COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

09/2017 VOL.60 NO.09

Moving Beyond the Turing Test with the Allen AI Science Challenge

Association for Computing Machinery

CAN-CWiC
Canadian Celebration of Women in Computing

http://www.can-cwic.ca/

The ACM Canadian Celebration of Women in Computing

November 3-4, 2017
Montreal, QC at Le Centre Sheraton Hotel

The Canadian Celebration of Women in Computing 2017

Registration starting September 1st, 2017

Come celebrate with us at the largest gathering of Women in Computing in Canada!
The conference will feature prominent keynote speakers, panels, workshops, presentations and posters, as well as a programming challenge and a large career fair.

For more information contact us at cancwic@gmail.com

UQÀM | Université du Québec à Montréal

ElleCode ACM-WS UQAM

acm Association for Computing Machinery

Previous A.M. Turing Award Recipients

1966 A.J. Perlis
1967 Maurice Wilkes
1968 R.W. Hamming
1969 Marvin Minsky
1970 J.H. Wilkinson
1971 John McCarthy
1972 E.W. Dijkstra
1973 Charles Bachman
1974 Donald Knuth
1975 Allen Newell
1975 Herbert Simon
1976 Michael Rabin
1976 Dana Scott
1977 John Backus
1978 Robert Floyd
1979 Kenneth Iverson
1980 C.A.R Hoare
1981 Edgar Codd
1982 Stephen Cook
1983 Ken Thompson
1983 Dennis Ritchie
1984 Niklaus Wirth
1985 Richard Karp
1986 John Hopcroft
1986 Robert Tarjan
1987 John Cocke
1988 Ivan Sutherland
1989 William Kahan
1990 Fernando Corbató
1991 Robin Milner
1992 Butler Lampson
1993 Juris Hartmanis
1993 Richard Stearns
1994 Edward Feigenbaum
1994 Raj Reddy
1995 Manuel Blum
1996 Amir Pnueli
1997 Douglas Engelbart
1998 James Gray
1999 Frederick Brooks
2000 Andrew Yao
2001 Ole-Johan Dahl
2001 Kristen Nygaard
2002 Leonard Adleman
2002 Ronald Rivest
2002 Adi Shamir
2003 Alan Kay
2004 Vinton Cerf
2004 Robert Kahn
2005 Peter Naur
2006 Frances E. Allen
2007 Edmund Clarke
2007 E. Allen Emerson
2007 Joseph Sifakis
2008 Barbara Liskov
2009 Charles P. Thacker
2010 Leslie G. Valiant
2011 Judea Pearl
2012 Shafi Goldwasser
2012 Silvio Micali
2013 Leslie Lamport
2014 Michael Stonebraker
2015 Whitfield Diffie
2015 Martin Hellman
2016 Sir Tim Berners-Lee

ACM A.M. TURING AWARD NOMINATIONS SOLICITED

Nominations are invited for the 2017 ACM A.M. Turing Award. This is ACM's oldest and most prestigious award and is given to recognize contributions of a technical nature which are of lasting and major technical importance to the computing field. The award is accompanied by a prize of $1,000,000. Financial support for the award is provided by Google Inc.

Nomination information and the online submission form are available on: http://amturing.acm.org/call_for_nominations.cfm

Additional information on the Turing Laureates is available on: http://amturing.acm.org/byyear.cfm .

The deadline for nominations/endorsements is January 15, 2018.

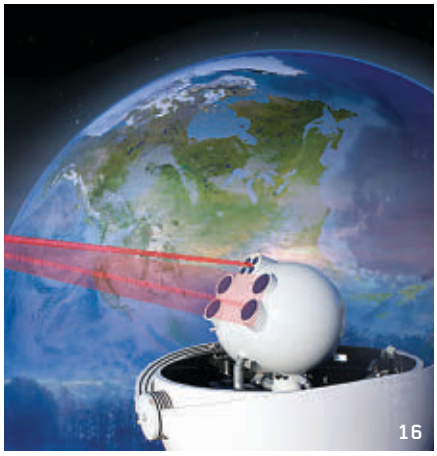
For additional information on ACM's award program please visit: www.acm.org/awards/

Association for Computing Machinery

Departments

Last Byte

News



Viewpoints

IMAGE COURTESY OF NASA

09/2017
VOL. 60 NO. 09

Practice

Articles' development led by acmqueue
queue.acm.org

Contributed Articles

Watch the authors discuss their work in this exclusive *Communications* video. https://cacm.acm.org/videos/moving-beyond-the-turing-test

Review Articles

Watch the author discuss his work in this exclusive *Communications* video. https://cacm.acm.org/videos/security-in-high-performance-computing-environments

Research Highlights

About the Cover: The Turing Test has long served as the imposing benchmark for artificial intelligence technology. Last year, researchers at the Allen Institute for Artificial Intelligence took a different route by devising a challenge that tested whether machines could handle the reasoning and understanding needed to complete an eighth-grade science test. See their results on p. 60. Cover photo by Andrey Popov, with robot illustration by Peter Crowther Associates.

PHOTO BY TAFFPIXTURE. ROBOT ILLUSTRATION BY PETER CROWTHER ASSOCIATES

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 U.S.A.

Association for Computing Machinery

DOI:10.1145/3125780 Simson Garfinkel, Jeanna Matthews, Stuart S. Shapiro, and Jonathan M. Smith

Toward Algorithmic Transparency and Accountability

Algorithms are replacing or augmenting human decision making in crucial ways. People have become accustomed to algorithms making all manner of recommendations, from products to buy, to songs to listen to, to social network connections. However, algorithms are not just recommending, they are also being used to make big decisions about people's lives, such as who gets loans, whose résumés are reviewed by humans for possible employment, and the length of prison terms. While algorithmic decision making can offer benefits in terms of speed, efficiency, and even fairness, there is a common misconception that algorithms automatically result in unbiased decisions. In reality, inscrutable algorithms can also unfairly limit opportunities, restrict services, and even improperly curtail liberty.

Information and communication technologies invariably raise these kinds of important public policy issues. How should self-driving cars be required to act? How private is information stored on a cellphone? Can electronic voting machines be trusted? How will the increasing uses of automation in the workplace impact workers? Since its founding, ACM's members have played a leading role in discussing these issues within the computing profession and with policymakers.

The ACM U.S. Public Policy Council (USACM) was established in the early 1990s as a focal point for ACM's interactions with U.S. government organizations, the computing community, and the public in all matters of U.S. public policy related to information technology. USACM came to prominence during the debates over cryptography and key escrow technology. Today, USACM continues to make public policy recommendations that are based on scientific evidence, follow recognized best practices in computing, and are grounded in the ACM Code of Ethics. It has established a reputation as a non-partisan, principled, and independent source of scientific and technical expertise, free from the influence of product vendors or other vested interests.

More recently, the ACM Europe Council Policy Committee (EUACM) has been doing the same in Europe. USACM and EUACM, both separately and jointly, provide information and analysis to policymakers and the public regarding important societal issues involving IT, including *algorithmic transparency and accountability*.

USACM and EUACM have identified and codified a set of principles intended to ensure fairness in this evolving policy and technology ecosystem.[a] These are: (1) awareness; (2) access and redress; (3) accountability; (4) explanation; (5) data provenance; (6) auditability; and (7) validation and testing.

Awareness speaks to educating the public regarding the degree to which decision making is automated. *Access and redress* means there is a way to investigate and correct erroneous decisions. *Accountability* rejects the common deflection of blame to an automated system by ensuring those who deploy an algorithm cannot eschew responsibility for its actions. *Explanation* means the logic of the algorithm, no matter how complex, must be communicable in human terms.

As many modern techniques are based on statistical analyses of large pools of collected data, decisions will be influenced by the choice of datasets for training, and thus knowing the data sources and their trustworthiness—that is, their *provenance*—is essential. *Auditability* for a decision-making system requires logging and record keeping, for example, for dispute resolution or regulatory compliance. Finally, *validation and testing* on an ongoing basis means that techniques such as regression tests, vetting of corner cases, or red-teaming strategies used in computer security should be employed to increase confidence in automated systems.

As organizations deploy complex algorithms for automated decision making, system designers should build these principles into their systems. In some cases, doing so will require additional research. For example, how to design and deploy large-scale neural networks while ensuring compliance with laws prohibiting discrimination against legally protected groups? This is especially crucial given the ability to infer characteristics such as gender, race, or disability status even if the computer system is not provided with that data directly. How should information on automated decisions be logged to ensure auditability? How can the operation of these networks be explained to technologists and non-technical policymakers alike?

One model for moving forward may be self-regulation by industry. Our experience, however, is that self-regulation is only possible when there is a consensus on a set of relevant standards. We hope our principles can serve as input to such an effort. If policymakers determine regulation is necessary, our principles are available, potentially in the way that the Code of Fair Information Practices provided a basis for decades of privacy regulation around the world.

