. Lock-free data structures are considered state-of-the-art, but they were designed for UMA. Creating new lock-free algorithms for NUMA is a herculean effort, as each data structure requires highly specialized new techniques. NR also required comparable effort, but once

realized, it can be used to provide all data structures with no extra work. With such a black-box method, we design for the architecture (in this case, NUMA) rather than for a data structure. We believe the community should investigate this black-box approach for future new architectures. C


**References**

1. Balakrishnan, M., Malkhi, D., Davis, J. P., Prabhakaran, V., Wei, M., Wobber, T. CORFU: a distributed shared log. *ACM Trans. Comp. Syst. 31*, 4 (Dec. 2013).
2. Calciu, I., Gottschlich, J.E., Herlihy, M. Using delegation and elimination to implement a scalable NUMA-friendly stack. In *USENIX Workshop on Hot Topics in Parallelism* (June 2013).
3. Calciu, I., Sen, S., Balakrishnan, M., Aguilera, M.K. Black-box concurrent data structures for NUMA architectures. In *International Conference on Architectural Support for Programming Languages and Operating Systems* (Apr. 2017), 207–221.
4. Fraser, K. Practical lock-freedom. Technical Report UCAM-CL-TR-579, University of Cambridge, Computer Laboratory (Feb. 2004).
5. Hendler, D., Incze, I., Shavit, N., Tzafrir, M. Flat combining and the synchronization-parallelism tradeoff. In *ACM Symposium on Parallelism in Algorithms and Architectures* (June 2010), 355–364.
6. Herlihy, M. Wait-free synchronization. *ACM Trans. Program. Lang. Syst. 11*, 1 (Jan. 1991), 124–149.
7. Herlihy, M., Shavit, N. *The Art of Multiprocessor Programming.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.
8. Herlihy, M.P., Wing, J.M. Linearizability: a correctness condition for concurrent objects. *ACM Trans. Program. Lang. Syst. 12*, 3 (July 1990), 463–492.
9. Lameter, C. NUMA (non-uniform memory access): an overview. *ACM Queue 11*, 7 (July 2013).
10. Lim, H., Han, D., Andersen, D.G., Kaminsky, M. MICA: a holistic approach to fast in-memory key-value storage. In *Symposium on Networked Systems Design and Implementation* (Apr. 2014), 429–444.
11. Mao, Y., Kohler, E., Morris, R.T. Cache craftiness for fast multicore key-value storage. In *European Conference on Computer Systems* (Apr. 2012), 183–196.
12. Metreveli, Z., Zeldovich, N., Kaashoek, M.F. CPHash: a cache-partitioned hash table. In *ACM Symposium on Principles and Practice of Parallel Programming* (Feb. 2012), 319–320.
13. Michael, M.M. Hazard pointers: safe memory reclamation for lock-free objects. *IEEE Trans. Parallel Distrib. Syst. 15*, 6 (June 2004), 491–504.
14. Porobic, D., Liarou, E., Tözün, P., Ailamaki, A. ATraPos: adaptive transaction processing on hardware islands. In *International Conference on Data Engineering* (Mar. 2014), 688–699.
15. Pugh, W. Skip lists: a probabilistic alternative to balanced trees. *Commun. ACM 33*, 6 (June 1990), 668–676.
16. Shalev, O., Shavit, N. Predictive log-synchronization. In *European Conference on Computer Systems* (Apr. 2006), 305–316.



**Irina Calciu and Marcos K. Aguilera**, VMware Research Group, Palo Alto, CA, USA.

**Siddhartha Sen**, Microsoft Research, New York, NY, USA.

**Mahesh Balakrishnan**, Yale University , New Haven, CT, USA.

DOI:10.1145/3282508

To view the accompanying paper, visit doi.acm.org/10.1145/3282510 rh

# Technical Perspective
# WebAssembly: A Quiet Revolution of the Web

By Anders Møller

WHEN JAVASCRIPT WAS introduced in 1995, it was intended as a small scripting language for interacting with the HTML DOM. A typical use was validating user form input or making simple animations. For many years, JavaScript programs were mostly small, and the majority of the program code in Web applications was running on the servers, not in the browsers. This changed with the advent of fast JavaScript engines like V8, which enabled a new generation of Web applications executed mostly in the browsers to provide a better user experience.

Within the last decade, commonplace JavaScript programs have grown to many thousands of lines of code, and JavaScript is used far beyond what anyone had anticipated in 1995. Despite the ongoing evolution of the language, it has been stretched to its limits. This has led to people developing compilers from other languages to JavaScript (although JavaScript is horrible as compilation target), and to language extensions and specialized runtime support (in particular, asm.js). Still, JavaScript has maintained a remarkable monopoly, being the only programming language supported by all main browsers. Until now.

The following paper gives an overview of the initial design of WebAssembly, a new low-level programming language for Web-based software. The language has well-defined semantics that ensures program execution to be independent of the underlying hardware and operating system. Browsers routinely run code from untrusted sources, so safety is obviously also of paramount importance. For this reason, WebAssembly is equipped with a type system that prevents certain safety violations at runtime.

The paper explains the rationale behind the main design decisions, gives an overview of the language, and describes preliminary experiences from developing the implementations. No individual features are groundbreaking from a programming languages design point of view (for example, the authors refer to the type system as "embarrassingly simple"); the value is in the sum of the design choices.

Perhaps the biggest feat is that the WebAssembly team, initiated by Luke Wagner at Mozilla and Ben Titzer at Google Munich, has managed to unify the browser vendors and coordinate the design and implementation effort. (The reader may remember the "browser wars" where certain browser vendors deliberately undermined standardization efforts in an attempt to gain market shares.)

The paper gives an overview of the formalization, including the operational semantics and the type system. Quite remarkably, the designers have chosen to formally specify the language, which has contributed to the clean design. This approach also demonstrates the power of formal techniques and mechanized language semantics: the essential type safety properties now have machine-checked proofs, which guarantees a solid foundation for the running software.

**WebAssembly provides a powerful platform for running non-JavaScript code on the Web.**

WebAssembly is now supported by all the modern browsers, and it has been embraced by a wide range of software companies. The primary goal of WebAssembly is (currently) not to replace JavaScript, but to complement it by making it easier to develop computationally demanding Web applications, such as games, software for audio/video processing, virtual reality systems, and CAD tools, as well as to port desktop applications to the Web.

This is just the first step. The initial focus of the WebAssembly team has been on compilation from C/C++, and the first major milestone has been reached. The future work will likely concentrate on extending WebAssembly with support for parallel execution, and beyond that memory management with garbage collection, which will simplify compilation from many high-level programming languages, for example, Java, C#, Swift, and OCaml. Although specifically designed with browser-based execution in mind, despite the name, there is actually not much "Web"-specific in WebAssembly. Perhaps the "write once, run anywhere" slogan once used by Java will be resurrected with WebAssembly? C

**Anders Møller** (amoeller@cs.au.dk) is a professor in Department of Computer Science at Aarhus University, Denmark.

DOI:10.1145/3282510

# Bringing the Web Up to Speed with WebAssembly

By Andreas Rossberg, Ben L. Titzer, Andreas Haas, Derek L. Schuff, Dan Gohman, Luke Wagner, Alon Zakai, J.F. Bastien, and Michael Holman

**Abstract**

**The maturation of the Web platform has given rise to sophisticated Web applications such as 3D visualization, audio and video software, and games. With that, efficiency and security of code on the Web has become more important than ever. WebAssembly is a portable low-level bytecode that addresses these requirements by offering a compact representation, efficient validation and compilation, and safe execution with low to no overhead. It has recently been made available in all major browsers. Rather than committing to a specific programming model, WebAssembly is an abstraction over modern hardware, making it independent of language, hardware, and platform and applicable far beyond just the Web. WebAssembly is the first mainstream language that has been designed with a formal semantics from the start, finally utilizing formal methods that have matured in programming language research over the last four decades.**

## 1. INTRODUCTION

The Web began as a simple hypertext document network but has now become the most ubiquitous application platform ever, accessible across a vast array of operating systems and device types. By historical accident, JavaScript is the only natively supported programming language on the Web. Because of its ubiquity, rapid performance improvements in modern implementations, and perhaps through sheer necessity, it has become a compilation target for many other languages. Yet JavaScript has inconsistent performance and various other problems, especially as a compilation target.

WebAssembly (or "Wasm" for short) addresses the problem of safe, fast, portable low-level code on the Web. Previous attempts, from ActiveX to Native Client to asm.js, have fallen short of properties that such a low-level code format should have:

- Safe, fast, and portable *semantics*:
  - safe to execute
  - fast to execute
  - language-, hardware-, and platform-independent
  - deterministic and easy to reason about
  - simple interoperability with the Web platform
- Safe and efficient *representation*:
  - maximally compact
  - easy to decode, validate and compile
  - easy to generate for producers
  - streamable and parallelizable

Why are these goals important? Why are they hard?

**Safe.** Safety for mobile code is paramount on the Web, since code originates from untrusted sources. Protection for mobile code has traditionally been achieved by providing a managed language runtime such as the browser's JavaScript Virtual Machine (VM) or a language plugin. Managed languages enforce *memory safety*, preventing programs from compromising user data or system state. However, managed language runtimes have traditionally not offered much for low-level code, such as C/C++ applications that do not use garbage collection.

**Fast.** Low-level code like that emitted by a C/C++ compiler is typically optimized ahead-of-time. Native machine code, either written by hand or as the output of an optimizing compiler, can utilize the full performance of a machine. Managed runtimes and sandboxing techniques have typically imposed a steep performance overhead on low-level code.

**Universal.** There is a large and healthy diversity of programming paradigms, none of which should be privileged or penalized by a code format, beyond unavoidable hardware constraints. Most managed runtimes, however, have been designed to support a particular language or programming paradigm well while imposing significant cost on others.

**Portable.** The Web spans not only many device classes, but different machine architectures, operating systems, and browsers. Code targeting the Web must be hardware- and platform-independent to allow applications to run across all browser and hardware types with the same deterministic behavior. Previous solutions for low-level code were tied to a single architecture or have had other portability problems.

**Compact.** Code that is transmitted over the network should be small to reduce load times, save bandwidth, and improve overall responsiveness. Code on the Web is typically transmitted as JavaScript source, which is far less compact than a binary format, even when minified and compressed. Binary code formats are not always optimized for size either.

WebAssembly is the first solution for low-level code on the Web that delivers on all of the above design goals. It is the result of an unprecedented collaboration across all

[a] WebAssembly engines are not assumed to spend time on sophisticated optimizations, because producers usually can take care of that more cheaply offline. Hence WebAssembly does not magically make code faster. But it allows other languages to bypass the cost and complexity of JavaScript.

The original version of this paper was published in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation* (Barcelona, Spain, June 18–23, 2017), 185–200.

major browser vendors and an online community group to build a common solution for high-performance applications.[a]

While the Web is the primary motivation for WebAssembly, its design—despite the name—carefully avoids any dependencies on the Web. It is an open standard intended for embedding in a broad variety of environments, and other such embeddings are already being developed.

To our knowledge, WebAssembly also is the first industrial-strength language that has been designed with a formal semantics from the start. It not only demonstrates the "real world" feasibility of applying formal techniques, but also that they lead to a remarkably clean and simple design.

## 2. A TOUR OF THE LANGUAGE

Even though WebAssembly is a binary code format, we define it as a *programming language* with syntax and structure. As we will see, that makes it easier to explain and understand and moreover, allows us to apply well-established formal techniques for defining its semantics and for reasoning about it. Hence, Figure 1 presents WebAssembly in terms of a grammar for its *abstract syntax*.

### 2.1. Basics

Let us start by introducing a few unsurprising concepts before diving into less obvious ones in the following.

**Modules.** A WebAssembly binary takes the form of a *module*. It contains definitions for *functions*, *globals*, *tables*, and *memories*.[b] Definitions may be *exported* or *imported*.

While a module corresponds to the static representation of a program, a module's dynamic representation is an *instance*, complete with its mutable *state*. Instantiating a module requires providing definitions for all imports, which may be exports from previously created instances. Computations is initiated by invoking an exported function.

Modules provide both *encapsulation* and *sandboxing*: because a client can only access the exports of a module, other internals are protected from tampering; dually, a module can only interact with its environment through its imports which are provided by a client, so that the client has full control over the capabilities given to a module. Both these aspects are essential ingredients to the safety of WebAssembly.

**Functions.** The code in a module is organized into individual *functions*, taking parameters and returnng results as defined by its *function type*. Functions can call each other, including recursively, but are not first class and cannot be nested. The call stack for execution is not exposed, and thus cannot be directly accessed by a running WebAssembly program, even a buggy or malicious one.

**Instructions.** WebAssembly is conceptually based on a *stack machine*: code for a function consists of a sequence of *instructions* that manipulate values on an implicit *operand stack*. However, thanks to the type system (Section 3.2), the layout of the operand stack can be statically determined at any point in the code, so that implementations can compile the data flow between instructions directly without ever materializing the operand stack. The stack organization is merely a way to achieve a compact program representation, as it has been shown to be smaller than a register machine.

**Traps.** Some instructions may produce a *trap*, which immediately aborts the current computation. Traps cannot (currently) be handled by WebAssembly code, but an embedder will typically provide means to handle this condition, for example, by reifying them as JavaScript exceptions.

**Machine types.** WebAssembly has only four basic *value types t* to compute with. These are integers and IEEE 754 floating point numbers, each with 32 or 64 bits, as available in common hardware. Most WebAssembly instructions provide simple operators on these data types. The grammar in Figure 1 conveniently distinguishes categories such as *unary* and *binary* operators, *tests*, *comparisons*, and *conversions*. Like hardware, WebAssembly makes no distinction between signed and unsigned integer types. Instead, where it matters, a *sign extension* suffix _u or _s to an instruction selects either unsigned or two's complement signed behavior.

**Variables.** Functions can declare mutable *local variables*, which essentially provides an infinite set of zero-initialized virtual registers. A module may also declare typed *global variables* that can be either mutable or immutable and require an explicit initializer. Importing globals allows a limited

[b] WebAssembly's text format closely resembles this syntax. For brevity we omit minor features regarding initialization of modules.

**Figure 1. WebAssembly abstract syntax.**

(value types) $t$ ::= i32 | i64 | f32 | f64
(packed types) $pt$ ::= i8 | i16 | i32
(function types) $ft$ ::= $t^* \to t^*$
(global types) $gt$ ::= $\text{mut}^?\ t$

$unop_{iN}$ ::= **clz** | **ctz** | **popcnt**
$unop_{fN}$ ::= **neg** | **abs** | **ceil** | **floor** | **trunc** | **nearest** | **sqrt**
$binop_{iN}$ ::= **add** | **sub** | **mul** | **div**_$sx$ | **rem**_$sx$ | **and** | **or** | **xor** | **shl** | **shr**_$sx$ | **rotl** | **rotr**
$binop_{fN}$ ::= **add** | **sub** | **mul** | **div** | **min** | **max** | **copysign**
$testop_{iN}$ ::= **eqz**
$relop_{iN}$ ::= **eq** | **ne** | **lt**_$sx$ | **gt**_$sx$ | **le**_$sx$ | **ge**_$sx$
$relop_{fN}$ ::= **eq** | **ne** | **lt** | **gt** |**le** | **ge**
$cvtop$ ::= **convert** | **reinterpret**
$sx$ ::= **s** | **u**

(instructions) $e$ ::= **unreachable** | **nop** | **drop** | **select** | **block** $ft$ $e^*$ **end** | **loop** $ft$ $e^*$ **end** | **if** $ft$ $e^*$ **else** $e^*$ **end** | **br** $i$ | **br_if** $i$ | **br_table** $i^+$ | **return** | **call** $i$ | **call_indirect** $ft$ | **get_local** $i$ | **set_local** $i$ | **tee_local** $i$ | **get_global** $i$ | **set_global** $i$ | $t$.**load** $(pt\_sx)^?$ $a$ $o$ | $t$.**store** $pt^?$ $a$ $o$ | **memory.size** | **memory.grow** | $t$.**const** $c$ | $t$.$unop_t$ | $t$.$binop_t$ | $t$.$testop_t$ | $t$.$relop_t$ | $t$.$cvtop$ $t\_sx^?$

(functions) $func$ ::= $ex^*$ **func** $ft$ **local** $t^*$ $e^*$ | $ex^*$ **func** $ft$ $im$
(globals) $glob$ ::= $ex^*$ **global** $gt$ $e^*$ | $ex^*$ **global** $gt$ $im$
(tables) $tab$ ::= $ex^*$ **table** $n$ $i^*$ | $ex^*$ **table** $n$ $im$
(memories) $mem$ ::= $ex^*$ **memory** $n$ | $ex^*$ **memory** $n$ $im$
(imports) $im$ ::= **import** "*name*" "*name*"
(exports) $ex$ ::= **export** "*name*"
(modules) $mod$ ::= **module** $func^*$ $glob^*$ $tab^?$ $mem^?$

form of configurability, for example, for linking. Like all entities in WebAssembly, variables are referenced by integer indices.