USACM and EUACM seek input and involvement from ACM's members in providing technical expertise to decision makers on the often difficult policy questions relating to algorithmic transparency and accountability, as well as those relating to security, privacy, accessibility, intellectual property, big data, voting, and other technical areas. For more information, visit www.acm.org/public-policy/usacm or www.acm.org/euacm. C

a https://www.acm.org/binaries/content/assets/public-policy/2017_usacm_statement_algorithms.pdf

The authors are members of the ACM U.S. Public Policy Council, for which **Stuart S. Shapiro** (s_shapiro@acm.org) serves as chair.

DOI:10.1145/3130331

Vinton G. Cerf

Take Two Aspirin and Call Me in the Morning

I use a lot of metaphors in this column and this one is about security. Security is much on my mind these days along with safety and privacy in an increasingly online, programmed world. There is surely little doubt that we are at risk as cyber-attacks increase in scope, scale, and complexity. Our lives are made complex by some of the responses: "Oh, you want to log into this service? what's your username and password? OK. Now go to your mobile to get a second password that I have sent you. You don't have cell service where you are? Too bad." I am *not* dissing two-factor authentication as I am a huge proponent, but I have experienced situations like this, or a dead battery and the frustrations are material. At that point, the system might turn to "answers to secret questions," but that opens up the possibility that your choices of questions and answers are discoverable with a search of the World Wide Web. Ugh.

So where does this leave us? I am fascinated by the metaphor of cyber security as a public health problem. Our machines are infected and they are sometimes also contagious. Our reactions in the public health world involve inoculation and quarantine and we tolerate this because we recognize our health is at risk if other members of society fail to protect themselves from infection. Sadly, virus detection seems to be closing the barn door after the horses have left, to mangle a metaphor. Zero Day attacks cannot be detected with previously cataloged viral signatures, for example. They may help, but perhaps not enough.

One wonders whether we should take the metaphor more seriously and quarantine computers showing signs of infection until they have been purged of their viral load? Of course, that raises the question "How do you know that computer or IOT device is infected?" and "How do you cleanse it?" Answering these questions might take you into potential privacy-violating territory: suppose your computer keeps track of every domain name and IP address it has interacted with. Could you use this list as a detector of potential hazard? Could you go to a service and say "Here's where I have been—am I at risk?" Alternatively, you might download a blacklist of bad sites and addresses and compare to your list of places. We've seen some of the negative side effects of spam blacklists so I am not sure this would work, to say nothing of the question: "Quis custodiet ipsos custodes?"[a]

I do wonder whether machine learning might be useful. Could my computer generate a profile of "normal" Internet interactions and warn me about unusual ones? Will the false alarm rate drive me crazy? How would I know if something is a false alarm? Is there anything like a center for disease control in this space? Google acquired a company called Virustotal[b] a few years ago that maintains a library of viral profiles that allows users to check whether particular URLs or files carry malware. Another site, Stopbadware.org, helps infected websites rid themselves of viral load. There are, of course, a number of companies that offer anti-virus detection software that tries to detect malware as it is encountered or ingested into a computer. So far, these efforts have had only limited success and lead me to wonder whether there are more effective ways of discovering infection by way of behavioral observation.

It is tempting to imagine a home router/firewall that does sophisticated, machine-learned observation to protect programmable devices at home, but since our laptops, mobiles, and other programmed devices roam with us, they really need an on-board detection system (or logging system?) to protect while on the road.

Perhaps we all need to get into a cyber-hygiene habit and run our devices through regular infection checks? And we surely need much better tools with which to detect and combat this endless escalation. We could also do with better user training and services to avoid unsafe places on the Internet and poor security practices that lead to compromise. While I am not advocating for an Internet driver's license, the preparation for such a metaphorical exam might do us all some good.

a Roughly, "Who will watch the watchmen?"
b https://www.virustotal.com

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

DOI:10.1145/3122847

Moshe Y. Vardi

Divination by Program Committee

Divination is the practice of an occultic ritual as an aid in decision making. It has old historical roots. According to the biblical book of Samuel I, in the 11th century BCE, the Hebrew King Saul sought wisdom from the Witch of Endor, who summoned the dead prophet Samuel, before his impending battle with the Philistines. Alexander the Great, after conquering Egypt in 332 BCE, visited the Oracle of Amun at the Siwa Oasis to learn about his future prospects. Divination can be practiced in many ways, including sortilege (casting of lots), reading tea leaves or animal entrails, random querying of texts, and more. Divination has been dismissed as superstition since antiquity; the Greek scholar Lucian derided divination already in the 2nd century CE. Yet the practice persists.

Developments in mathematics and in computer science in the 20th century shed new light on the power of divination. Unless we believe that divination truly allows us to consult the divine, we can view it simply as a form of randomization, which is recognized as a powerful construct in game theory and algorithm design. The classical game-theoretic example is the game of Rock-Scissors-Paper in which there is no Nash equilibrium of pure strategies, but there is a Nash equilibrium in which both players choose their actions uniformly at random. The classical Dining Philosophers Problem has no symmetric distributed deterministic solution, but, as shown by Michael Rabin, has such a solution if we allow randomization. The essential insight is that randomization is a powerful way to deal with incomplete information. Thus, as realized by the anthropologist Michael Dove in the 1970s, when the Kantu people of Borneo use birdwatching to decide which sites to farm and which sites to leave fallow, they are simply randomizing in the face of uncertainty about rain, pests, and more, but this randomization comes with a belief in the divine source of the decision. (See essay by Michael Sulson at https://goo.gl/RYb264.)

But what does this have to do with program committees? In 2014, the Neural Information Processing Systems Foundation (NIPS) Conference split the program committee into two independent committees, and then subjected 10% of the submissions—166 papers—to decision making by both committees. The two committees disagreed on 43 papers. Given the NIPS paper acceptance rate of 25%, this means that close to 60% of the papers accepted by the first committee were rejected by the second one and vice versa. (See analysis by Eric Price at https://goo.gl/fy5jLR.) This high level of randomness came as a surprised to many people, but I have found it quite expected. My own experience is that in a typical program-committee meeting there is broad agreement for acceptance about the top 10% of the papers, as well as broad agreement rejections about the bottom 25% of the papers. For the other 65% of the submissions, there is no agreement and the final accept/reject decision is fairly random. This is particularly true when the accept/reject decision pivots on issues such as significance and interestingness, which can be quite subjective. Yet, we seem to pretend that this random decision reflects the deep wisdom of the program committee.

I believe the NIPS experiment should not only teach us some humility, but should also suggest that we may want to reconsider the basic modus operandi of program committees. The standard approach in such committees can be viewed as "guilty until proven innocent." We expect only 25%–35% of the papers to be accepted, so the default decision is to reject unless there is strong agreement to accept. But the reality is that a different committee may have reached a different decision on the majority of accepted papers. Is it wise to reject papers based essentially on the whim of the program committee? If we switch mode to "innocent until proven guilty," we would reject only papers on which there is strong agreement to reject, and accept all other papers.

Beyond the increased fairness of "innocent until proven guilty," this approach would also increase the efficiency of the conference-publication system. A high rejection rate means that papers are submitted, resubmitted, and re-resubmitted, resulting in a very high reviewing burden on the community. It also results in the proliferation of conferences, which fragments research communities. As I argued in an earlier editorial (https://goo.gl/dUMkwZ), I believe the proper way to adapt to the growth of the computing research is to grow our conferences rather than proliferate conferences.

NIPS should be lauded for applying the "publication method" to scientific inquiry. It is up to the computing-research community to draw the conclusions and act accordingly!

Follow me on Facebook, Google+, and Twitter. C

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

DOI:10.1145/3128899

Computational Thinking Is Not Necessarily Computational

I APPLAUD PETER J. DENNING'S Viewpoint "Remaining Trouble Spots with Computational Thinking" (June 2017), especially for pointing out the subject itself is often characterized by "vague definitions and unsubstantiated claims"; "computational thinking primarily benefits people who design computations and . . . claims of benefit to nondesigners are not substantiated"; and "I am now wary of believing that what looks good to me as a computer scientist is good for everyone." Moreover, the accompanying table outlined various historic definitions of "computational thinking," including a comparison of what Denning called the "new" and the "traditional" view of the subject. However, my own interest in computational thinking differs somewhat from Denning's. First, I question the legitimacy of the term "computational" itself. Why say it, when the very subject is "computers" and the chief academic approach to their study is "computer science"? If one looks at how computers are actually used, it may come as a surprise to learn that few such uses actually involve computing. For example, applications that deal with scientific and engineering problems are of course heavily computing-focused, but, last I heard, they constitute only approximately 20% of all applications being developed worldwide. The most predominant applications—those for business—involve little computation beyond arithmetic. And systems programs like operating systems and compilers, the focus of much computer science study, historically at least, involve little or no computation and primarily concern manipulating information rather than numbers.

The problem is that computational-thinking enthusiasts, as Denning wrote, are driven to spread the subject across all academic majors. I certainly believe in the importance of programming and using computers for the variety of applications for which they provide benefit and that educational systems worldwide should provide the knowledge and skills that would help students move into the field, should that be their preference. But should computational thinking also be taught to artists, writers, poets, physicians, and lawyers? Not as I see it . . .

The faulty thinking behind the "computer science for all" approach to pedagogy is best seen in Denning's table, labeled "Traditional versus New Computational Thinking." Its entry on "domain knowledge" suggested traditionalists see domain knowledge as vitally important to the person doing the computational thinking, while "new" thinking says the importance of computational thinking is domain-independent. As a practicing programmer who has dabbled in many different application domains over a long professional career, I see it as beyond understanding how anyone could fail to see the importance of deeply knowing a domain to being able to solve problems in that domain.

Robert L. Glass, Toowong, Australia

Author Responds:

Computational thinking is the habits of mind developed from designing computations. The meaning of computation has evolved from the 1960s "sequence of states of a computer executing a program" to today's "evolution of an information process." This changed meaning reflects the ever-expanding reach of computing into all sectors of work and life. Many of today's most popular apps feature computations well beyond arithmetic, as in, say, facial recognition, speech transcription, driverless cars, and industrial robots. The computational thinking developed by those who worked on these achievements is much more powerful than the handful of programming concepts offered as the definition of "new CT."

Peter J. Denning, Monterey, CA

Time to Retire 'Computational Thinking'?

Peter J. Denning asked, "What is computational thinking?" in his Viewpoint (June, 2017), then quoted the following definition by Al Aho: "Abstractions called computational models are at the heart of computation and computational thinking. Computation is a process that is defined in terms of an underlying model of computation, and computational thinking is the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms." But as Aho's definition is highly circular, it reveals very little.

All disciplines rely on models. The only specifically computational word here is "algorithms." If we replaced it with similar words, like "procedures" or "sequences," we would arrive at such vacuous "definitions" as, say, "Medicine is a process that is defined in terms of an underlying model of medicine, and medical thinking is the thought processes involved in formulating problems so their solutions can be represented as medical steps and procedures." And "Drama is a process that is defined in terms of an underlying model of drama, and dramatic thinking is the thought processes involved in formulating problems so their solutions can be represented as dramatic steps and sequences." One could analogously "define" musical thinking, artistic thinking, chemical thinking, and so forth.

Unless somebody can come up with a more insightful definition, it is indeed time to retire "computational thinking."

Lawrence C. Paulson,
Cambridge, England

Toward a True Measure of Patent Intensity

In their article "How Important Is IT?" (July 2017), Pantelis Koutroumpis et al. described a methodology for assessing the importance of information and communications technologies (ICTs) compared to non-ICT technologies, using PatStat, a dataset from the European Patent Office of 90 million patents awarded from 1900 to 2014. Controlling for variables (such as patent office, year

of grant, and patent family), they concluded ICT patents are more influential than non-ICT patents because they receive significantly more citations and a considerably higher PageRank.

When one publication (not just those involving patents) is cited more often than some other publication, the more-cited one is thus more influential. However, patent publications are unique because they not only describe novel systems and methods but also hold commercial value and represent licensable assets for their holders. A patent may be cited hundreds of times yet still have relatively low financial value; on the other hand, a patent may be cited only rarely yet reflect enormous valuation.

Consider that in 2013, Kodak, the company that invented the digital camera, sold its portfolio of 1,100 digital photography-related patents to multiple licensees for $525 million (or $477.3K per patent). Earlier, Google bought Motorola Mobility and its 17,000 patents for $12.5 billion (or $735.3K per patent), and Microsoft acquired 800 patents from AOL for $1.06 billion (or $1.33M per patent). Snap paid the exceptional price of $7.7 million for Mobli's Geofilters patent, believed by TechCrunch to be the highest amount ever paid for a patent from an Israeli tech company. However, the valuations of most patents are unknown until they are indeed auctioned or sold off. For instance, ICT-related patents (such as those involving Google's and Microsoft's methods for faster Internet browsing)[1] may have impressive valuations, but those valuations are difficult to predict before actually being auctioned or sold off.

Considering non-ICT patents, the revenue streams of several pharmaceutical companies depend on patents and their corresponding expiration dates, and one patent could be worth billions over the course of its licensing period. Notable patented medications include Pfizer's Lipitor (for lowering fatty acids known as lipids), Bristol-Myers Squibb's Plavix (for preventing heart attacks and strokes), and Teva's Copaxone (for treating multiple sclerosis). Other non-ICT patents that have significantly and directly improved people's lives are cited only rarely, including those related to agriculture, transportation, and creation of new materials.

In the most recent U.S. Patent and Trademark Office's economy update,[2] the non-ICT "basic chemicals" category ranked first, with $64.5 billion in merchandise exports of selected intellectual-property-intensive industries, while "semiconductors and electronic components" was second at $54.8 billion. Most industries involve non-ICT technology. As for "patent intensity," or the ratio of patents to employees measured as patents/thousand jobs, "computer and peripheral equipment" and "communications equipment" topped the list, though this was due directly to the relatively high number of patents issued in the industry versus the industry's relatively low number of employees. Conclusions regarding level of influence of ICT technologies versus other types of technologies should thus be reported with care when a comparison is based solely on number of inventions and citations.

If such influence is indeed the basis for a comparison, then additional covariates should be controlled for, including the mean estimated valuation per patent, number of employees in the industry, and additional financial and industry-specific characteristics.

References

1. Kartoun, U. A user, an interface, or none. *Interactions 24*, 1 (Jan.-Feb. 2017), 20–21.
2. U.S. Patent and Trademark Office. *Intellectual Property and the US Economy: 2016 Update*. U.S. Patent and Trademark Office, Washington, D.C., 2016; https://www.uspto.gov/sites/default/files/documents/IPandtheUSEconomySept2016.pdf

Uri Kartoun, Cambridge, MA

Authors Respond:

Although there may be some correlation between patent price and technological influence, the relationship is neither clear nor systematic. Patent prices are more likely driven by how incremental/radical/breakthrough it is, whether its value is standalone or as part of a bundle, projected commercialization timescale, cost versus risk, bidder's experience, patent age, rate of technological change, and substitution and reverse-engineering risk, to say nothing of broader economic factors. Perhaps our technological-influence measure could thus be used to help understand patent pricing.

Pantelis Koutroumpis, London, U.K.,
Aija Leiponen, Ithaca, NY, and
Llewellyn D W Thomas, London, U.K.

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.

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:10.1145/3121430 http://cacm.acm.org/blogs/blog-cacm

Assuring Software Quality By Preventing Neglect

Robin K. Hill suggests software neglect is a failure of the coder to pay enough *attention and take* enough *trouble to ensure software quality.*

Robin K. Hill
The Ethical Problem of Software Neglect
http://bit.ly/2roEDf1
May 31, 2017

Ethical concern about technology enjoys booming popularity, evident in worry over artificial intelligence, threats to privacy, the digital divide, reliability of research results, and vulnerability of software. Concern over software shows in cybersecurity efforts and professional codes.[1] The black hats are hackers who deploy software as a weapon with malicious intent, and the white hats are the organizations that set safeguards against defective products. But we have a gray-hat problem—neglect.

My impression is that the criteria under which I used to assess student programs—rigorous thought, design, and testing, clean nested conditions, meaningful variable names, complete case coverage, careful modularization—have been abandoned or weakened. I have been surprised to find, at prestigious institutions working on open-source projects, that developers produce no documentation at all, as a matter of course, and that furthermore, during maintenance cycles, they do not correct the old source code comments, seeing such edits as risky and presumptuous. All of these people are fine coders, and fine people. Their practices seem oddly reasonable in the circumstances, under the pressure of haste, even while those practices degrade the understandability of the program. Couple that with the complexity of modern programs, and we conclude that, in some cases, programmers simply don't know what their code does.

Examples of software quality shortcomings readily come to mind—out-of-bounds values unchecked, complex conditions that identify the wrong cases, initializations to the wrong constant. Picture a clever and conscientious coder finishing up a calendar module before an important meeting. She knows that the test for leap years from the numeric `yyyy` value, `if (yyyy mod 4 = 0)` and `(yyyy mod 100 != 0)`, must be refined by some other rules to correct for what happens at longer periods, but this code is a prototype ... She retains the simple test, meaning to look up the specifics ... but her boss commits her code. No harm is foreseeable ... except that it turns out to interface with another module where the leap-year calculation incorporates the complete set of conditions, which is discovered to drive execution down the wrong path in some calculations. The program is designated for fixing but it continues to run, those in the know compensating for it somehow...

What sort of violation is neglect? It doesn't attack security because it occurs behind the firewall. It doesn't attack ideals of quality because no-one officially disputes those ideals. It is a failure of degree, a failure to pay *enough* attention and take *enough* trouble. Can philosophy help clarify what's wrong? An emerging theory called the *ethics of care* displaces the classical agent-centered morality of duty and justice, endorsing instead patient-centered morality as manifest real-time in relationships.[2,4]

The theory offers a contextual perspective rather than the cut-and-dried directives of more traditional views. While care can be construed as a virtue (relating to my prior post in this space[3]) or as a goal like justice, the promoters of care ethics resist a universal mandate. They may also reject this attempt to apply it to software, of all things; the heart of the matter for care ethics is the work of delivering care to a person in need.