So far so boring. In the following sections we turn our attention to more unusual features of WebAssembly's design.

### 2.2. Memory

The main storage of a WebAssembly program is a large array of raw bytes, the *linear memory* or simply *memory*. Memory is accessed with load and store instructions, where addresses are simply unsigned integer operands.

**Creation and growing.** Each module can define at most one memory, which may be shared with other instances via import/export. Memory is created with an initial size but may be dynamically grown. The unit of growth is a *page*, which is defined to be 64KiB, a choice that allows reusing virtual memory hardware for bounds checks on modern hardware (Section 5). Page size is fixed instead of being system-specific to prevent portability hazards.

**Endianness.** Programs that load and store to aliased locations with different types can observe byte order. Since most contemporary hardware has converged on little endian, or at least can handle it equally well, we chose to define WebAssembly memory to have little endian byte order. Thus the semantics of memory access is completely deterministic and portable across all engines and platforms.

**Security.** All memory access is dynamically checked against the memory size; out of bounds access results in a trap. Linear memory is disjoint from code space, the execution stack, and the engine's data structures; therefore compiled programs cannot corrupt their execution environment, jump to arbitrary locations, or perform other undefined behavior. At worst, a buggy or malicious WebAssembly program can make a mess of the data in its own memory. Consequently, even untrusted modules can be safely executed in the same address space as other code. Achieving fast in-process isolation is necessary for interacting with untrusted JavaScript and the various Web Application Programming Interfaces (APIs) in a high-performance way. It also allows a WebAssembly engine to be safely embedded into other managed language runtimes.

### 2.3. Control flow

WebAssembly represents control flow differently from most stack machines. It does not offer arbitrary jumps but instead provides *structured control flow* constructs more akin to a programming language. This ensures by construction that control flow cannot form irreducible loops, contain branches to blocks with misaligned stack heights, or branch into the middle of a multi-byte instruction. These properties allow WebAssembly code to be validated in a single pass, compiled in a single pass, and even transformed to an SSA-form intermediate representation in the same single pass.

**Control constructs.** As required by the grammar in Figure 1, the **block**, **loop** and **if** constructs must be terminated by an **end** opcode and be properly nested to be considered well-formed. The inner instruction sequences $e^*$ in these constructs form a *block*. Note that **loop** does not automatically iterate its block but allows constructing a loop manually with explicit branches. Every control construct is annotated with a function type $ft = t_1^* \to t_2^*$ describing its effect on the stack, popping values typed $t_1^*$ and pushing $t_2^*$.

**Branches.** Branches can be unconditional (**br**), conditional (**br_if**), or indexed (**br_table**). They have "label" immediates that do not denote positions in the instruction stream but reference outer control constructs by relative nesting depth. Hence, labels are effectively *scoped*: branches can only reference constructs in which they are nested. Taking a branch "breaks from" that construct's block;[c] the exact effect depends on the target construct: in case of a **block** or **if** it is a *forward* jump to its end (like a break statement); with a **loop** it is a *backward* jump to its beginning (like a continue statement). Branching *unwinds* the operand stack by implicitly popping all unused operands, similar to returning from a function call. This liberates producers from having to track stack height across sub-expressions and adding explicit drops to make them match.

**Expressiveness.** Structured control flow may seem like a severe limitation, but most high-level control constructs are readily expressible with the suitable nesting of blocks. For example, a C-style switch statement with fall-through,

```
switch (x) {                          block block block block
  case 0: ...A...                       br_table 0 1 2
  case 1: ...B... break;     becomes    end ...A...
  default: ...C...                      end ...B... br 1
}                                       end ...C...
                                      end
```

Slightly more finesse is required for fall-through between unordered cases. Various forms of loops can likewise be expressed with combinations of **loop**, **block**, **br** and **br_if**.

It is the responsibility of producers to transform unstructured and irreducible control flow into structured form. This is the established approach to compiling for the Web, where JavaScript is also restricted to structured control. In our experience building an LLVM backend for WebAssembly, irreducible control flow is rare, and a simple restructuring algorithm[18] is sufficient to translate any Control Flow Graph (CFG) to WebAssembly. The benefit of the restriction is that many algorithms in engines are much simpler and faster.

### 2.4. Function calls and tables

A function body is a block. Execution can complete by either reaching the end of the block with the function's result values on the stack, or by a branch exiting the function block, with the branch operands as the result values; the **return** instruction is simply shorthand for the latter.

**Calls.** Functions can be invoked *directly* using the **call** instruction which takes an immediate identifying the function to call. Function pointers can be emulated with the **call_indirect** instruction which takes a runtime index into a *table* of functions defined by the module. The functions in this table are not required to have the same type. Instead, the type of the function is checked dynamically against an

[c] The name **br** can also be read as "break" wrt. a block.

expected type supplied to the **call_indirect** instruction and traps in case of a mismatch. This check protects the integrity of the execution environment. The heterogeneous nature of the table is based on experience with asm.js's multiple homogeneous tables; it allows more faithful representation of function pointers and simplifies dynamic linking. To aid dynamic linking scenarios further, exported tables can be grown and mutated dynamically through external APIs.

**Foreign calls.** Functions can be imported into a module. Both direct and indirect calls can invoke an imported function, and through export/import, multiple module instances can communicate. Additionally, the import mechanism serves as a safe *Foreign Function Interface* (FFI) through which a WebAssembly program can communicate with its embedding environment. For example, on the Web imported functions may be *host* functions that are defined in JavaScript. Values crossing the language boundary are automatically converted according to JavaScript rules.

### 2.5. Determinism

WebAssembly has sought to provide a portable target for low-level code without sacrificing performance. Where hardware behavior differs it usually is corner cases such as out-of-range shifts, integer divide by zero, overflow or underflow in floating point conversion, and alignment. Our design gives deterministic semantics to all of these across all hardware with only minimal execution overhead.

However, there remain three sources of implementation-dependent behavior that can be viewed as non-determinism.

**NaN payloads.** WebAssembly follows the IEEE 754 standard for floating point arithmetic. However, IEEE does not specify the exact bit pattern for NaN values in all cases, and we found that CPUs differ significantly, while normalizing after every numeric operation is too expensive. Based on our experience with JavaScript engines, we picked rules that allow the necessary degree of freedom while still providing enough guarantees to support techniques like NaN-tagging.

**Resource exhaustion.** Available resources are always finite and differ wildly across devices. In particular, an engine may be *out of memory* when trying to grow the linear memory—semantically, the **memory.grow** instruction can non-deterministically return −1. A call instruction may also trap due to *stack overflow*, but this is not semantically observable from within WebAssembly itself since it aborts the computation.

**Host functions.** WebAssembly programs can call host functions which are themselves non-deterministic or change WebAssembly state. Naturally, the effect of calling host functions is outside the realm of WebAssembly's semantics.

WebAssembly does not (yet) have threads, and therefore no non-determinism arising from concurrent memory access. Adding threads is the subject of ongoing work.

### 2.6. Binary format

WebAssembly is transmitted as a binary encoding of the abstract syntax presented in Figure 1. This encoding has been designed to minimize both size and decoding time. A binary represents a single module and is divided into sections according to the different kinds of entities declared in it. Code for function bodies is deferred to a separate section placed after all declarations to enable *streaming compilation* as soon as function bodies begin arriving over the network. An engine can also *parallelize* compilation of function bodies. To aid this further, each body is preceded by its size so that a decoder can skip ahead and parallelize even its decoding.

The format also allows user-defined sections that may be ignored by an engine. For example, a custom section is used to store debug metadata such as source names in binaries.

## 3. SEMANTICS

The WebAssembly semantics consists of two parts: the *static semantics* defining *validation*, and the *dynamic semantics* defining *execution*. In both cases, the presentation as a language allowed us to adopt off-the-shelf formal methods developed in programming language research over the past decades. They are convenient and effective tools for declarative specifications. While we can not show all of it here, our specification is precise, concise, and comprehensive—validation and execution of all of WebAssembly fit on just two pages of our original paper.[4]

Furthermore, these formulations allow effective *proofs* of essential properties of this semantics, as are standard in programming language research, but so far have rarely ever been done as part of an industrial-strength design process.

Finally, our formalization enabled other researchers to easily *mechanize* the WebAssembly specification with theorem provers, thereby machine-verifying our correctness results as well as constructing a provably correct interpreter.

### 3.1. Execution

We cannot go into much detail in this article, but we want to give a sense for the general flavor of our formalization (see Haas et al.[4] for a more thorough explanation).

**Reduction.** Execution is defined in terms of a standard *small-step reduction* relation,[13] where each step of computation is described as a *rewrite rule* over a sequence of instructions. Figure 2 gives an excerpt of these rules.

For example, the instruction sequence

(**i32.const** 3) (**i32.const** 4) **i32.add**

is reduced to the constant (**i32.const** 7) according to the fourth rule. This formulation avoids the need for introducing the operand stack as a separate notion in the semantics—that stack simply consists of all leading *t*.**const** instructions in an instruction sequence. Execution terminates when an instruction sequence has been reduced to just constants, corresponding to the stack of result values. Therefore, constant instructions can be treated as *values* and abbreviated *v*.

To deal with control constructs, we need to squint a little and extend the syntax with a small number of auxiliary *administrative instructions* that only occur temporarily during reduction. For example, **label** marks the extent of an active block and records the continuation of a branch to it, while **frame** essentially is a call frame for function invocation. Through nesting these constructs, the intertwined nature of operand and control stack is captured, avoiding the need for separate stacks with tricky interrelated invariants.

**Figure 2. Small-step reduction rules (Excerpt).**

| | | | |
|---|---|---|---|
| (store) | $s$ | ::= | {**func** $fi^*$, **global** $v^*$, **table** $ti^*$, **mem** $mi^*$} |
| (frames) | $f$ | ::= | {**module** $m$, **local** $v^*$} |
| (module instances) | $m$ | ::= | {**func** $a^*$, **global** $a^*$, **table** $a^?$, **mem** $a^?$} |
| (function instances) | $fi$ | ::= | {**module** $m$, **code** $func$} (where $func$ is not an import and has all exports $ex^*$ erased) |
| (table instances) | $ti$ | ::= | $(a^?)^*$ |
| (memory instances) | $mi$ | ::= | $b^*$ |
| (values) | $v$ | ::= | $t$.**const** $c$ |
| (administrative operators) | $e$ | ::= | ... \| **trap** \| **call** $fi$ \| **label**$_n${$e^*$} $e^*$ **end** \| **frame**$_n${$f$} $e^*$ **end** |

$$
\begin{array}{rcll}
\textbf{nop} & \hookrightarrow & \epsilon & \\
v\ \textbf{drop} & \hookrightarrow & \epsilon & \\
(t.\textbf{const}\ c)\ t.unop & \hookrightarrow & t.\textbf{const}\ unop_t(c) & \\
(t.\textbf{const}_{c_1})\ (t.\textbf{const}_{c_2})\ t.binop & \hookrightarrow & t.\textbf{const}\ binop_t(c_1, c_2) & \\
v^n\ \textbf{block}\ (t_1^n \to t_2^m)\ e^*\ \textbf{end} & \hookrightarrow & \textbf{label}_m\{\epsilon\}\ v^n\ e^*\ \textbf{end} & \\
v^n\ \textbf{loop}\ (t_1^n \to t_2^m)\ e^*\ \textbf{end} & \hookrightarrow & \textbf{label}_n\{\textbf{loop}(t_1^n \to t_2^m)\ e^*\ \textbf{end}\}\ v^n\ e^*\ \textbf{end} & \\
\textbf{label}_n\{e^*\}\ v^*\ \textbf{end} & \hookrightarrow & v^* & \\
\textbf{label}_n\{e^*\}\ L^i[v^n\ (\textbf{br}\ i)]\ \textbf{end} & \hookrightarrow & v^n\ e^* & \text{where } L^0 ::= v^*[\_]\ e^* \text{ and } L^{k+1} ::= v^*\ \textbf{label}_n\{e^*\}\ L^k\ \textbf{end}\ e^* \\
f;(\textbf{get\_local}\ i) & \hookrightarrow & v & \text{if } f_{\text{local}}(i) = v \\
f;v\ (\textbf{set\_local}\ i) & \hookrightarrow & f';\epsilon & \text{if } f' = f \text{ with } \text{local}(i) = v \\
s;f;(\textbf{call}\ i) & \hookrightarrow & \textbf{call}\ s_{\text{func}}(f_{\text{func}}(i)) & \\
s;f;(\textbf{i32.const}\ i)\ (\textbf{call\_indirect}\ ft) & \hookrightarrow & \textbf{call}\ s_{\text{func}}(s_{\text{table}}(f_{\text{table}}(i)) & \text{if } s_{\text{func}}(s_{\text{table}}(f_{\text{table}})(i))_{\text{code}} = (\textbf{func}\ ft\ \textbf{local}\ t^*\ e^*) \\
s;f;(\textbf{i32.const}\ i)\ (\textbf{call\_indirect}\ ft) & \hookrightarrow & \textbf{trap} & \text{otherwise} \\
v^n\ (\textbf{call}\ fi) & \hookrightarrow & \textbf{frame}_m\{\text{module } fi_{\text{module}}, \text{local } v^n\ (t.\textbf{const}\ 0)^k\}\ e^*\ \textbf{end} \ldots & \\
\textbf{frame}_n\{f\}\ v^n\ \textbf{end} & \hookrightarrow & v^n & \mid \ldots \text{where } fi_{\text{code}} = (\textbf{func}\ (t_1^n \to t_2^m)\ \textbf{local}\ t^k\ e^*) \\
\textbf{frame}_n\{f\}\ L^k[v^n\ \textbf{return}]\ \textbf{end} & \hookrightarrow & v^n & \\
s;f_0;\ \textbf{frame}_n\{f\}\ e^*\ \textbf{end} & \hookrightarrow & s';f_0;\ \textbf{frame}_n\{f'\}\ e'\ \textbf{end} & \text{if } s;f;e^* \hookrightarrow s';f';e'^*
\end{array}
$$

**Configurations.** In general, execution operates relative to a global *store* $s$ as well as the current function's *frame* $f$. Both are defined in the upper part of Figure 2.

The store models the global state of a program and is a record of the lists of function, global, table and memory *instances* that have been allocated. An index into one of the store components is called an *address* $a$. We use notation like $s_{\text{func}}(a)$ to look up the function at address $a$. A module instance then maps static indices $i$ that occur in instructions to their respective dynamic addresses in the store.

To that end, each frame carries—besides the state of the function's local variables—a link to the module instance it resides in, applying notational short-hands like $f_{\text{table}}$ for $(f_{\text{module}})_{\text{table}}$. Essentially, every function instance is a *closure* over the function's module instance. An implementation can eliminate these closures by specializing generated machine code to a module instance.

For example, for a direct **call** $i$, the respective function instance is looked up in the store through the frame of the caller. Similarly, for an indirect call, the current table is looked up and the callee's address is taken from there (if the function's type $ft$ does not match the expected type, then a trap is generated). Both kinds of calls reduce to a common administrative instruction **call** $fi$ performing a call to a known function instance; reducing that further creates a respective frame for the callee and initializes its locals.

Together, the triple $s;f;e^*$ of store, frame, and instruction sequence forms a *configuration* that represents the complete state of the WebAssembly abstract machine at a given point in time. Reduction rules, in their full generality, then rewrite configurations not just instruction sequences.

## 3.2. Validation

On the Web, code is fetched from untrusted sources and must be *validated*. Validation rules for WebAssembly are defined succinctly as a *type system*. This type system is, by design, embarrassingly simple, and designed to be efficiently checkable in a single linear pass.

**Typing rules.** Again, we utilize standard techniques for defining our semantics declaratively, this time via a system of *natural deduction* rules.[12] Figure 3 shows an excerpt of the rules for typing instructions. They collectively define a *judgement* $C \vdash e : ft$, which can be read as "instruction $e$ is valid with type $ft$ under the assumptions embodied in the *context* $C$." The context records the types of all declarations that are in scope at a given point in a program. The type of an instruction is a function type that specifies its required *input* stack and the provided *output* stack.

Each rule consists of a conclusion (the part below the bar) and a possibly empty list of premises (the pieces above the bar). It can be read as a big implication: the conclusion holds if all premises hold. One rule exists for each instruction, defining when it is well-typed. A program is valid if and only if the rules can inductively derive that it is well-typed.

For example, the rules for constants and simple numeric operators are trivial *axioms*, since they do not even require a premise: an instruction of the form $t.binop$ always has type $t\ t \to t$, that is, consumes two operands of type $t$ and pushes one. The rules for control constructs require that their type matches the explicit annotation $ft$, and they extend the context with a local label when checking the inner block. Label types are looked up in the context when typing branch

**Figure 3. Typing rules (Excerpt).**

$$\text{(contexts)}\quad C \;::=\; \{\text{func } ft^*,\ \text{global } gt^*,\ \text{table } n^?,\ \text{memory } n^?,\ \text{local } t^*,\ \text{label } (t^*)^*,\ \text{return } (t^*)^?\}$$

$$\frac{}{C \vdash t.\mathbf{const}\ c : \epsilon \to t} \qquad \frac{}{C \vdash t.unop : t \to t} \qquad \frac{}{C \vdash t.binop : t\ t \to t} \qquad \frac{}{C \vdash \mathbf{nop} : \epsilon \to \epsilon} \qquad \frac{}{C \vdash \mathbf{drop} : t \to \epsilon}$$

$$\frac{ft = t_1^n \to t_2^m \qquad C,\ \text{label}(t_2^m) \vdash e^* :\ ft}{C \vdash \mathbf{block}\ ft\ e^*\ \mathbf{end} :\ ft} \qquad \frac{ft = t_1^n \to t_2^m \qquad C, \text{label}(t_1^n)\ \vdash e^* : ft}{C \vdash \mathbf{loop}\ ft\ e^*\ \mathbf{end} : ft} \qquad \frac{C_{\text{label}}(i) = t^*}{C \vdash \mathbf{br}\ i : t_1^*\ t^* \to t_2^*}$$

$$\frac{C_{\text{func}}(i) = ft}{C \vdash \mathbf{call}\ i : ft} \qquad \frac{ft = t_1^* \to t_2^* \qquad C_{\text{table}} = n}{C \vdash \mathbf{call\_indirect}\ ft : t_1^*\ \text{i32} \to t_2^*} \qquad \frac{C_{\text{return}} = t^*}{C \vdash \mathbf{return} : t_1^*\ t^* \to t_2^*}$$

$$\frac{C_{\text{local}}(i) = t}{C \vdash \mathbf{get\_local}\ i : \epsilon \to t} \qquad \frac{C_{\text{local}}(i) = t}{C \vdash \mathbf{set\_local}\ i : t \to \epsilon}$$

instructions, which require suitable operands on the stack to match the stack at the join point.

### 3.3. Soundness

The WebAssembly type system enjoys standard *soundness* properties.[16] Soundness proves that the reduction rules actually cover all execution states that can arise for valid programs. In other words, it proves the absence of undefined behavior. In particular, this implies the absence of *type safety* violations such as invalid calls or illegal accesses to locals, it guarantees *memory safety*, and it ensures the inaccessibility of code addresses or the call stack. It also implies that the use of the operand stack is structured and its layout determined statically at all program points, which is crucial for efficient compilation on a register machine. Furthermore, it establishes memory and state *encapsulation*—that is, abstraction properties on the module and function boundaries, which cannot leak information.

Given our formal definition of the language, soundness can be made precise as a fairly simple theorem:

THEOREM 3.1. (SOUNDNESS). *If* $\vdash s; f; e^* : t^n$ *(i.e., configuration* $s; f; e^*$ *is valid with resulting stack type* $t^n$*), then:*

- *either* $s; f; e^* \hookrightarrow^* s'; f'; (t.\mathbf{const}\ c)^n$ *(i.e., after a finite number of steps the instruction sequence has been reduced to values of the correct types),*
- *or* $s; f; e^* \hookrightarrow^* s'; f'; \mathbf{trap}$ *(i.e., execution traps after a finite number of steps),*
- *or execution diverges (i.e., there is an infinite sequence of reduction steps it can take).*

This formulation uses a typing judgement generalized to configurations whose definition we omit here. The property ensures that all valid programs either diverge, trap, or terminate with values of the expected types. The proofs are completely standard (almost boring) induction proofs like can be found in many text books or papers on the subject.

### 3.4. Mechanization

For our paper, we have done the soundness proofs by hand, on paper. We also have implemented a WebAssembly reference interpreter in OCaml that consists of a direct transliteration of the formal rules into executable code (Section 4.1). While both tasks were largely straightforward, they are always subject to subtle errors not uncovered by tests.

Fortunately, over the last 15 years, methodologies for *mechanizing* language semantics and their meta-theory in theorem provers have made significant advances. Because our formalization uses well-established techniques, other researchers have been able to apply mechanization to it immediately. As a result, there are multiple projects for mechanizing the semantics—and in fact, the full language definition (Section 4)—in three major theorem provers and semantics tools, namely Isabelle, Coq, and K. Their motivation is in both verifying WebAssembly itself as well as providing a foundation for other formal methods applications, such as verifying compilers targeting WebAssembly or proving properties of programs, program equivalences, and security properties.

The Isabelle mechanization was completed first and has already been published.[14] It not only contains a machine-verified version of our soundness proof, it also includes a machine-verified version of a validator and interpreter for WebAssembly that other implementations and our reference interpreter can be compared against. Moreover, in the process of mechanizing the soundness proof this work uncovered a few minor bugs in the draft version of our formalization and enabled us to fix it in a timely manner for publication.

## 4. STANDARDIZATION

The previous section reflects the formalization of WebAssembly as published in our Programming Language Design and Implementation (PLDI) paper[4] with a few minor stylistic modifications. However, our reason for developing this semantics was more than producing a paper targeted at researchers—it was meant to be the basis of the official language definition.[15] This definition, which is currently under review as a standard for the W3C, contains the complete formalization of the language.

### 4.1. Core language

The language definition follows the formalization and specifies abstract syntax, typing rules, reduction rules, and an abstract store. Binary format and text format are given as attribute grammars exactly describing the abstract syntax they produce. As far as we are aware, this level of rigor and precision is unprecedented for industrial-strength languages.

**Formalism.** Although the formal methods we use are standard in academic literature and computer science (CS)

curricula, a widely consumed standard cannot (yet) assume that all its readers are familiar with formal notation for semantics (unlike for syntax). Next to the formal rules, the specification hence also contains corresponding prose. This prose is intended to be a one-to-one "text rendering" of the formal rules. Although the prose follows the highly verbose "pseudo-COBOL" style of other language definitions, its eyeball proximity to a verified formalism aids spotting bugs. Having the formal rules featured centrally in the standard document hence benefits even readers that do not read them directly.

**Reference interpreter.** Along with the formalization and production implementations in browsers, we developed a reference interpreter for WebAssembly. For this we used OCaml due to the ability to write in a high-level stylized fashion that closely matches the formalization, approximating an "executable specification." The interpreter is used to develop the test suite, test production implementations and the formal specification, and to prototype new features.

**Proposal process.** To maintain the current level of rigor while evolving WebAssembly further, we have adopted a multi-staged proposal process with strong requirements. At various stages of a proposal, its champions must provide (1) an informal description, (2) a prose specification, (3) a prototype implementation, (4) a comprehensive test suite, (5) a formal specification, (6) an implementation in the reference interpreter, and (7) two implementations in independent production systems.

The process is public on the working group's Git repository, where specification, reference interpreter, and test suite are hosted. Creating a proposal involves asking the group to create a fork of the main "spec" repository and then iterating and reviewing all required additions there.

Obviously, a formal semantics is not straightforward in all cases. Where necessary, the working group is collaborating with research groups for non-trivial features, such as a suitable weak memory model for the addition of threads.

### 4.2. Embedding

WebAssembly is similar to a virtual Instruction Set Architecture (ISA) in that it does not define how programs are loaded into the execution engine or how they perform I/O. This intentional design separation is captured in the notion of *embedding* a WebAssembly implementation into an execution environment. The embedder defines how modules are loaded, how imports and exports are resolved, how traps are handled, and provides foreign functions for accessing the environment.

To strengthen platform-independence and encourage other embeddings of WebAssembly, the standard has been layered into separate documents: while the core specification only defines the virtual ISA, separate *embedder specifications* define its interaction with concrete host environments.

**JavaScript and the web.** In a browser, WebAssembly modules can be loaded, compiled and invoked through a JavaScript API. The rough recipe is to (1) acquire a binary module from a given source, for example, as a network resource, (2) instantiate it providing the necessary imports, and (3) call the desired export functions. Since compilation and instantiation may be slow, they are provided as asynchronous methods whose results are wrapped in promises. The JavaScript API also allows creating and initializing memories or tables externally, or accessing them as exports.

**Interoperability.** It is possible to link multiple modules that have been created by different producers. However, as a low-level language, WebAssembly does not provide any built-in object model. It is up to producers to map their data types to memory. This design provides maximum flexibility to producers, and unlike previous VMs, does not privilege any specific programming paradigm or object model.

Interested producers can define common ABIs *on top of* WebAssembly such that modules can interoperate in heterogeneous applications. This separation of concerns is vital for making WebAssembly universal as a code format.

## 5. IMPLEMENTATION

A major design goal of WebAssembly has been high performance without sacrificing safety or portability. Throughout its design process, we have developed independent implementations of WebAssembly in all major browsers to validate and inform the design decisions. This section describes some points of interest of those implementations.

**Implementation strategies.** V8 (Chrome), SpiderMonkey (Firefox) and JavaScriptCore (WebKit) reuse their optimizing JavaScript compilers to compile WebAssembly modules ahead-of-time. This achieves predictable high performance and avoids the unpredictability of warmup time which has often been a problem for JavaScript. Chakra (Edge) instead lazily translates individual functions to an interpreted internal bytecode format upon first execution, and later Just In Time (JIT)-compiles the hottest functions. The advantage is faster startup and potentially lower memory consumption. We expect more strategies to evolve over time.

**Validation.** In the four aforementioned implementations, the same algorithmic strategy using abstract control and operand stacks is used. Validation of incoming bytecodes occurs in a single pass during decoding, requiring no additional intermediate representation. We measured single-threaded validation speed at between 75MB/s and 150MB/s on a suite of representative benchmarks on a modern workstation. This is approximately fast enough to perform validation at full network speed.

**Baseline JIT compiler.** The SpiderMonkey engine includes two WebAssembly compilation tiers. The first is a fast baseline JIT that emits machine code in a single pass combined with validation. The JIT creates no Intermediate Representation (IR) but does track register state and attempts to do simple greedy register allocation in the forward pass. The baseline JIT is designed only for fast startup while an optimizing JIT is compiling the module in parallel in the background. V8 includes a similar baseline JIT in a prototype configuration.

**Optimizing JIT compiler.** All four engines include optimizing JITs for their top-tier execution of JavaScript and reuse them for WebAssembly. Both V8 and SpiderMonkey use SSA-based intermediate representations. As such, it was important that WebAssembly can be decoded to SSA form in a single pass. This is greatly helped by WebAssembly's structured control flow, making the decoding algorithm simpler

and more efficient and avoiding the limitation of JITs that usually do not support irreducible control flow. Reusing the advanced JITs from four different JavaScript engines has been a resounding success that allowed all engines to achieve high performance in a short time.

**Bounds checks.** By design, all memory accesses in WebAssembly can be guaranteed safe with a single dynamic bounds check, which amounts to checking the address against the current size of the memory. An engine will allocate the memory in a large contiguous range beginning at some (possibly non-deterministic) *base* in the engine's process, so that all access amounts to a hardware address *base+addr*. While *base* can be stored in a dedicated machine register for quick access, a more aggressive strategy is to *specialize* the machine code for each instance to a specific base, embedding it as a constant directly into the code, freeing a register. Although the base may change when the memory is grown dynamically, it changes so infrequently that it is affordable to *patch* the machine code when it does.

On 64 bit platforms, an engine can make use of virtual memory to eliminate bounds checks for memory accesses altogether. The engine simply reserves 8GB of virtual address space and marks as inaccessible all pages except the valid portion of memory near the start. Since WebAssembly memory addresses and offsets are 32 bit integers plus a static constant, by construction no access can be further than 8GB away from *base*. Consequently, the JIT can simply emit plain load/store instructions and rely on hardware protection mechanisms to catch out-of-bounds accesses.

**Parallel and streaming compilation.** With ahead-of-time compilation it is a clear performance win to parallelize compilation of WebAssembly modules, dispatching individual functions to different threads. For example, both V8 and SpiderMonkey achieve a 5-6× improvement in compilation speed with eight compilation threads. In addition, the design of the WebAssembly binary format supports *streaming* where an engine can start compilation of individual functions before the full binary has been loaded. When combined with parallelization, this minimizes cold startup.

**Code caching** Besides cold startup, warm startup time is important as users will likely visit the same Web pages repeatedly. The JavaScript API for the IndexedDB database allows JavaScript to manipulate and compile WebAssembly modules and store their compiled representation as an opaque blob. This allows a JavaScript application to first query IndexedDB for a cached version of their WebAssembly module before downloading and compiling it. In V8 and SpiderMonkey, this mechanism can offer an order of magnitude improvement of warm startup time.

### 5.1. Measurements

**Execution.** Figure 4 shows the execution time of the PolyBenchC benchmark suite running on WebAssembly on both V8 and SpiderMonkey normalized to native execution.[d] Times for both engines are shown as stacked bars, and the results show that there are still some differences between them due to different code generators.[e] We measured a VM startup time of 18 ms for V8 and 30ms for SpiderMonkey. These times are included along with compilation times as bars stacked on top of the execution time of each benchmark. Overall, the results show that WebAssembly is competitive with native code, with seven benchmarks within 10% of native and nearly all of them within 2× of native.

We also measured the execution time of the PolyBenchC benchmarks running on asm.js. On average, WebAssembly is 33.7% faster than asm.js. Especially validation is significantly more efficient. For SpiderMonkey, WebAssembly validation takes less than 3% of the time of asm.js validation. In V8, memory consumption of WebAssembly validation is less than 1% of that for asm.js validation.

**Code size.** Figure 5 compares code sizes between WebAssembly, minified asm.js, and ×86-64 native code. For the asm.js comparison we use the Unity benchmarks, for the native code comparison the PolyBenchC and SciMark benchmarks. For each function in these benchmarks, a yellow point is plotted at $\langle size_{asmjs}, size_{wasm} \rangle$ and a blue point at $\langle size_{x86}, size_{wasm} \rangle$. Any point below the diagonal represents code for which WebAssembly is smaller than the corresponding other representation. On average, WebAssembly code is 62.5% the size of asm.js, and 85.3% of native ×86-64 code.

## 6. RELATED WORK

Microsoft's ActiveX was a technology for code-signing ×86

**Figure 4. Relative execution time of the Poly-BenchC benchmarks on WebAssembly normalized to native code.**

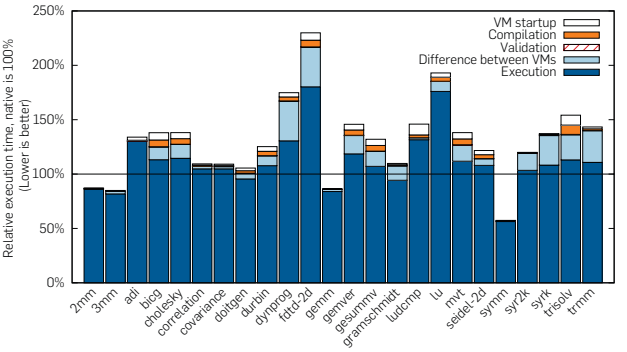
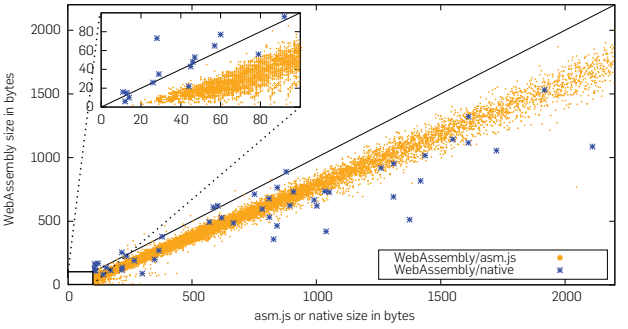



**Figure 5. Binary size of WebAssembly in comparison to asm.js and native code.**




[d] See Haas et al.[4] for details on the experimental setup for these measurements.
[e] V8 is faster on some benchmarks and SpiderMonkey on others. Neither engine is universally faster than the other.

binaries to run on the Web. It relied entirely upon trust and thus did not achieve safety through technical construction.