Yet software neglect seems exactly the type of transgression addressed by the ethics of care, if we allow its reinterpretation outside of human relationships. Appeal to the theory allows us to identify the opposite of care, that is, neglect, as the quality to condemn. This yields our account of software quality as an ethical issue, especially piquant in its application of tools from the feminist foundry to the code warrior culture. But little credit is due! We are not solving the problem, only embedding it in the terms of a philosophical platform. This account raises issues in the ethics of engineering, such as individual versus corporate responsibility (and whether corporate responsibility can be rendered coherent and enforceable short of the law). For a concise summary, see Section 3.3.2, on Responsibility, in Stanford Encyclopedia of Philosophy entry on the Philosophy of Technology.[5]

The quality that has corrected for neglect in the past is professionalism, by which I mean that the expert does what's best for the client even at a cost to personal time, energy, money, or prestige—within reason! Certainly these judgments are subjective, and viable when the professional is autonomous, when that single person exercises control over the product and its quality. Counterforces in the current tech business world are (1) employment, under which most programmers are not consultants, but rather given orders by a company; and (2) collaboration, under which most software is the product of committees, in effect. Professionalism also depends on strong personal identification with disciplinary peers and pride in the group's traditions.

In the face of knotty difficulties enforcing or fostering ideals of quality, one possible resolution, odd as it may seem, is simply to acknowledge the situation, to admit to the public that software is not always reliable, or mature, or even understood. Given its familiarity with bug fixes, the public may not be unduly shocked. If we prefer to reject that fatalistic move, the pressing question is, are there some public standards that developers can and will actually follow? The collective response will determine whether software engineering is a profession. I urge all coders who wish to take pride in their jobs to read the draft professional standards,[1] which mention code quality in Section 2.1.

What sort of violation is neglect? It doesn't attack security because it occurs behind the firewall. It doesn't attack ideals of quality because no one officially disputes those ideals.

We see that ethical issues appear not only in the external social context, but in the heart of software, the coding practice itself, a gray-hat problem, if you will. We hope that the ethics of care can somehow help to alleviate those issues.

References

1. Association for Computing Machinery. *Code 2018 Project.* https://ethics.acm.org/.
2. Burton, B.K., and Dunn, C.P. *Ethics of Care.* Encyclopædia Britannica, https://www.britannica.com/topic/ethics-of-care.
3. Hill, R.K. *Ethical Theories Spotted in Silicon Valley.* Blog@CACM, March 16, 2017, https://cacm.acm.org/blogs/blog-cacm/214615-ethical-theories-spotted-in-silicon-valley/fulltext.
4. Sander-Staudt, M. *Care Ethics.* The Internet Encyclopedia of Philosophy, 2017. http://www.iep.utm.edu/care-eth/.
5. Franssen, M., Lokhorst, G., and van de Poel, I. *Philosophy of Technology.* The Stanford Encyclopedia of Philosophy (Fall 2015 Edition), Edward N. Zalta (ed.). https://plato.stanford.edu/archives/fall2015/entries/technology/.

Note: While the Web encyclopedias, as cited, provide good surveys of current philosophical views, pursuit of any ideas in depth will require reading original research.

Comments

This is possibly the most important paragraph of the article, outlining the exact problem in the industry:

"The quality that has corrected for neglect in the past is professionalism, by which I mean that the expert does what's best for the client even at a cost to personal time, energy, money, or prestige—within reason! Certainly these judgments are subjective, and viable when the professional is autonomous, when that single person exercises control over the product and its quality. Counterforces in the current tech business world are (1) employment, under which most programmers are not consultants, but rather given orders by a company; and (2) collaboration, under which most software is the product of committees, in effect. Professionalism also depends on strong personal identification with disciplinary peers and pride in the group's traditions."

It sounds like, short of working for enlightened organizations, software developers should be leaning towards more autonomy and self-ownership.

I recently read Developer Hegemony *(a very bold title!), http://amzn.to/2pA18wB, and it addresses that side of the issue by encouraging more professionalism and autonomy.*

There's already a strong movement in favor of Software Craftsmanship, and the free software and open source movements both seem to care more about quality than most companies (though they do neglect documentation sometimes). For example, we already prefer software written by recognizably smart/professional developers.

Here's hoping to more autonomy in the future and the allowance of our professionalism to counteract the neglect of software.

—Rudolf Olah

Robin K. Hill is an adjunct professor in the Department of Philosophy at the University of Wyoming.

Introducing *ACM Transactions on Human-Robot Interaction*

Now accepting submissions to ACM THRI

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

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

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

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

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

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

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

Association for Computing Machinery

Science | DOI:10.1145/3121434

Samuel Greengard

It's All About Image

Image recognition technology is advancing rapidly. Researchers are discovering new ways to tackle the task without enormous datasets.

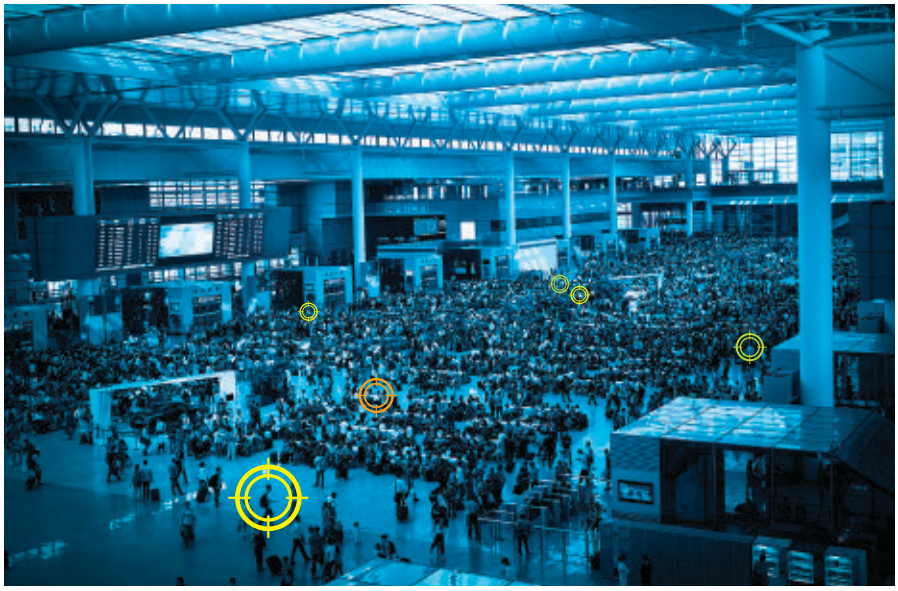


IMAGE FROM SHUTTERSTOCK.COM

Discovering the secrets of the universe is not a task for the timid and the impatient; there's a need to peer into the deepest reaches of outer space and try to make sense of distant galaxies, stars, gas clouds, quasars, halos, and black holes. "Understanding how these objects behave and how they interact gives us answers to how the universe was formed and how it works," says Kevin Schawinski, an astrophysicist and assistant professor in the Institute for Astronomy at ETH Zurich, the Swiss Federal Institute of Technology.

The problem is that traditional tools such as telescopes can see only so far, even with radical advances in optics and the placement of observatories in space, where they are free of the light and dust of Earth. For instance, the Hubble Telescope changed the way astrophysicists and astronomers viewed deep space by delivering far clearer images than previously possible. Of course, in this context, distance and time are inextricably linked. "But the images still do not allow us to see as far back in time as we would like," Schawinski says. "The farther we can see, the more we can understand about the origins of the universe and how it has evolved."

Enter computer image recognition, artificial neural networks, and data science; together, they are changing the equation. As huge volumes of data stream in, they are able to find answers to previously unfathomable questions. In recent years, scientists have begun to train neural nets to analyze data from images captured by cameras in telescopes located on Earth and in space. In many cases, the resulting machine-based algorithms can sharpen blurs and identify distant objects better than humans can.

"Data science and big data are revolutionizing many areas of astrophysics," says François Lanusse, a postdoctoral researcher in the McWilliams Center for Cosmology at Carnegie Mellon University.

Indeed, the combination of more data, advances in data science, and new methods that allow researchers to easily and cheaply train neural networks is allowing scientists to boldly see where they have never seen before. No less important, these advances are not limited to astrophysics and astronomy; they have touched an array of other fields and have advanced autonomous vehicles, robots, drones, smartphones and more. They're also being used to better understand everything

from how linguistic patterns contribute to racism to identifying the potential severity of hurricanes as they form.

Says Jeff Clune, an assistant professor of computer science at the University of Wyoming, "Until very recently, computers did not see and understand the world very well. The ability to train neural nets quickly and easily is transforming image recognition and enabling remarkable breakthroughs."

Researchers are turning to convolutional systems modeled from human visual processing, and generative systems that rely on a statistical approach.

Picture Perfect

Artificial neural nets are nothing new. The concept originated in the 1940s and researchers have experimented with them for the last quarter-century. Yet it was only over the last few years that the technology has matured to the point where computer image recognition and other artificial intelligence (AI) capabilities have become viable. Using anywhere from one to sometimes hundreds of graphical processing units (GPUs), these training networks—which function in a similar way to neural pathways in the human brain—recognize patterns in data that other computing systems cannot. Layered nodes learn from each other—and from other networks—much like the way children learn. Remarkably, because of their overall complexity, nobody knows exactly how each trained artificial neural net produces its useful results.

Rapid advancements in neural nets and deep learning are a result of several factors, including faster and better GPUs, larger nets with deeper layers, huge labeled datasets to train on, new and different types of neural nets, and improved algorithms. Typically, for computer image recognition, researchers feed lots of pictures of things—motorcycles, chimpanzees, trees, or space objects, for example—into the system so the neural net can learn what an object looks like and how to differentiate it from others. If a researcher is training the neural net to recognize animals, the system tends to learn faster and better if old data is transferred to the new task. For instance, if the original task was to identify lions and zebras, adding this data to the job of identifying elk and bears will help.

The system succeeds because there is now a shared knowledge between the two paths. "Already being good at one task makes a neural network faster and better at learning the second task," Clune explains. "The system already has a basic understanding of things that are common to both tasks, such as eyes, ears, legs, and fur." As training proceeds and a neural net becomes smarter, it can identify photos and other images it has never seen before. For example, Clune has achieved an accuracy rate as high as the 96.6% in the neural net compared to the 40,000+ humans who volunteered to label the same images. Others have found that the neural nets actually outperform humans. Remarkably, "In most cases, we can train a neural net within a couple of days," he says.

Of course, this doesn't mean that all systems are equally effective–and that the results are consistently useful. There's also the goal of pushing the boundaries of computer image recognition further. At present, researchers train systems using labels. This means designating images for one type of animal 'a lion' and another 'a zebra,' or one galaxy 'a spiral' and another 'an elliptical.' The problem with this approach is that it's time consuming and sometimes expensive. What is more, "sometimes you don't have labels, or they are noisy labels," says Ce Zhang, an assistant professor in the Systems Group at ETH Zurich. For instance, a "cougar" label might confuse the system if it is presented with both the car and the animal.

Consequently, researchers are interested in an emerging area of deep learning that relies on different training methods, as well as unsupervised learning. University researchers as well as companies such as Alphabet, which operates Google Brain and DeepMind, have begun to study this space. They are turning to convolutional systems modeled after the visual processing that takes place in humans, and generative systems that rely on a more conventional statistical-based approach to learn the features of a dataset.

The end goal? "We want to just hand the computer the data and the algorithm and have it deliver results," Schawinski says. "This type of capability would revolutionize astrophysics, but also science in general."

A Sharper Focus

Advances in AI are now pushing the boundaries of neural nets and deep learning into an almost sci-fi realm, though the results produced by these systems are very real. Consider: Clune now uses generative systems to produce artificial images that look completely real to the human eye. These photo-realistic images range from birds and insects to mountains and even vehicles. He describes the technology as a "game changer." Remarkably, over time, certain neurons in the deep learning network become better than others at recognizing and generating specific things, such as eyes, noses, bugs, or volcanoes. "The system actually figures out what it needs to recognize and know and allocates neurons to these concepts automatically," he says.

To be sure, generative networks have value that extends beyond producing artificial images for art, video games, or augmented reality/virtual reality (AR/VR). Researchers have begun to use generative networks in competition with image-recognition networks to generate even more accurate results. Within this scenario, the generator network creates fake images and the image recognition network, known as a discriminator, analyzes the images and attempts to separate the real from the fake images. The discriminator later checks the validity of its findings and uses those results to further refine its algorithm. Over time, the discriminator becomes smarter and tells the generator how to adapt its output to generate even more realistic images.

The advantage of this approach is

that the discriminator, referred to as a *generative adversarial net* (GAN), learns over time what matters most in the image, Zhang says. At a certain point, the system displays almost human-like intuition, he says; "results improve significantly." Interestingly, this approach not only improves the quality of image detection, it may also trim the time required to train a network by reducing the number of images—essentially the volume of data—required to obtain useful results. Says Zhang: "An interesting question is how can we lower the requirement of a neural network in terms of how much data it needs to achieve the current level of quality?"

Another step is to make today's artificial neural nets easier to use. The technology is still in its infancy and researchers often struggle to use tools and technology effectively. In some cases, they have to work with multiple nets in an iterative fashion to find one that works best. As a result, Zhang has developed a software program, *ease.ml*, that configures deep learning neural networks in a more automated and efficient way. This includes optimizing components such as CPUs, GPUs, and FPGAs and providing a declarative language for better managing algorithms.

"Right now, the user needs to deal with a lot of different decisions, including the type of neural net they want to use. There may be 20 different neural nets available for the same task. Choosing the right model and reducing complexity is important," he explains.

Already, the software, combined with other deep learning techniques—including an algorithm called ZipML that reduces data representation without reducing accuracy—has cut noise and sharpened images significantly for the astrophysics group at ETH Zurich. As a result, Schawinski and others can now peer more deeply into the universe.

"Unlike other areas of science, we cannot run experiments in a lab and simply analyze the results," ETH Zurich explains. "We are dependent on telescopes and images to look back in time. We have to piece together all these fixed snapshots—essentially huge datasets—to gain insight and knowledge."

Adds Lanusse: "Classical methods of astronomy and astrophysics are rapidly being superseded by data science and machine learning. They not only do a job better, but they also offer new ways of looking at the data."

The view into the future is equally compelling. Lanusse says that in the coming years neural networks will drive enormous advances in fields beyond astrophysics. These systems will not only detect, recognize, and classify objects, they will understand what is taking place in an image or in a scene in real time. This, of course, could profoundly impact everything from the way autonomous vehicles operate to how medical diagnostics work. Ultimately, they will help us unlock the mysteries of our planet and the universe. They will deliver a level of understanding that wouldn't have been imaginable only a few years ago.

Says Lanusse, "Computer image recognition is advancing rapidly. We are finding ways to train networks faster and better. Every gain in speed and accuracy of even a few percent makes a profound difference in the real-world impact." C

Further Reading

Nguyen, A., Yosinski, J., Bengio, Y., Dosovitskiy, A., and Clune, J.
Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space. Computer Vision and Pattern Recognition (CVPR '17), 2017. http://www.evolvingai.org/ppgn

Lanusse, F., Quanbin, M, Li, N., Collett, T.E., Li, C., Ravanbakhsh, S., Mandelbaun, R., and Poczos, B.
CMU DeepLens: Deep Learning for Automatic Image-based Galaxy-Galaxy Strong Lens Finding. March 2017. arXiv:1703.02642. https://arxiv.org/abs/1703.02642.

Wang, K., Guo, P., Luo, A., Xin, X., and Duan, F.
Deep neural networks with local connectivity and its application to astronomical spectral data. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, 2016, pp. 002687-002692. doi: 10.1109/SMC.2016.7844646. http://ieeexplore.ieee.org/document/7844646/

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.
Generative Adversarial Networks. June 2014. eprint arXiv:1406.2661. http://adsabs.harvard.edu/cgi-bin/bib_query?arXiv:1406.2661.

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

ACM Member News

ENSURING TECHNOLOGY BEHAVES CORRECTLY

"Things should do what they are expected to do, according to a specification," says Marta Kwiatkowska, professor of computing systems at the University of Oxford. She explains that something should happen with high probability, within an appropriate or expected time or expected range. "My main focus is on developing verification techniques and model checking for probabilistic systems, which ensure software, systems, hardware, and protocols behave correctly."

Kwiatkowska has held a statutory chair in the Department of Computer Science at Oxford, and a professorial fellowship at the University's Trinity College, since 2007. Prior to that, she was a professor in the School of Computer Science at the University of Birmingham, a lecturer at the University of Leicester, and an assistant professor at Jagiellonian University in Krakow, Poland.

She earned an undergraduate degree in computer science at Jagiellonian University, writing programs on punch cards in PASCAL. Kwiatkowska then earned a master's degree from Oxford, and a Ph.D. in computer science from the University of Leicester.

Initially her research interests centered on concurrent and distributed systems, but in 1995 Kwiatkowska started working on verification techniques. Her research covers a range of applications including biological systems, DNA computations, and analyzing the behavioral correctness of pacemakers, among others.

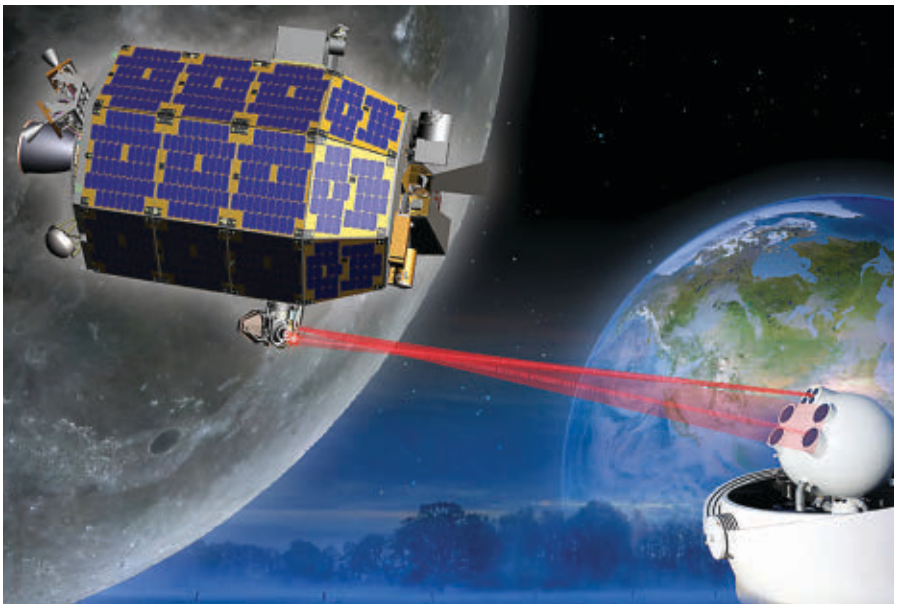
Kwiatkowska now studies autonomous systems and the application of verification techniques to robotics. "We need to develop methods to verify the correctness of the behavior of robots," she says. "I am also looking at verification for machine learning, specifically neural networks, which are now being used in perception algorithms for self-driving cars."