Native Client[17] was the first system to introduce a sand-boxing technique for machine code on the Web that runs at near native speed. It relies on static validation of ×86 machine code, requiring code generators to follow certain patterns, such as bit masks before memory accesses and jumps. Instead of hardware-specific ×86 code, Portable Native Client (PNaCl) uses a stable subset of LLVM bitcode[6] as an interchange format.

Emscripten[18] is a framework for compiling C/C++ applications to a specialized subset of JavaScript that later evolved into asm.js,[1] an embedded domain specific language that serves as a statically-typed assembly-like language and eschews the dynamic types of JavaScript through additional type coercions coupled with a module-level validation of interprocedural invariants.

Efficient memory safety is a hard design constraint of WebAssembly. Previous systems such as CCured[11] and Cyclone[5] have imposed safety at the C language level, which generally requires program changes. Other attempts have enforced it at the C abstract machine level with combinations of static and runtime checks, sometimes assisted by hardware. For example, the Secure Virtual Architecture[2] defines an abstract machine based on LLVM bitcode that enforces the SAFECode[3] properties.

We investigated reusing other compiler IRs that have a binary format, such as LLVM. Disadvantages with LLVM bitcode in particular are that it is not entirely stable, has undefined behavior, and is less compact than a stack machine. Furthermore, it requires every consumer to either include LLVM, which is notoriously slow, or reimplement a fairly complex LLVM IR decoder/verifier. In general, compiler IRs are better suited to optimization and transformation, and not as compact, verifiable code formats.

In comparison to safe "C" machines, typed intermediate languages, and typed assembly languages,[9] WebAssembly radically reduces the scope of responsibility for the VM: it is not required to enforce the type system of the original program at the granularity of individual objects; instead it must only enforce memory safety at the much coarser granularity of a module's memory. This can be done efficiently with simple bounds checks or virtual memory techniques.

The speed and simplicity of bytecode validation is key to good performance and high assurance. Our work was informed by experience with stack machines such as the Java Virtual Machine (JVM)[8], Common Intermediate Language (CIL)[10], and their validation algorithms. It took a decade of research to properly systematize the details of correct JVM verification,[7] including the discovery of vulnerabilities. By designing WebAssembly in lockstep with a formalization we managed to make its semantics drastically simpler: for example, instead of 150 pages for JVM bytecode verification, just a single page of formal notation.

## 7. FUTURE DIRECTIONS

The initial version of WebAssembly presented here consciously focuses on supporting *low-level* code, specifically compiled from C/C++. A few important features are still missing for fully comprehensive support of this domain and will be added in future versions, such as *exceptions*, *threads*, and *Single Instruction Multiple Data (SIMD)* instructions. Some of these features are already being prototyped in implementations of WebAssembly.

Beyond these, we intend to evolve WebAssembly further into an attractive target for *high-level* languages by including relevant primitives like *tail calls*, *stack switching*, or *coroutines*. A highly important goal is to provide access to the advanced and highly tuned *garbage collectors* that are built into all Web browsers, thus eliminating the main shortcoming relative to JavaScript when compiling to the Web.

In addition to the Web, we anticipate that WebAssembly will find a wide range of uses in other domains. In fact, multiple other embeddings are already being developed: for sandboxing in content delivery networks, for smart contracts or decentralized cloud computing on blockchains, as code formats for mobile devices, and even as mere standalone engines for providing portable language runtimes.

Many challenges lie ahead in supporting all these features and usage scenarios equally well while maintaining the level of precision, rigor, and performance that has been achieved with the initial version of WebAssembly. C


**References**

1. asm.js. http://asmjs.org. Accessed: 2016-11-08.
2. Criswell, J., Lenharth, A., Dhurjati, D., Adve, V. Secure virtual architecture: a safe execution environment for commodity operating systems. *Operating Systems Review 41*, 6 (Oct. 2007), 351–366.
3. Dhurjati, D., Kowshik, S., Adve, V. SAFECode: enforcing alias analysis for weakly typed languages. In *Programming Language Design and Implementation (PLDI)* (2006).
4. Haas, A., Rossberg, A., Schuff, D., Titzer, B., Gohman, D., Wagner, L., Zakai, A., Bastien, J. Bringing the web up to speed with WebAssembly. In *Programming Language Design and Implementation (PLDI)* (2017).
5. Jim, T., Morrisett, J.G., Grossman, D., Hicks, M.W., Cheney, J., Wang, Y. Cyclone: a safe dialect of C. In *USENIX Annual Technical Conference (ATEC)* (2002).
6. Lattner, C., Adve, V. LLVM: a compilation framework for lifelong program analysis & transformation. In *Code Generation and Optimization (CGO)* (2004).
7. Leroy, X. Java bytecode verification: algorithms and formalizations. J. *Automated Reason. 30*, 3–4 (Aug. 2003), 235–269.
8. Lindholm, T., Yellin, F., Bracha, G., Buckley, A. The Java Virtual Machine Specification (Java SE 8 Edition). Technical report, Oracle, 2015.
9. Morrisett, G., Walker, D., Crary, K., Glew, N. From system F to typed assembly language. *ACM Trans. Program. Lang. Sys. (TOPLAS) 21*, 3 (May 1999), 527–568.
10. Necula, G.C., McPeak, S., Rahul, S.P., Weimer, W. CIL: intermediate language and tools for analysis and transformation of C programs. In *Compiler Construction (CC)* (2002).
11. Necula, G.C., McPeak, S., Weimer, W. CCured: Type-safe retrofitting of legacy code. In *Principles of Programming Languages (POPL)* (2002).
12. Pierce, B. *Types and Programming Languages*. The MIT Press, Cambridge, Massachusetts, USA, 2002.
13. Plotkin, G. A structural approach to operational semantics. J. *Logic and Algebraic Program.* (2004), 60–61:17–139.
14. Watt, C. Mechanising and verifying the WebAssembly specification. In *Certified Programs and Proofs (CPP)* (2018).
15. WebAssembly Community Group. WebAssembly Specification, 2018. https://webassembly.github.io/spec/.
16. Wright, A., Felleisen, M. A syntactic approach to type soundness. *Inf. Comput. 115*, 1 (Nov. 1994), 38–94.
17. Yee, B., Sehr, D., Dardyk, G., Chen, B., Muth, R., Ormandy, T., Okasaka, S., Narula, N., Fullagar, N. Native client: a sandbox for portable, untrusted ×86 native code. In *IEEE Symposium on Security and Privacy* (2009).
18. Zakai, A. Emscripten: an LLVM-to-JavaScript compiler. In *Object-Oriented Programming, Systems, Languages, & Applications (OOPSLA)* (2011).



**Andreas Rossberg** (rossberg@mpi-sws.org), Dfinity Stiftung, Germany.

**Ben L. Titzer and Andreas Haas** ([titzer, ahaas]@google.com), Google GmbH, Germany.

**Derek L. Schuff,** (dschuff@google.com), Google Inc, USA.

**Dan Gohman, Luke Wagner, and Alon Zakai** ([sunfishcode, luke, azakai]@mozilla.com), Mozilla Inc, USA.

**JF Bastien** (jfbastien@apple.com), Apple Inc, USA.

**Michael Holman** (michael.holman@microsoft.com), Microsoft Inc, USA.

## Auburn University

**Department of Computer Science and Software Engineering (CSSE)**

***Multiple Faculty Positions in Cybersecurity***

Auburn CSSE invites applications from faculty candidates specializing in all areas related to security, such as *AI/machine learning/data science applications to security, blockchain, cryptocurrency, cybercrime and cyberidentity, cyberinfrastructure protection, digital forensics, reverse engineering, secure cloud, secure mobile systems, secure networks, security enhanced operating systems, secure software engineering, and securing the Internet of Things*. We seek candidates at the Assistant Professor level, however outstanding candidates at a senior level will also be considered. A Ph.D. degree in computer science, software engineering or a closely related field must be completed by the start of appointment. Excellent communication skills are required.

CSSE is home to the Auburn Cyber Research Center (http://cyber.auburn.edu), and is affiliated with the McCrary Institute for Critical Infrastructure Protection and Cyber Systems (http://mccrary.auburn.edu). The department currently has 21 full-time tenure-track and six teaching-track faculty members, who support strong undergraduate and graduate programs (M.S. in CSSE, M.S. in Cybersecurity Engineering, M.S. in Data Science and Engineering expected in fall 2019, and Ph.D. in CSSE). Faculty research areas include artificial intelligence, architecture, computational biology, computer science education, cybersecurity, data science, energy-efficient systems, human-computer interaction, Internet of Things, learning science, machine learning, modeling and simulation, multi-agent systems, networks, software engineering and wireless engineering. Further information may be found at the department's home page http://www.eng.auburn.edu/csse.

Auburn University is one of the nation's premier public land-grant institutions. It is ranked 52nd among public universities by U.S. News and World Report. The university is nationally recognized for its commitment to academic excellence, its positive work environment, its student engagement, and its beautiful campus. Auburn residents enjoy a thriving community, recognized as one of the "best small towns in America," with moderate climate and easy access to major cities or to beach and mountain recreational facilities. Situated along the rapidly developing I-85 corridor between Atlanta, Georgia, and Montgomery, Alabama, Auburn residents have access to excellent public school systems and regional medical centers.

Applicants should submit a cover letter, curriculum vita, research vision, teaching philosophy, and names of three to five references at http://aufacultypositions.peopleadmin.com/postings/3058. There is no application deadline. The application review process will continue until successful candidates are identified. Selected candidates must be able to meet eligibility requirements to work legally in the United States at the time of appointment for the proposed term of employment. Auburn University is an Affirmative Action/Equal Opportunity Employer. It is our policy to provide equal employment opportunities for all individuals without regard to race, sex, religion, color, national origin, age, disability, protected veteran status, genetic information, sexual orientation, gender identity, or any other classification protected by applicable law.

## Boston College

**Assistant Professor of the Practice or Lecturer in Computer Science**

The Computer Science Department of Boston College seeks to fill one or more non-tenure-track teaching positions, as well as shorter-term visiting teaching positions. All applicants should be committed to excellence in undergraduate education, and be able to teach a broad variety of undergraduate computer science courses. Faculty in longer-term positions will participate in the development of new courses that reflect the evolving landscape of the discipline.

Minimum requirements for the title of Assistant Professor of the Practice, and for the title of Visiting Assistant Professor, include a Ph.D. in Computer Science or closely related discipline. Candidates who have only attained a Master's degree would be eligible for the title of Lecturer, or Visiting Lecturer. See https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html for more information.

**To apply go to http://apply.interfolio.com/54268. Application process begins October 1, 2018.**

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at http://www.bc.edu/offices/diversity.

## Boston College

**Associate or Full Professor of Computer Science**

**Description:**
The Computer Science Department of Boston College is poised for significant growth over the next several years and seeks to fill faculty positions at all levels beginning in the 2019-2020 academic year. Outstanding candidates in all areas will be considered, with a preference for those who demonstrate a potential to contribute to cross-disciplinary teaching and research in conjunction with the planned Schiller Institute for Integrated Science and Society at Boston College. See https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html and https://www.bc.edu/bc-web/schools/mcas/sites/schiller-institute.html for more information.

**Qualifications:**
A Ph.D. in Computer Science or a closely related discipline is required, together with a distinguished track record of research and external funding, and evidence of the potential to play a leading role in the future direction of the department, both in the recruitment of faculty and the development of new academic programs.

**To apply go to http://apply.interfolio.com/54226. Application process begins October 1, 2018.**

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at http://www.bc.edu/offices/diversity.

## Boston College

**Tenure Track, Assistant Professor of Computer Science**

The Computer Science Department of Boston College is poised for significant growth over the next several years and seeks to fill faculty positions at all levels beginning in the 2019-2020 academic year. Outstanding candidates in all areas will be considered, with a preference for those who demonstrate a potential to contribute to cross-disciplinary teaching and research in conjunction with the planned Schiller Institute for Integrated Science and Society at Boston College. A Ph.D. in Computer Science or a closely related discipline is required for all positions. See https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html and https://www.bc.edu/bc-web/schools/mcas/sites/schiller-institute.html for more information.

Successful candidates for the position of Assistant Professor will be expected to develop strong research programs that can attract external research funding in an environment that also values high-quality undergraduate teaching.

Minimum requirements for all positions in-

clude a Ph.D. in Computer Science or closely related discipline, an energetic research program that promises to attract external funding, and a commitment to quality in undergraduate and graduate education.

**To apply go to https://apply.interfolio.com/54208. Application review begins October 1, 2018.**

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at http://www.bc.edu/offices/diversity.

## Bradley University
**Tenure Track Assistant Professor**

The Department of Computer Science and Information Systems at Bradley University invites applications for a Tenure Track Assistant Professor position starting in Fall 2019. The Tenure Track Assistant Professor position requires a PhD in Computer Science or a closely related field; we will consider candidates working on their dissertation with research interests in computer science or computer information systems. Please visit www.bradley.edu/humanresources/opportunities for full position description and application process.

## The College at Brockport
**Two Tenure Track Assistant Professor**

Applications are invited for **two** tenure track Assistant Professor positions in the Department of Computing Sciences (home of two ABET accredited programs, https://www.brockport.edu/academics/computing_sciences/) beginning Fall 2019. Doctoral degree in Computer Science or in a closely related field is required. ABD (all but dissertation) applicants are considered. ABD hires must earn their doctoral degree within six months of appointment. Hired faculty will be expected to teach, engage in research and scholarship activities, and participate in service activities appropriate to rank.

**For Position 1:** Expertise and/or teaching experience in one or more of the following areas will be sought: Computer Networks (hardware and software), Cybersecurity, Computer Architecture, Operating Systems.

**For Position 2:** Expertise and/or teaching experience in one or more of the following areas will be sought: Data Structures and Algorithms, Database Systems, Software Engineering and Development (especially on mobile platforms).

Preference will be given to candidates with expertise in the areas mentioned for each position although candidates with expertise in any area of Computer Science will be considered.

Apply online at jobs.brockport.edu. Preference will be given to application materials

## CMU CENTRAL MICHIGAN UNIVERSITY — SOFTWARE ENGINEERING (ASSISTANT PROFESSOR, TENURE-TRACK)

**Location:** Mt. Pleasant, MI

**Category:** Faculty–Science and Engineering–Computer Science

**Type:** Full-time (9-month Academic Year)

The College of Science and Engineering at Central Michigan University is accepting applications for a tenure-track (Assistant Professor) position in the area of Software Engineering to commence August 2019 in the Department of Computer Science. This applicant is expected to contribute to the Department in both graduate and undergraduate teaching, scholarship and service activities. The applicant is also expected to contribute in the growth and success of our current and future programs, and in fostering and growing relationships with other departments both on campus and at other institutions.

The candidate should demonstrate the capability to teach a wide range of core and advanced computer science classes and labs, specializing in the area of software engineering. We currently have undergraduate and graduate courses in software engineering, object-oriented design and capstone courses; successful applicants must be able to teach these courses in addition to other courses to support our programs. Possible areas of research related to software engineering may include but are not limited to software architectures, value-based software design, software security design, automated testing, formal software engineering methods or verification and validation methods. The successful candidate shall justify how their research and teaching fit in the software engineering domain.

A terminal degree in Computer Science, Software Engineering, Information Technology, or a related area is required. Applicants who are ABD in Computer Science or a related area will be considered but the degree should be completed before start date. Applicants must demonstrate a strong commitment to excellence in teaching, scholarship and in providing service to the Department, University and profession. Ability to perform the essential functions of the job with or without reasonable accommodations.