—John Delaney

Technology | DOI:10.1145/3121442 Gregory Mone

Broadband to Mars

Scientists are demonstrating that lasers could be the future of space communication.



Artist's conception of how a NASA spacecraft would use lasers to communicate with Earth.

IMAGE COURTESY OF NASA

In March, the U.S. National Aeronautics and Space Administration (NASA) announced that its planned Orion spacecraft, which could one day carry astronauts to the Moon and Mars, will include a new kind of communication system. Typically, manned and unmanned vehicles and probes use radio waves to send and receive information. For decades, though, scientists have been pushing toward using laser-based communications in space. Lasers are no faster, but they can deliver far more information than radio waves in the same amount of time. NASA's Apollo missions to the Moon were capable of transmitting 51kb worth of data per second, for example, but Orion's planned Laser-Enhanced Mission and Navigation Operational Services (LEMNOS) system could send back more than 80 megabytes each second from the lunar surface.

That stream could be packed with rich scientific data, or it could include ultra-high-resolution video of distant worlds. Scaled-up versions of this system could dispatch movies of dust devils, storms, or even astronauts walking on the surface of Mars. During the six-month-long trip to the Red Planet, space travelers could potentially trade videos with family members back on Earth, and mitigate the psychological toll of the long journey.

The LEMNOS project is just one of many planned or existing laser-based communications systems in orbit and beyond.

These recent and anticipated advances cannot be attributed to a single, revolutionary breakthrough, according to experts. Instead, this new age of laser-based broadband in space has resulted from steady improvements in detectors, actuators, control systems, and more.

Broadband in Orbit

The idea of laser communications in space has been around nearly since the invention of the laser itself, notes Abi Biswas, supervisor of the Optical Communications Systems group at NASA's Jet Propulsion Laboratory in Pasadena, CA. From a basic physics standpoint, Biswas says the advantage is clear: lasers occupy the higher-frequency end of the electromagnetic spectrum, relative to radio waves. That means the beam itself is much narrower. If you were to aim a beam of radio waves back at Earth from Mars, the beam would spread out so much that the footprint would be much larger than the size of our planet. "If you did the same thing with a laser," Biswas says, "the beam footprint would be about the size of California."

When those beams are sent with the same amount of power, the laser ends up concentrating more power on that receiver. "You can send many more bits of information for the same amount of power," Biswas explains. Relative to radio, laser or optical communications can transmit anywhere from 10 to 100 times as much data.

The advantages are not limited to solar system exploration; they can be applied within Earth's orbit as well. The European Space Agency (ESA) and Airbus recently put lasers to work as a broadband data transfer technology, the European Data Relay System (EDRS).

Normally, a satellite flying in low Earth orbit transmits data only when it is within view of a ground station. As a result, it may take 90 minutes for the ground station to receive data after it has been collected.

In the EDRS system, lasers are used both to send more data and to accelerate its transfer. A geostationary satellite locks onto the low-orbiting satellite via laser the moment it passes over the horizon, then remains connected as the craft soars over the hemisphere below. The observing satellite begins transmitting data via laser once the link is established. The satellite can transfer far more data this way, but it also gets that data to the ground faster. Instead of waiting for the observing satellite to fly within view of the ground station, the laser transfer begins once the craft establishes line of sight with the geostationary craft, which then transmits data to the

ground via radio. "You cut down the time or delivery of the data to the end user on the ground from hours to 10 to 20 minutes," says Michael Witting, program manager for ESA EDRS.

This speed, combined with the ability to transmit more high-resolution satellite images, will allow organizations to track the movement of ice in polar regions to help ships navigate the Arctic crossing. Officials could monitor oil spills, earthquakes, floods, and other instances in which information needs to travel quickly to disaster response teams.

The EDRS is already in use, and ESA is scheduled to launch a second satellite in 2018.

NASA has a similar project in the works, and while the link does not extend all the way to the Moon or Mars, Witting says the technical challenges were significant. The system operates over approximately 45,000 kilometers (about 28,000 miles), and each laser terminal must locate and remain locked on the other throughout flight. "It's like taking a torch from Europe and hitting a coin in New York," Witting says— all while the coin is racing at about 17,000 miles per hour.

Lasers from the Moon

As you move out to larger distances, such as the Moon or Mars, the challenge increases. Biswas compares the effort involved with hitting a target on Earth from the Moon or Mars to trying to look at a small object through a one-meter-long straw; holding that straw steady enough to keep it focused is a tremendous challenge. If not held steady and aimed accurately, the California-sized footprint of a laser beam traveling from Mars could actually miss its target on Earth, and fail to transmit the data.

Experts say the success of NASA's 2013 test of such a system, the Lunar Laser Communication Demonstration (LLCD), can be attributed to a number of advances, including improvements in the actuators that make micro-adjustments to the position of the beam, ensuring it remains on target, and advances in the control systems that determine exactly where it needs to aim. When the laser struck Earth, the beam was six kilometers wide, but the receiver was less than one meter wide. One way around this would be to build a larger receiving antenna, but the goal of the LLCD was to show that an optical communication system could work without a massive—and massively expensive—dish on the ground. "You have to figure out, how can I catch this dancing signal onto a very sensitive detector and then add very little noise?" asks Don Boroson, a research fellow in the Massachusetts Institute of Technology (MIT) Lincoln Laboratory's Communication Systems Division, and a major contributor to the LLCD.

For the Moon demonstration, Boroson says the group used an old idea known as error correction coding, which intelligently bundles in redundant bits, so you can still decipher an entire message even if you only catch part of the beam. So, if they were trying to send a message that was 10,000 bits, they'd add in another 10,000 carefully chosen redundant bits, and send 20,000 in all. Then, even if only half of that message was received, the original 10,000-bit code could still be deciphered. This approach was critical, Boroson explains; "it allowed us to have as small as possible a receiver on the ground and still do these very high data rates and make no errors. We did the lunar link with half a watt and a four-inch telescope in space, and we still did 622 megabits per second to the ground."

Making Every Photon Count

Pushing beyond satellite or lunar communication increases the technical difficulty, because the laser beam loses energy at a rate proportional to the square of the distance between transmitter and receiver. Scaling up the power used to generate the laser is not an option, Biswas explains, because the laser systems would become too large and expensive. "As you get farther and farther away, you have to improve the efficiency of your system," says Biswas. "You have to make every photon count."

Despite the challenges of larger distances, physicist Philip Lubin of the University of California, Santa Barbara, argues that lasers would still be a preferred means of communication for missions to the edges of our solar system and beyond. Lubin has been working on a project to propel miniature spacecraft beyond the solar system using a phased array of either ground- or space-based lasers. The spacecraft would have a modest laser to send back data, and Lubin says the array used to propel the craft could also be engineered to receive its messages. "If we're setting up to blast something out with lasers, then why not use that system to send something back?" he asks. That something probably will not include video, but the lasers could dispatch images and other information.

Back on Earth, larger receiving telescopes would help pick up signals from the Moon, Mars, or beyond. Currently, NASA scientists are demonstrating how laser communications systems work with small receivers, but with the kind of ground telescopes that measured 10 to 15 meters across, it would be possible to catch far more light and information. Boroson doesn't expect those receivers to be built anytime soon, but he does anticipate laser communications will be used more and more.

"It's going to happen slowly," says Boroson. "First we'll see lots of systems around the Earth, then a few systems further out in space, and then more and more. But it's all coming, it's definitely coming." C

Further Reading

Biswas, A. Piazzolla, S. Moision, B., vand Lisman, D.
Evaluation of deep-space laser communication under different mission scenarios, *Proceedings of SPIE*, 2012.

Boroson, D.M. and Robinson, B.S.
The Lunar Laser Communication Demonstration: NASA's First Step Toward Very High Data Rate Support of Science and Exploration Missions, *Space Science Reviews*, Volume 185, 2014.

Lubin, P.
A Roadmap to Interstellar Flight, *Journal of the British Interplanetary Society*, vol. 69, 2016.

Space Data Without Delay
http://bit.ly/2pcIlt2

Hemmati, H.
Deep Space Optical Communications, John Wiley & Sons, 2006.

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

Society | DOI:10.1145/3121436 Logan Kugler

Why GPS Spoofing Is a Threat to Companies, Countries

Technology that falsifies navigation data presents significant dangers to public and private organizations.

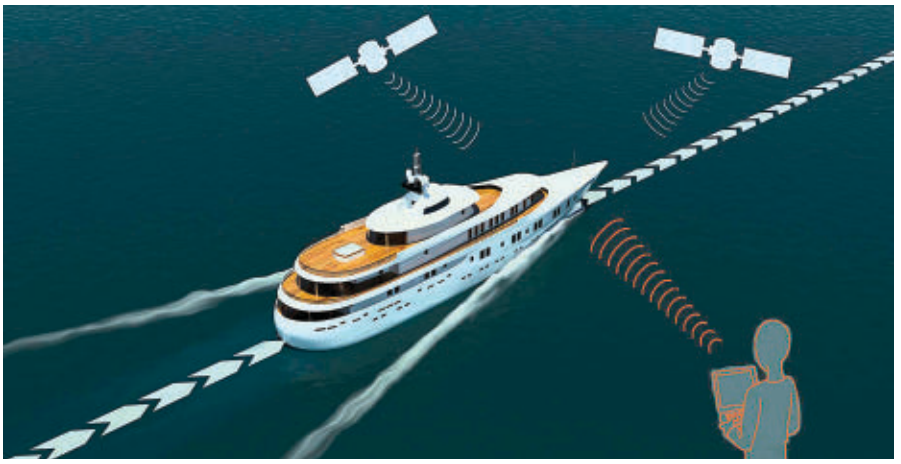
When the crew of an $80-million superyacht in the Ionian Sea checked its computer, they realized they were drifting slightly off course, likely as a result of strong currents buffeting their ship. The crew made adjustments and went back to work—without realizing they were now taking directions from a hacker.

In the bowels of the ship, Todd Humphreys, an associate professor in the Department of Aerospace Engineering and Engineering Mechanics at the University of Texas at Austin, worked with his team to feed the superyacht's crew false navigation data using a few thousand dollars worth of hardware and software.

The crew was completely unaware they were now piloting in a direction of Humphreys' choosing.

Thankfully, it was all an experiment that took place with the yacht owner's blessing. If it had been real, Humphreys could have sent the superyacht 1,000 miles off-course into the hands of a rogue government, terrorist group, or professional criminal organization—and the crew would not have realized it until it was far too late.

Welcome to the very real dangers posed by Global Positioning System (GPS) spoofing, or the dark art of convincing computers you are somewhere that you're not. It is surprisingly easy—and shockingly dangerous, because we're not prepared for it at all.



Part of an animation showing how a radio navigation research team from The University of Texas at Austin was able to successfully spoof the GPS system of an $80-million private yacht.

GPS Is Easy to Spoof

The U.S. Global Positioning System consists of 24 satellites that orbit Earth. GPS devices receive signals from the nearest satellites that allow them to determine their precise location, whether you're looking for creatures in the wildly popular Pokémon Go app, or going to war in a billion-dollar battleship. A range of GPS devices and networks are used for everything from military applications to commercial needs—and all the use cases in between.

Yet all of these systems rely on the data from the network of GPS satellites. If you can corrupt the data coming from those satellites, you can create a world of headaches for systems that rely on this data.

GPS spoofing can be performed with relatively low-cost tech, which is an expensive problem for the people, companies, and governments that trust the system implicitly. In the case of Humphreys' superyacht hacking, he and his team used about $2,000 worth of tech. Even in more advanced spoofing scenarios, the technology is still straightforward, says Dinesh Manandhar, an associate professor and GPS expert at the University of Tokyo.

"A device that can generate GPS signals is necessary. Such devices are available from GPS signal simulator device manufacturers," Manandhar explains. These devices are used to test GPS receivers in factories. As such, they can be programmed to transmit a signal that makes receivers behave any way you like.

"So far as I know, no commercial GPS receivers offer any strong defense against spoofing or even any reliable spoofing detection capability," says Humphreys.

Stealing an $80-Million Superyacht

In 2013, Humphreys, then a researcher in the Department of Aerospace Engineering and Engineering Mechanics at the Cockrell School of Engineering, was invited, along with a team of students, aboard an $80-million yacht in the Ionian Sea to test their GPS spoofing technology. Using his hardware and software rig, Humphreys managed to falsify GPS data used by the ship, effectively giving him control over the vessel.

Humphreys explained GPS receivers calculate their distance from several satellites at the same time. Each satellite has a code—called a pseudorandom noise (PRN) code—that identifies which satellite in the GPS network is broadcasting. Humphreys' spoofing equipment slowly replaced the real GPS signals with fake ones, working delicately so the ship's system did not detect an abrupt change in signal.

IMAGE COURTESY OF UNIVERSITY OF TEXAS AT AUSTIN

The spoofed GPS reported the yacht was three degrees off-course. The crew, unaware when the experiment would take place, adjusted the ship's course based on the spoofed GPS. The crew assumed it was due to natural forces such as water currents and crosswinds."

GPS spoofing can be used for all sorts of nefarious purposes. As seen with the yacht, cargo shipments are at risk, especially dangerous or high-value ones that are required to follow designated GPS routes. Geofences—or digitally proscribed boundaries—are used to protect sensitive data in many corporations; GPS spoofing could be used to access that data well out of the bounds intended.

Once you add emerging technologies, like self-driving cars, to the mix, it gets even scarier. Autonomous vehicles use GPS data at regular intervals not only to understand where they are, but also to decide where to drive passengers and cargo.

Humphreys' yacht spoofing was the first time commercial tech had been used in such an effective—and powerful—demonstration.

Now, said Manandhar, it is even easier to acquire spoofing technology. "Recently, software-based low-cost devices have become available that cost less than $1,000."

Cargo shipments are at risk from GPS spoofing, as are geofences—digitally proscribed boundaries used by many corporations to protect sensitive data.

A Problem for Governments, People

It is not just yacht owners who need to be concerned; the problem is especially acute for national governments and international bodies, which are waking up to the dangers posed by GPS spoofing.

Incredibly, Europe's Galileo global navigation satellite system—the European Union's version of GPS—operated beginning in December 2016 "with no way to protect civilian users from hacking attempts," reported ZDNet.

University of Leuven researchers Ashur and Rijmen say they have developed an authentication protocol to deter the forging of Galileo's navigation data.

The protocol, called the TESLA signature, is designed to complement location data with a cryptographic "signature," so Galileo's satellites would send both navigation data and the cryptographic signature to the receiving client. The client would not trust the data right away; only when the signature was verified would the client use the GPS data it had received.

"Using cryptography makes it hard to forge a signature, such that even an adversary that can feed the client with false data cannot forge a signature, thus the client does not use forged data," Ashur says.

This would prevent, say, spoofing the signal to hijack a self-driving car or reroute a drone that relied upon the data.

However, the Galileo system, which comes fully online in 2020, presented a unique obstacle: low bandwidth. Galileo has relatively low-bandwidth signals that make a typical approach to the problem, using public-key cryptography, impossible.

"The uniqueness of our solution is that it uses symmetric cryptography and can thus fit into the bandwidth constraints," says Ashur. The protocol is scheduled to go into effect in 2018, according to ZDNet. Until all 24 of Galileo's satellites are deployed and operational in 2020, however, the protocol will "operate in test mode."

In the meantime, manufacturers are starting to pay attention to the problem, says Humphreys. Some, like u-blox, a Swiss company that creates wireless semiconductors and modules for consumer, automotive, and industrial markets, offer anti-spoofing measures such as the capability to detect fake global navigation satellite system (GNSS) signals, as well as a message integrity protection system to prevent "man in the middle" attacks.

Humphreys also points to the U.S. Department of Homeland Security's recent document on anti-spoofing, "Improving the Operation and Development of Global Positioning System (GPS) Equipment Used by Critical Infrastructure," as a sign that the right parties are taking GPS spoofing seriously.

Manandhar has developed anti-spoofing methodologies for Japanese satellites that may be used in the next generation to be sent into orbit, he says. He recommends that major navigation data provider countries like the U.S., Japan, the European Union, China, and India conduct official joint discussions on the security of their systems at the International Committee on Global Navigation Satellite Systems, an organization under the umbrella of the United Nations.

The dangers, however, are not going away. Humphreys worries particularly that spoofing the GPS-sourced timing used to regulate financial databases could create havoc. Industries like financial services, he says, "have backups in place, but on close inspection one realizes that the backups themselves are either short-term or eventually trace their source to GPS."

"A coordinated attack that understood the finance world's dependency on GPS would be hard to detect and even harder to defeat," he cautions. C

Further Reading

Psiaki, M., and Humphreys, T.
Protecting GPS from Spoofers Is Critical to the Future of Navigation, ***IEEE Spectrum*****, Jul 29, 2016,** **http://spectrum.ieee.org/telecom/security/protecting-gps-from-spoofers-is-critical-to-the-future-of-navigation**

Amirtha, T.
Satnav spoofing attacks: Why these researchers think they have the answer, ***ZDNet*****, Mar 27, 2017,** **http://www.zdnet.com/article/satnav-spoofing-attacks-why-these-researchers-think-they-have-the-answer/**

U.S. Department of Homeland Security, National Cybersecurity & Communications Integration Center, National Coordinating Center for Communications Improving the Operation and Development of Global Positioning System (GPS) Equipment Used by Critical Infrastructure, http://bit.ly/2oZewfz

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

Milestones | DOI:10.1145/3122790 Lawrence M. Fisher

Turing Laureates Celebrate Award's 50th Anniversary

ACM recently held a conference in celebration of the first 50 years of the ACM A.M. Turing Award.