**Contact:** Central Michigan University

**University Link:** **https://www.cmich.edu/Pages/default.aspx**

**Computer Science Link:**
**https://www.cmich.edu/colleges/se/comp_sci/pages/default.aspx**

**Apply Online at:** **https://www.jobs.cmich.edu/postings/search**

CMU is an AA/EO institution, providing equal opportunity to all persons, including minorities, females, veterans, and individuals with disabilities (see **http://www.cmich.edu/ocrie**).

received by January 18, 2019 and application materials will continue to be accepted until the position is filled. EOE/AA employer: M/F/DIS/VET

## Columbia University
**Faculty Positions**

Columbia Engineering invites applications for faculty positions in the Department of Computer Science at Columbia University in the City of New York. Applications at all levels will be considered. Applications are sought in all areas of computer science, with particular emphasis on, but not limited to, the following areas: Computer Systems with an emphasis on Hardware Systems and Cybersecurity.

Candidates must have a Ph.D. or its professional equivalent by the starting date of the appointment. Applicants for this position at the Assistant Professor without tenure level must demonstrate the potential to do pioneering research and to teach effectively. The successful candidate is expected to contribute to the advancement of their field and the department by developing an original and leading externally funded research program, and by contributing to the undergraduate and graduate educational mission of the Department.

Applications should be submitted electronically: http://pa334.peopleadmin.com/postings/1610 and include the following: a cover letter, current CV, teaching statement, brief summary of research, and three letters of recommendation. At least two of the letters of recommendation must address teaching ability. Review of applications will begin on December 1st, 2018 and will continue until the positions are filled.

*Columbia University is an Equal Opportunity/Affirmative Action employer - Disability/Veteran.*

## Florida State University
**Department of Computer Science**
***Tenure-Track Assistant Professor Positions***

The Department of Computer Science at the Florida State University invites applications for two tenure-track Assistant Professor positions to begin August 2019. The positions are 9-month, full-time, tenure-track, and benefits eligible. We are seeking outstanding theoretical and applied applicants in the broad areas of Data Sciences and Trustworthy Computing. The focused areas are Formal Methods and Verification, Software Engineering, Data Analytics, Embedded and/or Cyber-Physical Systems, Visualization, Machine Learning, Digital Forensics, Compilers and Programming Languages, and Computer Architecture.

Applicants should hold a PhD in Computer Science or closely related field at the time of appointment, and have excellent research and teaching accomplishments or potential. The department currently has 22 tenure-track and 7 specialized faculty members and offers degrees at the BS, MS, and PhD levels. Our annual research expenditure has been growing in the past several years and reached $3.7 Million in the 2018 fiscal year. The department is an NSA/DHS Center of Academic Excellence in Cyber Defense Education (CAE/CDE) and Research (CAE-R). FSU is classified as a Carnegie Research I university. Its primary role is to serve as a center for advanced graduate and professional studies while emphasizing research and providing excellence in undergraduate education. Further information can be found at: http://www.cs.fsu.edu

Screening will begin December 1, 2018 and will continue until the positions are filled. Please apply online with curriculum vitae, statements of teaching and research philosophy, and the names of three references, at: http://www.cs.fsu.edu/positions/apply.html

Questions can be e-mailed to Prof. Weikuan Yu, Faculty Search Committee Chair, recruitment@cs.fsu.edu.

**Equal Employment Opportunity**
An Equal Opportunity/Access/Affirmative Action/Pro Disabled & Veteran Employer committed to enhancing the diversity of its faculty and students. Individuals from traditionally underrepresented groups are encouraged to apply.

FSU's Equal Opportunity Statement can be viewed at: http://www.hr.fsu.edu/PDF/Publications/diversity/EEO_Statement.pdf

## Iowa State University
**Assistant Professor**

The Department of Computer Science in the College of Liberal Arts and Sciences at Iowa State University seeks outstanding applicants for two or more tenure-track faculty positions at the rank of Assistant Professor.

**Summary**
The successful candidates will be responsible for developing and sustaining a strong research program; developing collaborative and interdisciplinary research; publishing in top venues; supervising outstanding graduate students; teaching undergraduate and graduate courses; and enhancing ISU through professional and institutional service. We are interested in exceptional candidates from all areas of computer science. Preference will be given to the areas of databases and security and privacy.

**Required Education & Experience**
- Ph.D. or equivalent degree in computer science or a closely related field.
- Publication in top tier venues.

**Preferred Education & Experience**
- A strong publication record.
- Demonstrated ability or potential to develop a high impact, externally funded research program.
- Demonstrated ability to enhance ISU strategic research areas including, but not limited to, databases, security and privacy.
- Experience working with or teaching diverse groups or diverse students.
- Demonstrated ability or potential to excel in teaching at the undergraduate and graduate levels.
- Demonstrated ability or potential to supervise research of undergraduate and graduate students.
- Demonstrated ability or potential to engage in professional and institutional service and leadership.

Guaranteed Consideration Date: November 25, 2018

**Application Information**
For more information or to apply for this position, please visit http://www.iastatejobs.com/postings/36291 and complete the employment application.

Please be prepared to enter or attach the following during the application process:
1) Resume/Curriculum Vitae
2) Letter of Application/Cover Letter
3) Statement of Teaching Philosophy. Please include any experience working with or teaching diverse groups of students.
4) Research Statement.
5) 1 or 2 representative publications.
6) Reference contact information for three references.

Iowa State University is an Equal Opportunity/Affirmative Action employer. All qualified applicants will receive consideration for employment without regard to race, color, religion, sex, national origin, disability, or protected Veteran status and will not be discriminated against.

## Johns Hopkins University
**Lecturer/Sr. Lecturer in Computer Science**

The Department of Computer Science at Johns Hopkins University seeks applicants for a full-time teaching position. This is a career-oriented, renewable appointment that is responsible for the development and delivery of courses primarily to undergraduate students both within and outside the major. These positions carry a 3 course load per semester, usually with only 2 different preps. Teaching faculty are also encouraged to engage in departmental and university service and may have advising responsibilities. Opportunities to teach graduate level courses will depend on the candidate's background. Extensive grading support is given to all instructors. The university has instituted a non-tenure track career path for full-time teaching faculty culminating in the rank of Teaching Professor.

Johns Hopkins is a private university known for its commitment to academic excellence and research. The Computer Science department is one of nine academic departments in the Whiting School of Engineering, on the beautiful Homewood Campus. We are located in Baltimore, MD in close proximity to Washington, DC and Philadelphia, PA. See the department webpages at https://cs.jhu.edu for additional information about the department, including undergraduate programs and current course descriptions.

Applicants for the position should have a Ph.D. in Computer Science or a closely related field; applicants with a Master's degree and significant relevant industry experience will also be considered. Demonstrated excellence in and commitment to teaching, and excellent communication skills are expected of all applicants. Applications may be submitted online at http://apply.interfolio.com/55708. Questions may be directed to lecsearch2018@cs.jhu.edu. For full consideration, applications should be submitted by December 1, 2018. Applications will be accepted until the position is filled.

The Johns Hopkins University is committed to active recruitment of a diverse faculty and student body. The University is an Affirmative Action/Equal Opportunity Employer of women, minorities, protected veterans and individuals with

disabilities and encourages applications from these and other protected group members. Consistent with the University's goals of achieving excellence in all areas, we will assess the comprehensive qualifications of each applicant.

The Whiting School of Engineering and the Department of Computer Science are committed to building a diverse educational environment.

### Max-Planck-Institute for Software Systems

**Tenure-Track Faculty Position at MPI-SWS**

Applications are invited for tenure-track faculty in all areas of computer science. Pending final approval, we expect to fill one position.

A doctoral degree in computer science or related areas and an outstanding research record are required. Successful candidates are expected to build a team and pursue a highly visible research agenda, both independently and in collaboration with other groups.

MPI-SWS is part of a network of over 80 Max Planck Institutes, Germany's premier basic-research organisations. MPIs have an established record of world-class, foundational research in the sciences, technology, and the humanities. The institute offers a unique environment that combines the best aspects of a university department and a research laboratory: Faculty enjoy full academic freedom, lead a team of doctoral students and post-docs, and have the opportunity to teach university courses; at the same time, they enjoy ongoing institutional funding in addition to third-party funds, a technical infrastructure unrivaled for an academic institution, as well as internationally competitive compensation.

The institute is located in the German cities of Saarbruecken and Kaiserslautern, in the tri-border area of Germany, France, and Luxembourg.

We maintain an international and diverse work environment and seek applications from outstanding researchers worldwide. The working language is English; knowledge of the German language is not required for a successful career at the institute.

Qualified candidates should apply on our application website (https://apply.mpi-sws.org). To receive full consideration, applications should be received by December 1st, 2018.

The institute is committed to increasing the representation of women and minorities, as well as of individuals with physical disabilities. We particularly encourage such individuals to apply. The initial tenure-track appointment is for five years; it can be extended to seven years based on a midterm evaluation in the fourth year. A permanent contract can be awarded upon a successful tenure evaluation in the sixth year.

### Purdue University

**Tenure-Track/Tenured Faculty Positions**

The Department of Computer Science in the College of Science at Purdue University solicits applications for at least two tenure-track or tenured positions at the Assistant or Associate Professor levels in areas of computer science. This search is in addition to the separate searches in the areas of theory, cybersecurity, databases and machine learning, and an interdisciplinary hiring effort in data science.

VIRGINIA TECH™

## FACULTY POSITIONS
## Department of Computer Science

The Department of Computer Science at Virginia Tech (**cs.vt.edu**) seeks applicants for at least three faculty positions, including two open rank positions in computer systems and a tenure-track Assistant Professor position in AI/ML. Candidates must have a Ph.D. in computer science or related field at the time of appointment and a rank-appropriate record of scholarship and collaboration in computing research. Successful candidates should give evidence of commitment to issues of diversity in the campus community. Tenured and tenure-track faculty are expected to teach graduate and undergraduate courses, mentor graduate students, and develop a high-quality research program.

**ASSISTANT, ASSOCIATE, OR FULL PROFESSORS IN COMPUTER SYSTEMS.** Candidates are sought for multiple tenured or tenure-track positions in computing systems. Topics of interest include, but are not limited to: operating systems, file and storage systems, distributed systems, cloud, IoT and edge systems, mobile systems, secure and reliable systems, embedded systems, system management, and virtualization. Candidates with interests in systems research topics applicable to high-performance blockchain systems and applications are particularly encouraged to apply. We also welcome candidates at the interface to related areas such as computer architecture, networking, programming languages, and analytics. Successful candidates will have the opportunity to leverage significant ongoing collaboration and support from industrial partners, including block. one, with strengths in decentralized systems and blockchain applications. This initiative is also part of ongoing investment that has led to six tenure-track faculty members joining the department and the stack@cs Center for Computer Systems since 2013. Stack@cs researchers consistently publish and showcase high-impact results at top international venues. Current stack@cs strengths include parallel and distributed systems, file and storage systems, cloud, IoT and edge systems, compilers and runtime systems, computer architecture, systems security, and high-performance computing. Current stack@cs faculty have earned 8 Early Career Awards and numerous faculty awards from industry.

Applications must be submitted online to **jobs.vt.edu** for posting **#TR0180127**. Applicant screening will begin on November 26, 2018 and continue until the position is filled. Inquiries should be directed to Dr. Kirk Cameron, Search Committee Chair, **systemssearch@cs.vt.edu**.

**ASSISTANT PROFESSOR IN AI/ML.** The department seeks applicants for a tenure-track Assistant Professor position in artificial intelligence and/or machine learning. Exceptional candidates at higher ranks may also be considered. Candidates with core research interests in AI/ML are especially encouraged to apply. Some example areas include deep learning, reinforcement learning, natural language processing, probabilistic models, optimization, and learning theory. Example application areas of interest include computer vision, robotics, social informatics, high performance computing, and intelligent systems. The department has an active group of researchers in AI/ML, many of whom are members of the Discovery Analytics Center (**dac.cs.vt.edu**), which leads data science research on campus. CS faculty also collaborate in other interdisciplinary research groups, including the Center for Human Computer Interaction (**hci.vt.edu**), the Center for Business Intelligence and Analytics (**cbia.pamplin.vt.edu**), the Hume Center for National Security and Technology (**hume.vt.edu**) and the Biocomplexity Institute (**bi.vt.edu**). Successful candidates will also have the opportunity to engage in transdisciplinary research, curriculum, and outreach initiatives with other university faculty working in the Data & Decisions Destination Area, one of several university-wide initiatives (**provost.vt.edu/destination-areas**).

Applications must be submitted online to **jobs.vt.edu** for posting **#TR0180129**. Applicant screening will begin on December 10, 2018 and continue until the positions are filled. Inquiries should be directed to Dr. Edward Fox, Search Committee Chair, **fox@vt.edu**.

The Department of Computer Science has 53 teaching faculty, including 47 tenured or tenure-track faculty, over 950 undergraduate majors, and more than 250 graduate students. Department annual research expenditures over the last four years average $12 million. The department is in the College of Engineering, whose undergraduate program ranks 13th and graduate program ranks 30th among all U.S. engineering schools *(US News & World Report).* Virginia Tech's main campus is located in Blacksburg, VA, in a region consistently ranked among the country's best places to live.

*These positions require occasional travel to professional meetings.*

*Selected candidates must pass a criminal background check prior to employment.*

*Virginia Tech is an AA/EEO employer, committed to building a culturally diverse faculty; we strongly encourage applications from traditionally underrepresented communities.*

Applicants must hold a PhD in Computer Science or a related discipline, have demonstrated excellence in research and strong commitment to teaching. Successful candidates will be expected to conduct research in their fields of expertise, teach courses in computer science, and participate in department and university activities.

These positions are part of a continued expansion in a large-scale hiring effort across key strategic areas in the College of Science. Under new leadership, the college is pursuing significant new initiatives which complement campus-wide thrusts including an Integrative Data Science Initiative.

The Department of Computer Science offers a stimulating academic environment with active research programs in most areas of computer science. The department offers undergraduate programs in Computer Science and Data Science, and graduate MS and PhD programs including a Professional MS in Information Security. For more see https://www.cs.purdue.edu.

Applicants should apply online at https://hiring.science.purdue.edu. A background check will be required for employment. Review of applications and interviews will begin in October 2018, and will continue until the positions are filled. Inquiries can be sent to fac-search@cs.purdue.edu.

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. Salary and benefits are competitive, and Purdue is a dual-career friendly employer.

**Purdue University is an EOE/AA employer. All individuals, including minorities, women, individuals with disabilities, and veterans are encouraged to apply.**

## Purdue University

### Tenure-Track/Tenured Faculty Positions in Data Management and Machine Learning

The Department of Computer Science in the College of Science at Purdue University solicits applications for at least two tenure-track or tenured positions at the Assistant, Associate or Full Professor levels in the areas of large scale data management and/or machine learning. This search complements and runs parallel to three other faculty searches covering all other areas of computer science and an interdisciplinary search in the area of data science.

We are particularly interested in candidates whose work focuses on large scale data management and analytics, cloud-based data systems, and ML systems. We are equally interested in interactive learning, representation learning, algorithmic transparency, complex decision making, and cognitive systems to enhance our current strengths in machine learning and data mining. Highly qualified applicants in other areas will be considered. Applicants must hold a PhD in Computer Science or a related discipline, have demonstrated excellence in research and strong commitment to teaching. Successful candidates will be expected to conduct research in their fields of expertise, teach courses in computer science, and participate in department and university activities.

The positions are part of a continued expansion in a large-scale hiring effort across key strategic areas in the College of Science. Under new leadership, the college is pursuing significant new growth and initiatives which complement campus-wide thrusts including an Integrative Data Science Initiative. Opportunities for collaboration exist across mathematics, probability, statistics, and the physical and life sciences.

The Department offers a stimulating academic environment with active research programs in most areas of computer science. The department offers undergraduate programs in Computer Science and Data Science, and graduate MS and PhD programs including a Professional MS in Information Security. For more information see https://www.cs.purdue.edu.

Applicants should apply online at https://hiring.science.purdue.edu. A background check will be required for employment. Review of applications and interviews will begin in October 2018, and will continue until positions are filled. Inquiries can be sent to fac-search@cs.purdue.edu.

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 the cover letter, indicating past experiences, current interests or activities, and/or future goals to promote a climate that values diversity, and inclusion. Salary and benefits are competitive, and Purdue is a dual-career friendly employer.

**Purdue University is an EOE/AA employer. All individuals, including minorities, women, individuals with disabilities, and veterans are encouraged to apply.**

## Purdue University

### Tenure-Track/Tenured Faculty Positions in Security

The Department of Computer Science in the College of Science at Purdue University solicits applications for at least two tenure-track or tenured positions at the Assistant, Associate or Full Professor levels in the area of cybersecurity. This search complements and runs parallel to three other faculty searches covering all other areas of computer science and an interdisciplinary search in the area of data science.

All areas of cybersecurity will be considered. We expect new hires to complement and enhance existing departmental strength in cybersecurity which includes cryptography, data security and privacy, network security, software security, and systems security. Applicants must hold a PhD in Computer Science or a related discipline, have demonstrated excellence in research and strong commitment to teaching. Successful candidates will be expected to conduct research in their fields of expertise, teach courses in computer science, and participate in department and university activities.

These positions are part of a continued expansion in a large-scale hiring effort across key strategic areas in the College of Science. Under new leadership, the college is pursuing significant new growth and initiatives which complement investments in Purdue's most recent strategic thrust, a campus-wide Integrative Data Science Initiative.

The Department of Computer Science offers a stimulating academic environment with active research programs in most areas of computer science. The department offers undergraduate programs in Computer Science and Data Science, and graduate MS and PhD programs including a Professional MS in Information Security. For more information see https://www.cs.purdue.edu.

Applicants should apply online at https://hiring.science.purdue.edu. A background check will be required for employment. Review of applications and interviews will begin in October 2018, and will continue until the positions are filled. Inquiries can be sent to fac-search@cs.purdue.edu.

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. Salary and benefits are competitive, and Purdue is a dual-career friendly employer.

**Purdue University is an EOE/AA employer. All individuals, including minorities, women, individuals w ith disabilities, and veterans are encouraged to apply.**

## Purdue University

### Tenure-Track/Tenured Faculty Positions in Theoretical Computer Science

The Department of Computer Science in the College of Science at Purdue University solicits applications for at least two tenure-track or tenured positions at the Assistant, Associate or Full Professor levels in theoretical computer science. This search complements and runs parallel to three other faculty searches covering all other areas of computer science. It is part of a broader effort to increase presence in the area of theoretical computer science.

We are particularly interested in candidates whose work in theoretical computer science focuses on the design and analysis of algorithms, randomness in computation, graph algorithms, and quantum computing, and whose interests complement existing departmental strengths. Highly qualified applicants in other areas of theoretical computer science will be considered. Applicants must hold a PhD in Computer Science or a related discipline, have demonstrated excellence in research and strong commitment to teaching. Successful candidates will be expected to conduct research in their fields of expertise, teach courses in computer science, and participate in department and university activities.

These positions are part of a continued expansion in a large-scale hiring effort across key strategic areas in the College of Science. Under new leadership, the college is pursuing significant new initiatives which complement campus-wide thrusts including an Integrative Data Science Initiative. Opportunities for collaboration exist across mathematics, probability, statistics, and the physical and life sciences.

The Department of Computer Science offers a stimulating academic environment with active research programs in most areas of computer science. The department offers undergraduate pro-

grams in Computer Science and Data Science, and graduate MS and PhD programs including a Professional MS in Information Security. For more information see https://www.cs.purdue.edu.

Applicants should apply online at https://hiring.science.purdue.edu. A background check will be required for employment. Review of applications and interviews will begin in October 2018, and will continue until positions are filled. Inquiries can be sent to fac-search@cs.purdue.edu.

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 the cover letter, indicating their past experiences, current interests or activities, and/or future goals to promote a climate that values diversity, and inclusion. Salary and benefits are competitive, and Purdue is a dual-career friendly employer.

**Purdue University is an EOE/AA employer. All individuals, including minorities, women, individuals with disabilities, and veterans are encouraged to apply.**

## Rutgers University

### Multiple Tenure-Track Positions

The Computer Science Department at Rutgers University invites applications for several tenure-track positions focusing on (a) Cybersecurity / Computer Systems and Networking, and (b) Artificial Intelligence and Machine Learning. Responsibilities include teaching undergraduate and graduate level courses in various fields of Computer science and supervision of PhD students based on funded projects. The appointments will start September 1st, 2019.

Qualifications: Applicants should show evidence of exceptional research promise with potential for external funding, and commitment to quality advising and teaching. Hired candidates must complete their Ph.D. in Computer Science or a closely related field by August 31, 2019. Applications received by January 14, 2019 will be given priority.

To apply for the Cybersecurity / Computer Systems and Networking position, go to: http://jobs.rutgers.edu/postings/76404

To apply for the Artificial Intelligence and Machine Learning position, go to: http://jobs.rutgers.edu/postings/76620

If you have further questions, please email the hiring committee at: kagarwal@cs.rutgers.edu.

Rutgers subscribes to academic diversity and encourages applications from individuals with varied experiences and backgrounds. Women, minorities, dual-career couples, and persons with disabilities are encouraged to apply. Rutgers is an affirmative action/equal opportunity employer.

## Swarthmore College

### Visiting Assistant Professor in Computer Science

The Department of Computer Science at Swarthmore College invites applications for multiple two-year positions at the rank of Visiting Assistant Professor to begin Fall semester 2019.

Swarthmore College is a highly selective liberal arts college, located in the suburbs of Philadelphia, whose mission combines academic rigor with social responsibility. The Computer Science Department currently has nine tenure-track faculty and four visiting faculty. Faculty teach introductory courses as well as advanced courses in their research areas. We have grown significantly in both faculty and students in the last five years. Presently, we are one of the most popular majors at the College and expect to have over 60 Computer Science majors graduating this year (2019).

**Qualifications**
Applicants must have a Ph.D. in Computer Science or expected by Fall 2019. Applicants strong in any area of computer science will be considered.

**Application Instructions**
Applicants should include a cover letter, a curriculum vitae, a research statement, a teaching statement and three letters of recommendation, including at least one letter specifically commenting on teaching. Applications will not be considered until letters of recommendation have been submitted. Please address any questions you may have to Kathy Reinersmann, Computer Science Department at kreiner1@swarthmore.edu.

Applications will be reviewed on a rolling basis until all positions are filled. For information and to apply, please visit https://apply.interfolio.com/56920.

**Equal Employment Opportunity Statement**
Swarthmore College actively seeks and welcomes applications from candidates with exceptional qualifications, particularly those with demonstrable commitments to a more inclusive society and world. Swarthmore College is an Equal Opportunity Employer. Women and minorities are encouraged to apply.

## University of Central Missouri

### Faculty Positions in Computer Science - Multiple Positions

The School of Computer Science and Mathematics at the University of Central Missouri is accepting applications for two tenure-track positions in Computer Science at the rank of Assistant Professor or Associate Professor. The appointment will begin August 2019.

**Required Qualifications:**
- Ph.D. in Computer Science by August 2019
- Demonstrated ability to teach existing courses at the undergraduate and graduate levels
- Ability to develop a quality research program and secure external funding
- Commitment to engage in curricular development/assessment at the undergraduate and graduate levels
- A strong commitment to excellence in teaching, research, and continued professional growth
- Excellent verbal and written communication skills

**The Application Process:**
To apply online, go to https://jobs.ucmo.edu. Apply to position #997374 or #998332. The following items should be attached: a letter of interest,

a curriculum vitae, a teaching and research statement, copies of transcripts, and a list of at least three professional references including their names, addresses, telephone numbers and email addresses. Official transcripts and three letters of recommendation will be requested for candidates invited for on-campus interview.

Initial screening of applications begins November 15, 2018 and continues until position is filled.

AA/EEO/ADA. Women and minorities are encouraged to apply.

UCM is located in Warrensburg, MO, which is 35 miles southeast of the Kansas City metropolitan area. It is a public comprehensive university with about 13,000 students. The School of Computer Science and Mathematics offers undergraduate and graduate programs in Computer Science, Cybersecurity and Software Engineering with over 700 students. The undergraduate Computer Science and Cybersecurity programs are accredited by the Computing Accreditation Commission of ABET.

## University of Chicago

**Assistant Professor/Associate Professor/ Professor**

The Department of Computer Science at the **University of Chicago** invites applications from qualified candidates for faculty positions at the ranks of Assistant Professor, Associate Professor, and Professor. The University of Chicago is in the midst of an ambitious, multi-year effort to significantly expand its computing and data science activities including a new, state-of-the-art home for the Department of Computer Science that opened this year. Candidates with research interests in all areas of computer science will be considered. However, applications are especially encouraged in all aspects of AI and Machine Learning, Data Science and Analytics, Security, Human-Computer Interaction, and Visual Computing.

Candidates must have demonstrated excellence in research and a strong commitment to teaching. Completion of all requirements for a Ph.D. in Computer Science or a related field is required at the time of appointment. Candidates for Associate Professor and Professor positions must have demonstrated leadership in their field, have established an outstanding independent research program and have a record of excellence in teaching and student mentorship.

Applications must be submitted through the University's Academic Jobs website.

To apply for Assistant Professor, go to https://tinyurl.com/cs-asst-prof

To apply for Associate Professor, go to https://tinyurl.com/cs-assoc-professor

To apply for Professor, go to https://tinyurl.com/cs-professor

The following materials are required:

- ▸ cover letter
- ▸ curriculum vitae including a list of publications
- ▸ statement describing past and current research accomplishments and outlining future research plans
- ▸ description of teaching philosophy and experience
- ▸ the names of at least three references

Please have your references submit their letters via e-mail to recommend@cs.uchicago.edu.

Review of applications will begin on December 15, 2018 and continue until positions are filled.

The University of Chicago has the highest standards for scholarship and faculty quality, is dedicated to fundamental research, and encourages collaboration across disciplines. We encourage connections with researchers across campus in such areas as bioinformatics, mathematics, molecular engineering, natural language processing, statistics, public policy, and social science to mention just a few.

The Department of Computer Science (cs.uchicago.edu) is the hub of a large, diverse community of researchers focused on advancing the foundations of computing and driving its most advanced applications. The larger computing and data science community at the University of Chicago includes the Department of Statistics, the Center for Data and Applied Computing, the Toyota Technological Institute at Chicago (TTIC), the Polsky Center for Entrepreneurship and Innovation, the Mansueto Institute for Urban Innovation and the Argonne National Laboratory.

The Chicago metropolitan area provides a diverse and exciting environment. The local economy is vigorous, with international stature in banking, trade, commerce, manufacturing, transportation and a growing tech start-up scene. Lifestyle advantages include diverse cultures, fantastic restaurants, vibrant theater, world-renowned symphony, opera, jazz, and blues. The University is located in Hyde Park, a Chicago neighborhood on the Lake Michigan shore just a few minutes from downtown.

The University of Chicago is an Affirmative Action/Equal Opportunity/Disabled/Veterans Employer and does not discriminate on the basis of race, color, religion, sex, sexual orientation, gender identity, national or ethnic origin, age, status as an individual with a disability, protected veteran status, genetic information, or other protected classes under the law. For additional information please see the University's Notice of Nondiscrimination at http://www.uchicago.edu/about/non_discrimination_statement/. Job seekers in need of a reasonable accommodation to complete the application process should call 773-702-0287 or email ACOppAdministrator@uchicago.edu with their request.

## University of Illinois at Chicago

**Lecturer – Non-Tenure Track – Computer Science**

The Computer Science Department at the University of Illinois at Chicago is seeking multiple full-time teaching faculty members to start Fall 2019. The lecturer teaching track is a long-term career track that starts with the Lecturer position, and offers opportunities for advancement to Senior Lecturer. Candidates would be working alongside 13 full-time teaching faculty with over 150 years of combined teaching experience and 12 awards for excellence. The department seeks candidates dedicated to teaching; candidates must have evidence of effective teaching, or present a convincing case of future dedication and success in the art of teaching. Content areas of interest include introductory programming, data structures, computer organization/systems, web development, data science, software engineering, and machine learning. The standard teaching load is 2-3 undergraduate courses per semester (depending on course enrollment).

The University of Illinois at Chicago (UIC) is one of the top-10 most diverse universities in the US (US News and World Report), a top-10 best value (Wall Street Journal and Times Higher Education) and a hispanic serving institution. UIC's hometown of Chicago epitomizes the modern, livable, vibrant city. Located on the shore of Lake Michigan, Chicago offers an outstanding array of cultural and culinary experiences. As the birthplace of the modern skyscraper, Chicago boasts one of the world's tallest and densest skylines, combined with an 8100-acre park system and extensive public transit and biking networks.

Minimum qualifications include an MS in Computer Science or a closely related field or appropriate graduate degrees for specific course material (e.g., computer ethics), and either (a) demonstrated evidence of effective teaching, or (b) convincing argument of future dedication and success in the art of teaching. Applications are submitted online at https://jobs.uic.edu/. In the online application, include a curriculum vitae, names and addresses of at least three references, a statement providing evidence of effective teaching, and a statement describing your past experience in activities that promote diversity and inclusion (or plans to make future contributions), and recent teaching evaluations. For additional information contact Professor Mitch Theys, Committee Chair, mtheys@uic.edu.

## University of Illinois at Chicago (UIC)

**Open Rank Tenure Track Faculty Positions - Computer Science**

Located in the heart of Chicago, the Department of Computer Science at the University of Illinois at Chicago (UIC) invites applications for several full-time tenure-track positions at all ranks. All candidates must have a PhD in Computer Science or a closely related field by the appointment's starting date. Candidates will be expected to demonstrate excellence in research and to teach effectively at the undergraduate and graduate levels.

We seek candidates in all areas of computing, with special but not exclusive interest in fields related to computer vision, machine learning/data science, human-computer interaction, systems/software engineering, programming languages and compilers, and applied cryptography. Applicants working at the intersection of computer science and related disciplines are also encouraged to apply.

Applications must be submitted at https://jobs.uic.edu/, and must include a curriculum vitae, teaching and research statements, and names and addresses of at least three references. Links to a professional website such as Google Scholar or ResearchGate are recommended. Applicants may contact the faculty search committee search@cs.uic.edu for more information. For fullest consideration, applications must be submitted by November 15, 2018. Applications will be accepted until the positions are filled.

The Department of Computer Science at UIC, which will be hiring between 20 and 35 new

faculty in the next 6 years, has 36 tenure-system faculty and 4 research faculty with strong and broad research agendas, and 13 clinical/teaching faculty. The department is committed to building a diverse faculty preeminent in its missions of research, teaching, and service to the community. Candidates who have experience engaging with a diverse range of faculty, staff, and students, and contributing to a climate of inclusivity are encouraged to discuss their perspectives on these subjects in their application materials.

UIC is a major public research university (Carnegie R1) with about 2,800 faculty and over 31,000 students. UIC is committed to increasing access to education, employment, programs and services for all. The University of Illinois is an Equal Opportunity, Affirmative Action employer. Minorities, women, veterans, and individuals with disabilities are encouraged to apply. UIC is committed to supporting the success of dual-career couples.

Chicago epitomizes the modern, livable, vibrant, and diverse city. World-class amenities like the lakefront, arts and culture venues, festivals, and two international airports make Chicago a singularly enjoyable place to live. Yet the cost of living, whether in an 88th floor condominium downtown or on a tree-lined street in one of the nation's finest school districts, is remarkably affordable. The University of Illinois conducts background checks on all job candidates upon acceptance of contingent offer of employment. Background checks will be performed in compliance with the Fair Credit Reporting Act.

## University of Maryland, Baltimore County (UMBC)

### Open Rank Tenure-Track Faculty Positions in Artificial Intelligence/Knowledge Management and Human-Centered Computing

The Department of Information Systems (IS) at UMBC invites applications for two open rank tenure-track faculty positions starting August 2019. We are searching for candidates with research interests and experience in Artificial Intelligence / Knowledge Management (KM), and in Human-Centered Computing (HCC).

Candidates for the positions will have expertise in conducting research that intersects with and extends other current active research areas in the IS department.

Major research areas in the department include Artificial Intelligence, Data Science, Human-centered Computing, Health Information Technology, and Software Engineering. Strong candidates with research emphases in other areas may also be considered.

Candidates must have earned a PhD in Information Systems, Human-Centered Computing or a related field no later than August 2019. Candidates are expected to establish a collaborative, externally funded, and nationally recognized research program; they are expected to contribute to teaching a variety of graduate and undergraduate courses offered by the department effectively.

The Department offers degrees at both the undergraduate and graduate levels. Further details on our research, academic programs, and faculty can be found at http://is.umbc.edu and http://hcc.umbc.edu.

UMBC is a national model for diversity and inclusive excellence in STEM through its Meyerhoff Scholar http://meyerhoff.umbc.edu/ and CWIT Scholar programs http://cwit.umbc.edu - to promote diversity and prepare students from underrepresented groups for careers in STEM. We especially welcome applications from candidates who are willing to contribute to the diversity mission of the university. The IS Department is committed to increasing the diversity of our community. Members of underrepresented groups including women, minorities, veterans, and individuals with disabilities are especially encouraged to apply.

To apply for the **AI/KM position**: visit http://apply.interfolio.com/55824. Candidates' experience will be evaluated commensurate with the rank they are applying to. For inquiries on the AI/KM position, please contact Dr. George Karabatis at (410) 455-3940 or georgek@umbc.edu. Review of applications will begin on November 30, 2018 and will continue until the position is filled, subject to the availability of funds.

To apply for the **HCC position**: visit https://apply.interfolio.com/55820. Questions regarding the HCC position may be addressed to: Dr. Anita Komlodi (komlodi@umbc.edu) and Dr. Ravi Kuber (rkuber@umbc.edu). Full consideration will be given to those applicants who submit all materials by November 15, 2018.

**UMBC is an Affirmative Action/Equal Opportunity Employer and welcomes applications from minorities, women, veterans, and individuals with disabilities.**

## University of Michigan - Dearborn

### Assistant Professors in Computer and Information Science

The CIS Department at the University of Michigan - Dearborn invites applications for multiple tenure-track assistant professor positions in all areas of computer science, with special emphasis on software engineering, computer systems (e.g., distributed systems), data science/AI/machine learning, and data management. The expected starting date is September 1, 2019. Although candidates at the Assistant Professor rank are preferred, exceptional candidates may be considered for the rank of Associate Professor depending upon experience and qualifications. We offer competitive salaries and start-up packages.

The CIS Department offers several B.S. and M.S. degrees, and a Ph.D. degree. The current research areas in the department include artificial intelligence, computational game theory, computer graphics, data science, data management, energy-efficient systems, game design, graphical models, machine learning, multimedia, natural language processing, networking, security, service and cloud computing, software engineering, and wearable sensors and health informatics. These areas of research are supported by several established labs and many of these areas are currently funded by federal agencies and industries.

**Qualifications:**
Qualified candidates must have earned a Ph.D. degree in computer science or a closely related

UNLV

## Computer Science, Big Data/Health Disparities Assistant/Associate/Professor UNLV College of Engineering

The Department of Computer Science at the University of Nevada, Las Vegas (UNLV) invites applications for a full-time tenure-track/tenured faculty position [R0112275] at all ranks in the area of Big Data commencing Fall 2019.

The selected applicant will be expected to perform excellent research, teaching, and service in Big Data including data mining, data analytics, data visualization, database modeling, machine learning, scalable computing, software and hardware systems for big data processing, and distributed/parallel computing.

The UNLV Top Tier Initiative is an extension of our vision of entering the top 100 American research universities, as designated by the Carnegie Foundation as a Highest Research Activity (R1) University. The Computer Science department is integral to UNLV's Top Tier Initiative with research activities and expenditures in Big Data, Programming Language, Cybersecurity and Theory areas. Our faculty members have received many prestigious research and educational awards, including The U.S. Congressional Recognition Award, The White House Champions of Change award in computer science education, The U.S. Army Mentorship Award, IEEE Computer Society Distinguished Service Award, etc.

For more information, please visit **https://www.unlv.edu/jobs**

For assistance with the application process, please contact UNLV Human Resources at (702) 895-3504 or **applicant.inquiry@unlv.edu**.

*EEO/AA/Vet/Disability Employer*

discipline by September 1, 2019. Candidates will be expected to do scholarly and sponsored research, as well as teaching at both the undergraduate and graduate levels.

**Applications:**
Applicants should send a letter of intent, indicating which one of the following four areas fits you the best: (1) software engineering; (2) computer systems; (3) data science/AI/machine learning; (4) data management or other areas. You should also submit a curriculum vitae, statements of teaching and research interests, evidence of teaching performance (if any), and a list of three references through Interfolio at: http://apply.interfolio.com/56046.

Review of applications will begin immediately and continue until suitable candidates are appointed.

The University of Michigan - Dearborn, as an equal opportunity/affirmative action employer, complies with all applicable federal and state laws regarding nondiscrimination and affirmative action. The University of Michigan is committed to a policy of equal opportunity for all persons and does not discriminate on the basis of race, color, national origin, age, marital status, sex, sexual orientation, gender identity, gender expression, disability, religion, height, weight, or veteran status in employment, educational programs and activities, and admissions. Inquiries or complaints may be addressed Office of Institutional Equity, 4901 Evergreen Road, Suite 1020, Administrative Services Building, Dearborn, Michigan 48128-1491, (313) 593-5190.

## University of Minnesota-Twin Cities

### Tenure-Track Faculty Positions in Computer Science and Engineering

The Department of Computer Science & Engineering at the University of Minnesota-Twin Cities is hiring to fill multiple tenure-track positions at the assistant professor level, although higher levels of appointments may be considered when commensurate with experience and accomplishments. Candidates with teaching and research interests in software engineering; human-computer interaction; theoretical computer science; systems related to big data processing, cloud, embedded, mobile and stream computing; privacy and security in data and information systems; and platforms and paradigms to support big data and Internet-of-Things (IoT) are encouraged to apply.

One of the positions is in support of a University-wide initiative (MnDRIVE) on robotics, sensors, and advanced manufacturing (http://cse.umn.edu/mndrive). Topics of interest include machine learning; computer graphics/simulation/visualization; robot design, manipulation, mobility, planning, algorithmic foundations, and human-robot interaction; and embedded systems.

The Department of Computer Science & Engineering is fully committed to a diverse faculty because excellence emerges when individuals with different backgrounds and experiences engage. We therefore welcome applications from individuals who will further expand that diversity; women and other underrepresented groups are especially encouraged to apply. Candidates must have a Ph.D. in Computer Science or a closely related discipline at the time of appointment. Submit materials as described at https://wwws.cs.umn.edu/appflow/faculty18. Consideration of completed applications will begin December 1, 2018, and continue until the positions are filled. The University of Minnesota is an equal opportunity employer and educator.

## The University of Tennessee, Knoxville (UTK)

### Five (5) Tenure Track Faculty Positions in Computer Science, Computer Engineering, or Electrical Engineering

The Department of Electrical Engineering and Computer Science (EECS) at The University of Tennessee, Knoxville (UTK) is seeking candidates for five (5) tenure track faculty members at the assistant or associate professor level in computer science, computer engineering, or electrical engineering. Applicants should have an earned Ph.D. in Computer Science, Computer Engineering, Electrical Engineering, or a related field. The department is expanding its teaching and research in the areas of (1) data analytics, machine learning, and artificial intelligence, (2) internet of things and mobile computing systems, including cybersecurity, cloud and fog computing, embedded systems, signal processing, and energy efficiency, (3) VLSI circuits and systems, including digital design, analog/mixed signal circuits, and beyond CMOS technologies, and (4) power systems, power electronics, smart grids, and/or cyber-physical security with strong applications in power grids. We welcome applicants in these and other areas of computer science, computer engineering, and electrical engineering. Successful candidates will be expected to teach at both undergraduate and graduate levels, to establish a vigorous funded research program, and to have a willingness to collaborate with other faculty in research.

EECS is housed in a new $37.5 million teaching and research facility completed in 2012. The department currently has an enrollment of more than 800 undergraduate and 250 graduate students, with a faculty of 45, and research expenditures that exceed $17 million per year. EECS offers two undergraduate minors in cybersecurity and datacenter technology and management that were started in 2015. Successful candidates will be expected to contribute to the expansion of related educational and research activities. UTK is a leading research institution with strong research partnerships with organizations such as the nearby Oak Ridge National Laboratory (ORNL) where several UT faculty have joint positions or research ties.

The Knoxville campus of the University of Tennessee is seeking candidates who have the ability to contribute in meaningful ways to the diversity and intercultural goals of the University. The University of Tennessee welcomes and honors people of all races, genders, creeds, cultures, and sexual orientations, and values intellectual curiosity, pursuit of knowledge, and academic freedom and integrity. Interested candidates should apply through the departmental web site at http://www.eecs.utk.edu/people/employment/ and submit a cover letter, a curriculum vitae, a statement of research and teaching interests, and contact information for three references. Review of applications will begin on December 21, 2018, and continue until the positions are filled.

Applicants should have an earned Ph.D. in Computer Science, Computer Engineering, Electrical Engineering, or a related field. Successful candidates will be expected to teach at both undergraduate and graduate levels, to establish a vigorous funded research program and to have willingness to collaborate with other faculty in research.

*The University of Tennessee is an EEO/AA/Title VI/Title IX/Section 504/ADA/ADEA institution in the provision of its education and employment programs and services. All qualified applicants will receive equal consideration for employment without regard to race, color, national origin, religion, sex, pregnancy, marital status, sexual orientation, gender identity, age, physical or mental disability, or covered veteran status.*

## The University of Texas at San Antonio (UTSA)

### Faculty Position in Computer Science

The Department of Computer Science at The University of Texas at San Antonio (UTSA) invites applications for **two open rank** (Assistant, Associate or Full Professor) positions, starting in Fall 2019. One position is targeted towards faculty with expertise and interest in artificial intelligence (AI). Outstanding candidates from all areas of AI will be considered, and preference will be given to applicants with expertise in cyber adversarial learning, AI for resource-constrained systems (such as IoTs and embedded systems), or AI (such as natural language processing, computer vision and deep learning) as it relates to health-related applications. The second position will consider outstanding candidates from all areas of computer science and special considerations will be given to candidates in the areas of data science, cyber security, quantum computing, programming languages and compilers, architecture, and systems.

See http://www.cs.utsa.edu/fsearch for more information on the Department and application instructions. Screening of applications will begin immediately. Application received by **January 2, 2019** will be given full consideration. The search will continue until the positions are filled or the search is closed. The University of Texas at San Antonio is an Affirmative Action/Equal Opportunity Employer. Women, minorities, veterans, and individuals with disabilities are encouraged to apply.

**Department of Computer Science**
**RE: Faculty Search**
**The University of Texas at San Antonio**
**One UTSA Circle**
**San Antonio, TX 78249-0667**
**Phone: 210-458-4436**

# 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.

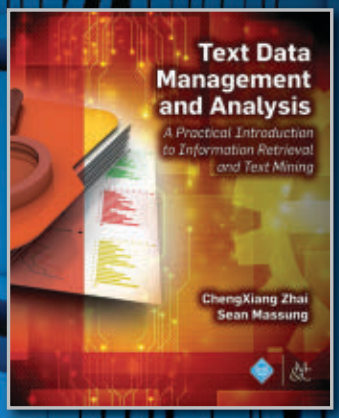
The following institutions currently participate in ACM's Academic Department Membership program:

- AAU Klagenfurt
- Abilene Christian University
- Amherst College
- Appalachian State University
- Armstrong State University
- Ball State University
- Bellevue College
- Berea College
- Binghamton University
- Boise State University
- Bryant University
- Calvin College
- Colgate University
- Colorado School of Mines
- Cornell University
- Creighton University
- Cuyahoga Community College
- Edgewood College
- Franklin University
- Gallaudet University
- Georgia Institute of Technology
- Governors State University
- Harding University
- Harvard University
- Hofstra University
- Hope College
- Howard Payne University
- Indiana University - Bloomington
- Indiana University Bloomington, Information & Library Science
- Kent State University
- La Sierra University
- Messiah College
- Metropolitan State University
- Missouri State University
- Montclair University
- Mount Holyoke College
- New Jersey Institute of Technology
- Northeastern University
- Old Dominion University
- Pacific Lutheran University
- Potomac State College of West Virginia
- Regis University
- Roosevelt University
- Rutgers University
- SUNY Oswego
- Saint Louis University
- San Jose State University | Davidson College of Engineering
- Shippensburg University
- Simmons College
- St. John's University
- Stanford University
- Stetson University
- The Ohio State University
- The Pennsylvania State University
- The State University of New York at Fredonia
- The University of Alabama
- The University of Memphis
- Trine University
- Trinity University
- UC San Diego
- UNC Charlotte
- USC, University of Southern California
- Union College
- Union University
- University of California, Riverside
- University of California, Santa Cruz
- University of Colorado Boulder
- University of Colorado, Denver
- University of Houston
- University of Illinois at Chicago
- University of Jamestown
- University of Liechtenstein
- University of Maryland, Baltimore County
- University of Nebraska at Kearney
- University of Nebraska at Omaha
- University of New Mexico
- University of North Dakota
- University of Pittsburgh
- University of Porto, Faculty of Engineering
- University of Puget Sound
- University of Victoria
- University of Wisconsin - Parkside
- University of Wyoming
- University of the Fraser Valley
- Virginia Commonwealth University
- Wake Forest University
- Wayne State University
- Wellesley College
- Western New England University
- William Jessup University - Rocklin Campus
- Worcester State University

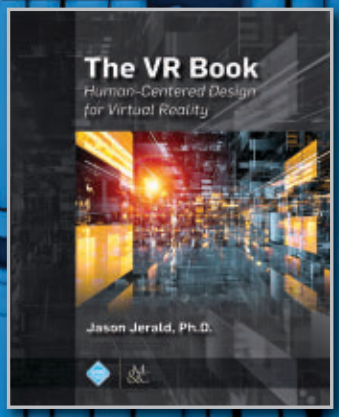
Through this program, each faculty member receives all the benefits of individual professional membership, including *Communications of the ACM*, member rates to attend ACM Special Interest Group conferences, member subscription rates to ACM journals, and much more.

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

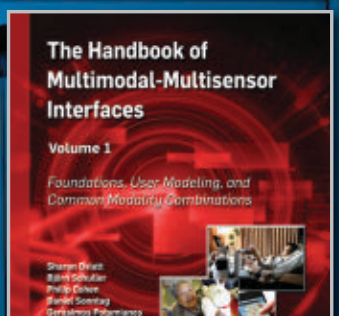
Association for Computing Machinery

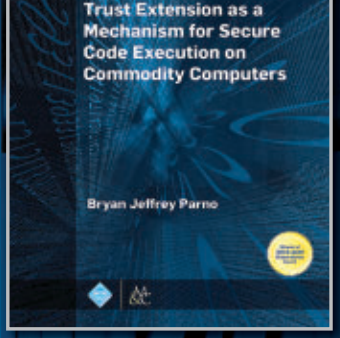


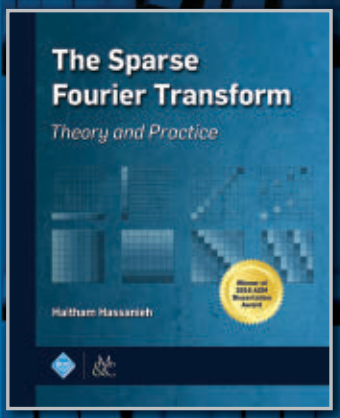


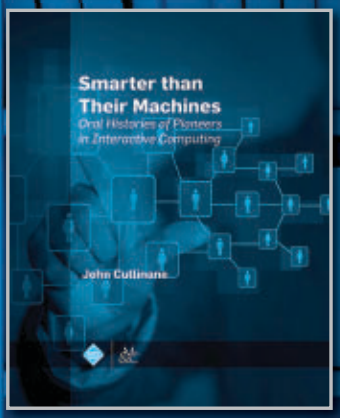


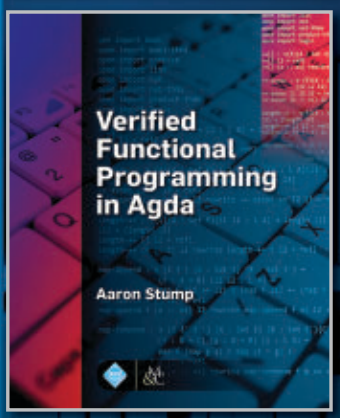







# In-depth. Innovative. Insightful.

Inspired by the need for high-quality computer science publishing at the graduate, faculty, and professional levels, ACM Books are affordable, current, and comprehensive in scope.

## Full Collection I Title List Now Available

For more information, please visit **http://books.acm.org**



**Association for Computing Machinery**
2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA
Phone: +1-212-626-0658 Email: acmbooks-info@acm.org

[CONTINUED FROM P. 128] mentors, Roger Nash Baldwin, who founded the ACLU (American Civil Liberties Union) in 1919.

Altruism is perhaps the most important virtue we need to maintain, especially in times of adversity. "And that's the word: *Altruism*."

With today's fake news and myriad forms of disinformation, perhaps we need a word similar to altruism for the "unselfish concern for the truth," which we might pronounce as *all-true-ism*.

**What do you make of the fallout from the 2016 election?**

As a civilization, we must work even harder to promote common sense, reality, the relevance of science and dependable engineering, and above all, the truth. All of these are fundamental to human rights, election integrity, freedom of speech, civil rights, and the preservation of democracy. The risks relating to technologies such as artificial intelligence, machine learning, the Internet of Things, and social media, are generally not sufficiently well understood from the perspective of security—especially in the absence of trustworthy systems and trustworthy people, and in the presence of misanthropes with no respect for values, ethics, morals, and established knowledge.

**Are you optimistic about the GDPR (the European Union's General Data Protection Regulation) and other privacy initiatives?**

Many efforts relating to security and privacy fall victim to the reality that our computer systems are still inherently untrustworthy, and easily attacked by the Russians, Chinese, corrupt insiders, and potentially everyone else. Everything seems to be hackable, or otherwise adversely influenced, including elections, automobiles, the Internet of Things, cloud servers, and more. Essentially, the needs for better safety, reliability, security, privacy, and system integrity that I highlighted 24 years ago in my book, *Computer-Related Risks*, are still with us in one form or another today. If we do not have systems that are sufficiently trustworthy, respecting privacy remains even more challenging.

**Before 2016, do you think computer scientists were guilty of focusing too closely on the technical security of voting systems, and not enough on the hackability of human behavior? Aside from the public opprobrium that Facebook and other social media outlets have since faced, do you think we are doing better to incorporate the total scope of threats to free and fair elections?**

"As a civilization, we must work even harder to promote common sense, reality, the relevance of science and dependable engineering, and above all, the truth."

Given the enormous risks of direct-recording election equipment (DREs) with only proprietary software, proprietary data formats, and proprietary data during elections, and no meaningful audit trails or possibilities for remediating obviously fraudulent results, our initial efforts were urgently devoted to making the case for voter-verified paper trails that would be the ballot choices of record. For example, I testified in January 1995 for the New York City Board of Elections, and David Dill, Barbara Simons, and I spoke in multiple hearings in 2003 before the Santa Clara County (CA) supervisors, who were planning to acquire $24-million worth of paperless DREs. Dan Boneh, David Dill, Doug Jones, Avi Rubin, Dave Wagner, Dan Wallach, and I participated for seven years beginning in 2005 in an NSF (National Science Foundation) collaborative effort called ACCURATE: A Center for Correct, Usable, Reliable, Auditable and Transparent Elections. (For more recent analysis, see *Broken Ballots: Will Your Vote Count*, by Doug Jones and Barbara Simons; http://www.timbergroves.com/bb/.)

It was evident that unauditable DREs were a huge weak link. On the other hand, I have long maintained that essentially every step in the election process represents a potential weak link to undermine democracy. For example, various dirty-tricks efforts in Richard Nixon's House, Senate, and Presidential elections were a harbinger of the use of non-technological tactics. The Kerry Swift-boating attacks in 2004 should have been another warning sign. However, the 2016 election should really bring Citizens' United, targeted disinformation, creative redistricting, and other issues to the forefront, although most of the computer scientists working in this area are generally still focused primarily on the computer systems, because the politicians have often been rather disinterested in the big picture or in the technology—apart from some recent concerns about Facebook, Cambridge Analytica, and related issues.

**Redistricting and disenfranchisement are also huge concerns.**

As I write this, the Supreme Court has just upheld Ohio's law to remove voters who are not voting "frequently enough." In addition, the Supremes seem to be unable to cope with mathematical reasoning and sound logic.

Leah, as you yourself have suggested to me quite incisively in a broader context, "if we can't agree on parameters for using or distributing a particular set of tools, we cede it to malicious forces as a matter of course."

**When we spoke previously, you were working on an effort to develop new systems that could be much more trustworthy. Can you give me an update?**

Certainly. Our hardware-software design and development efforts based on our CHERI (Capability Hardware Enhanced RISC Instructions) instruction-set architecture (ISA) began in 2010, and will now continue into early 2021. We are also formally verifying that the ISA satisfies certain critical properties. This is joint work between SRI and the University of Cambridge. Our website (http://www.cl.cam.ac.uk/research/security/ctsrd/) includes the latest hardware ISA specification (along with several variant possible CHERI implementations, and ongoing tech transfer), as well as our published papers.

**Leah Hoffmann** is a technology writer based in Piermont, NY, USA.

DOI:10.1145/3283226

Leah Hoffmann

# Q&A
# Promoting Common Sense, Reality, Dependable Engineering

*Peter G. Neumann traces a lifetime devoted to identifying computing risks.*

ON JUNE 6, Peter G. Neumann, computer security's "designated holist"—a name given to him by ISCA's *Information Security* magazine—received an award designed to recognize not just his contributions to the field of systems design, but the broader impact his work has made to privacy, civil liberties, and democracy. Neumann was presented with the Lifetime Achievement Award at the Champions of Freedom event held by the Electronic Privacy Information Center (EPIC). During the reception, Neumann highlighted the need to counter the abuse of emerging technologies by promoting common sense, reality, and dependable engineering. Here, he elaborates on those ideas.

**What was it like to receive EPIC's Lifetime Achievement Award? It seems like a good fit for someone who takes a big-picture view of systems, privacy, and behavioral and technological failures.**

I was delighted to have Rush Holt introduce me, highlighting many of my holistically motivated interests. Rush has been enormously valuable when it comes to science and technology, including his efforts to help achieve greater integrity, fairness, and for quite a long time huge support for our community efforts to have much greater trustworthiness to our elections.

The recipients of the Champions of Freedom Awards were Alex Padilla and Matthew Dunlap, Secretaries of State for California and Maine, both of whom have been enormously effective in striving for greater integrity in elections. Ralph Nader was also present, having long been a champion of automobile and consumer safety. This was a delightful venue, and it was quite an honor to be among those leaders.

The event also genuinely reflected my long and polymorphic association with Marc Rotenberg (one-time executive director of Computer Professionals for Social Responsibility before he created EPIC 24 years ago).

**"Altruism is perhaps the most important virtue we need to maintain, especially in times of adversity."**

**Can you share some of your remarks from the awards ceremony? I understand that you spoke about "altruism"—can you explain why you brought up that term, especially in the context of privacy and civil liberties?**

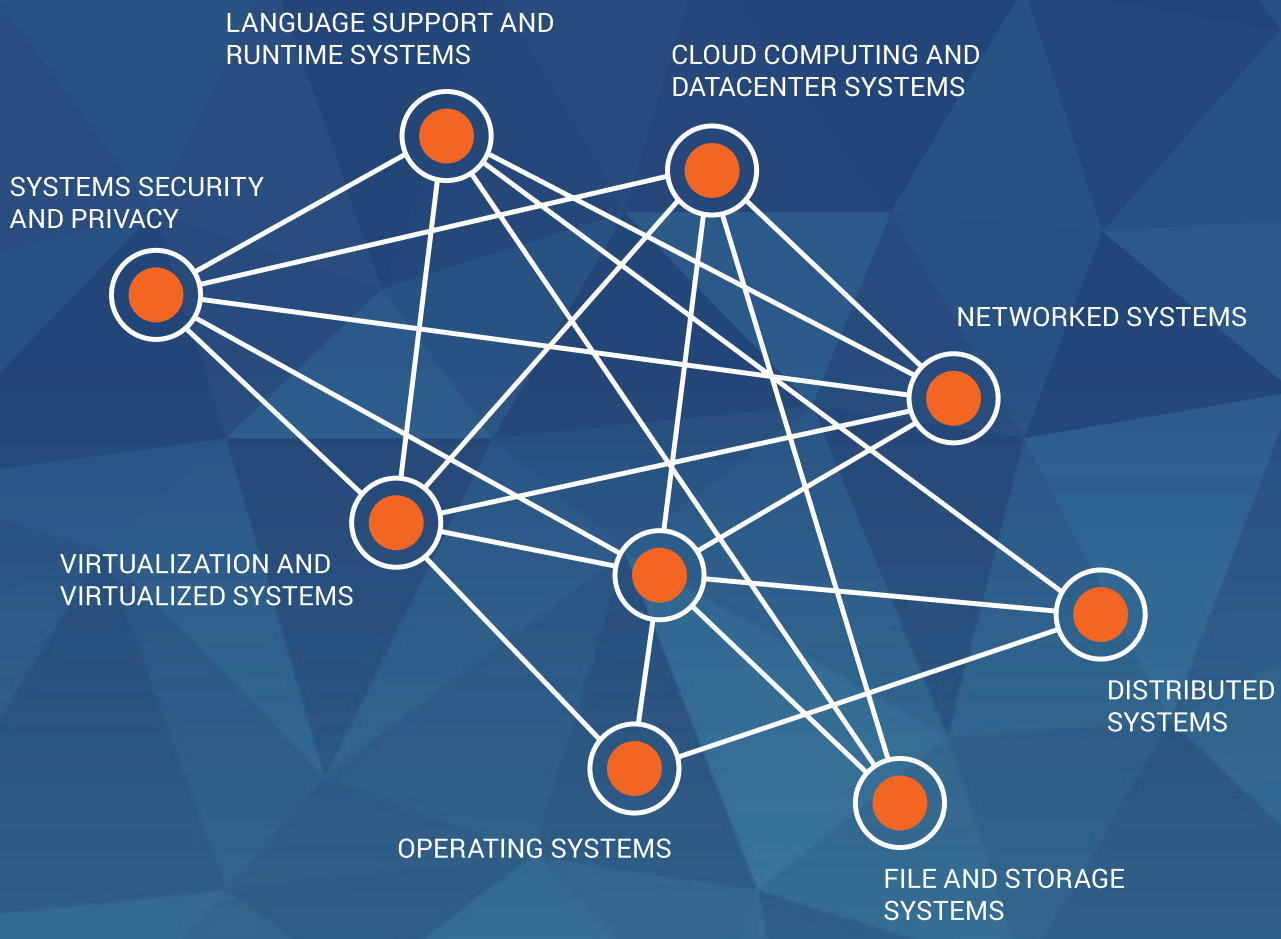
I attempted to channel Stephen Colbert when I observed that there is an important word that was highly relevant to the common interests of everyone present, but which otherwise seems to be sorely lacking in Washington. That word is *altruism* ("unselfish concern for the welfare of others," including in this context their human rights and privacy). Rush, Ralph, and Marc all share that quality, as did one of my favorite [CONTINUED ON P. 127]

PHOTO BY JIM WILSON/THE NEW YORK TIMES/REDUX

25 - 28 MARCH | DRESDEN | GERMANY

14TH EUROPEAN CONFERENCE ON COMPUTER SYSTEMS

# <EURO/SYS'19>




## ORGANIZING COMMITTEE

**GENERAL CHAIR**
Christof Fetzer TU Dresden

**PROGRAM COMMITTEE CHAIRS**
George Candea EPFL
Robbert Van Renesse Cornell University

## REGISTRATION OPEN

Early Bird Registration until **22 Feb, 2019**

For more information visit
**www.eurosys2019.org**
eurosys2019@tu-dresden.de
#eurosys2019

Association for Computing Machinery
SIGOPS
TECHNISCHE UNIVERSITÄT DRESDEN
cfaed

The Art, Science, and Engineering of Programming

# ‹Programming› 2019

acm In-Cooperation

SIGPLAN SIGSOFT

## Genova, Italy April 1–4, 2019

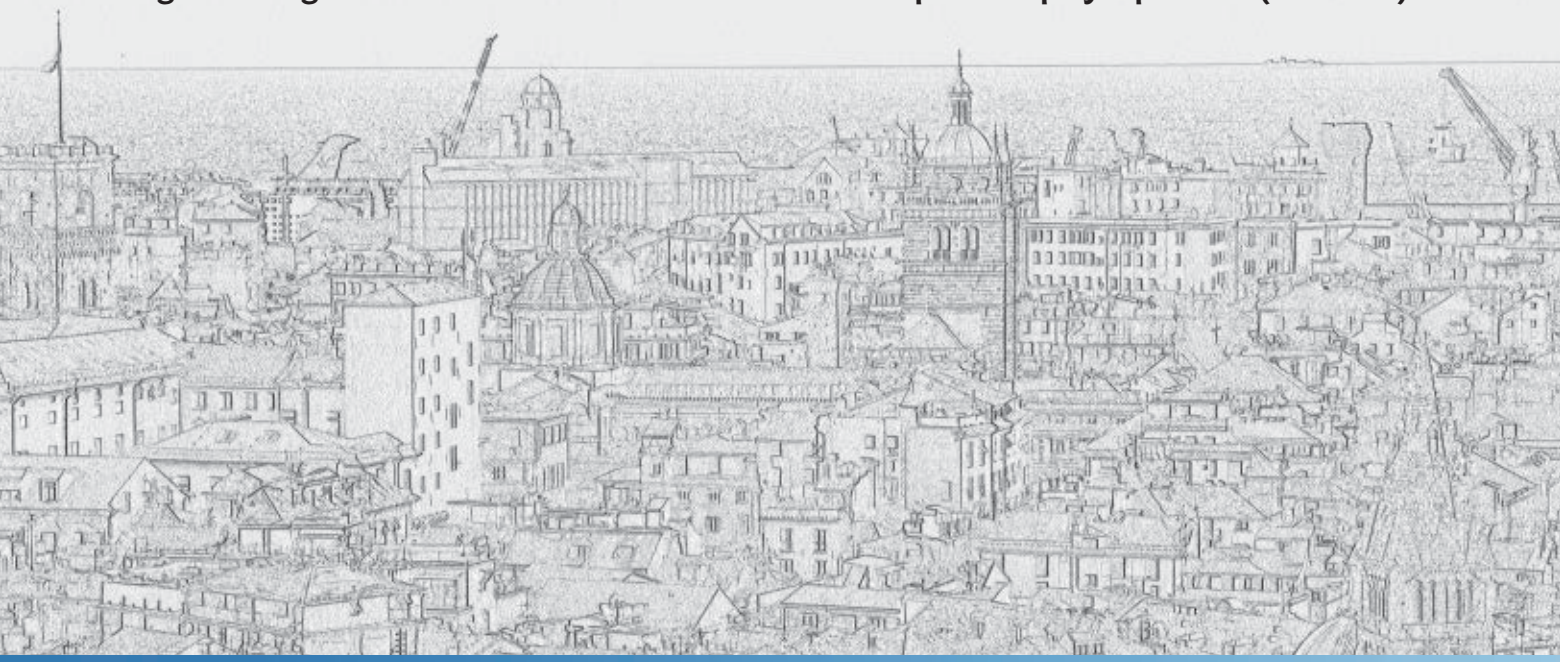
https://2019.programming-conference.org

The International Conference on the Art, Science, and Engineering of Programming, ‹Programming› for short, is a new conference focused on programming topics including the experience of programming.

‹Programming› seeks papers that advance knowledge of programming on any relevant topic, including programming practice and experience.

The Art, Science, and Engineering of Programming accepts papers that advance knowledge of programming. Almost anything about programming is in scope, but in each case there should be a clear relevance to the act and experience of programming. Additionally, papers must be written in a scholarly form. Scholarly works are those that describe ideas in the context of other ideas that are already known, so to contribute to the systematic and long-standing chaining of knowledge.

**‹Programming› 2019 will be co-located with the European Lisp Symposium (ELS 2019)**



**General Chair**
Davide Ancona, University of Genova

**Local Organizing Chair**
Elena Zucca, University of Genova

**Program Chair**
Matthew Flatt, University of Utah

**Workshops Chairs**
Walter Cazzola, University of Milano
Stefan Marr, University of Kent

**Program Committee**
Anya Helene Bagge
Mehdi Bagherzadeh
Walter Cazzola
Ravi Chugh
Joeri De Koster
Christos Dimoulas
Susan Eisenbach
Richard P. Gabriel
Jeremy Gibbons
Matthew Flatt
Michael Greenberg
Philipp Haller
Robert Hirschfeld
Eunsuk Kang
Stephen Kell
Stefan Marr
Tamara Rezk
Joshua Sunshine
Steffen Zschaler

Dibris
UNIVERSITÀ DEGLI STUDI DI GENOVA

AOSA