"Just over 50 years ago, ACM awarded its first A.M. Turing Award to Alan Perlis for his work on advanced programming techniques and compiler construction," said ACM president Vicki L. Hanson. "In total, 64 people from around the world have received the Turing Award, recognizing work that laid the foundations of modern computing."

The award was presented to its 65th recipient, Sir Tim Berners-Lee, at the event in June.

The conference included more than 20 Turing Laureates speaking on topics related to their fields of study.

After welcomes from Hanson, program chair Craig Partridge, and master of ceremonies (and past ACM president) Dame Wendy Hall, 2008 Turing Laureate Barbara Liskov (who received the award "for contributions to practical and theoretical foundations of programming language and system design, especially related to data abstraction, fault tolerance, and distributed computing") offered a presentation on the "Impact of Turing Recipients' Work" focusing on the impact of early Turing recipients, which she described as "tremendous."

A session on "Advances in Deep Neural Networks" featured 2011 Turing Laureate Judea Pearl ("for fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning"), who spoke about an evolutionary advance 40,000 years ago that allowed *Homo sapiens* to advance past competitor species *Homo erectus* and the Neanderthals. "The ability to imagine things that do not physically exist ... the ability to model one's environment, imagine other worlds, served to accelerate evolution in favor of *Homo sapiens*," he said.

The session on "Restoring Personal Privacy Without Compromising National Security" featured 2015 Turing Laureate Whitfield Diffie (co-recipient of the award with Martin Hellman "for inventing and promulgating both

Among the 22 Turing Laureates in attendance at the conference were: Front row, from left: Whitfield Diffie (2015), Martin Hellman (2015), Robert Tarjan (1986), Barbara Liskov (2008). Second row, from left: Vinton Cerf (2004), Richard Karp (1985), Richard Stearns (1993), Dana Scott (1976). Third row, from left: Ivan Sutherland (1988), Leslie Valiant (2010), Robert Kahn (2004). Fourth row, from left: Frederick Brooks (1999), Raj Reddy (1994), William (Velvel) Kahan (1989), Donald Knuth (1974).

PHOTOGRAPHS BY MISTI LAYNE

asymmetric public-key cryptography, including its application to digital signatures, and a practical cryptographic key-exchange method"), who observed that calls by government agencies to incorporate "backdoors" in computing systems that would allow them to bypass normal authentication or encryption are not really necessary. "New backdoors aren't required; the security failures of most programs give the government ample opportunity to 'break in.'"

In a discussion about "Preserving Our Past For The Future," 2004 Turing Laureate (and ACM past president) Vint Cerf ("with Robert E. Kahn, for pioneering work on internetworking, including the design and implementation of the Internet's basic communications protocols, TCP/IP, and for inspired leadership in networking") related an anecdote about coming across an old 3.5-inch floppy disk and tracking down a compatible disk drive, but still being unable to open the files on the disk because they were saved in an outdated version of WordPerfect. "Backward compatibility suffers because you can't keep everything," like the version of WordPerfect needed to open those files, he said.

In a session on the future of microelectronics entitled "Moore's Law Is Really Dead: What's Next?" moderator John Hennessy of Stanford University said, "We're reaching the end of silicon technology as we know it. " As a result, said Doug Burger of Microsoft Research, "We're entering a wild, messy, destructive time. It sounds like a lot of fun." Margaret Martonosi of Princeton University said, "We're entering a post-ISA, Post-CPU era ... we need to be exploring design processes to be domain-specific, and we need to train students that way as well."

Butler Lampson, the 1992 Turing Laureate ("for contributions to the development of distributed, personal computing environments and the technology for their implementation: workstations, networks, operating systems, programming systems, displays, security, and document publishing"), said, "There's plenty of room at the top; there's room in software, algorithms, and hardware." He added, "We know there's a lot of software bloat, that we can get rid of, at a cost." Also, he said, "A consequence of hardware changes not going to be invisible anymore is, you need a strategy for changes in the software stack."

Laureates, from left, Vinton Cerf, Edward Feigenbaum, and Raj Reddy.

A panel on Moore's Law was moderated by John Hennessy (left) and included Doug Burger, Norman Jouppi, Butler Lampson (1992), and Margaret Martonosi.

With regard to hardware advances, Lampson said, "What people care about is that the cost of running their application drops. "

Norman P. Jouppi, Distinguished Hardware Engineer at Google, concluded Moore's Law is "not dead, it's just resting."

Regarding "Challenges in Ethics and Computing," 1994 Turing Laureate Raj Reddy (co-recipient with Ed Feigenbaum "for pioneering the design and construction of large-scale artificial intelligence systems, demonstrating the practical importance and potential commercial impact of artificial intelligence technology") said, "We need to identify technological solutions to societal problems. I believe we can." One of those solutions, he said, might be "designing self-healing systems in every system we design."

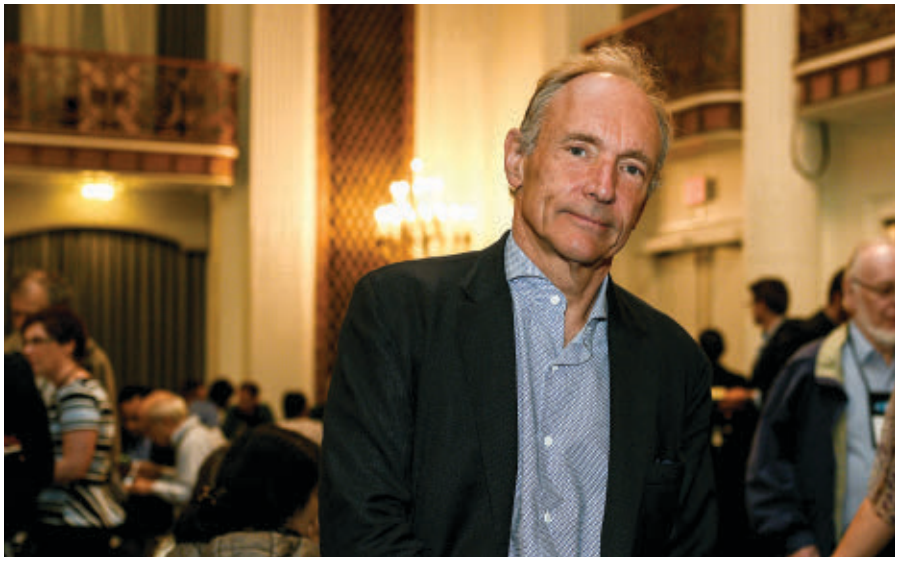
In the future, Reddy said, there will be no separation between humans and technology. "Humans will have technology in their bodies and be able to do things no person or computer could do alone. That system should have ethics.

Kenneth Thompson (1983).

Leonard Adleman (2002).

Andrew Chi-Chih Yao (2000).

Judea Pearl (2011) moderated a panel on deep neural networks.



The newest Turing Laureate—Sir Tim Berners-Lee.

Unfortunately, that's trumped by laws and government."

"Accountability is what we want from all systems," Reddy said. "The role of philosophers/ethicists is to convince the government," because "if it is not written into the law, nothing will change. Unless we find mechanisms to get it into the legal system, we can have all kinds of discussions and nothing will happen."

Opening the second day of the conference, 1974 Turing Laureate Donald Knuth ("for his major contributions to the analysis of algorithms and the design of programming languages, and in particular for his contributions to the 'art of computer programming' through his well-known books in a continuous series by this title") addressed "Computer Science as a Major Body of Accumulated Knowledge." Computer science, he said, shares with mathematics "the great privilege that we can invent problems to work on." Basically, he said, computer science and mathematics are "are two parallel disciplines with a lot in common, but a distinct difference."

Knuth said he was both " optimistic and pessimistic" about artificial intelligence, and that he is "more pessimistic when it is based on the notion humans make rational decisions."

The 79-year-old Knuth said he considers "computer programming is art, in the sense that it's not from nature, as well as being beautiful."

As a member of the panel discussing "Quantum Computing: Far Away? Around the Corner? Or Maybe Both at the Same Time?" 2000 Turing Laureate Andrew Chi-Chih Yao ("in recognition of his fundamental contributions to the theory of computation, including the complexity-based theory of pseudo-random number generation, cryptography, and communication complexity") said, " I am a believer in quantum computing," adding, "it seems clear that the technology of quantum computing is going to have a big practical impact."

Yao described quantum computing as "a great experiment, and we're all waiting to see what can come of it." He also called is "a great paradigm for interdisciplinary computing."

The session on "Augmented Reality: From Gaming to Cognitive Aids and Beyond" was the only session to feature two Turing Laureates: 1988's Ivan Sutherland ("for his pioneering and visionary contributions to computer graphics, starting with Sketchpad, and continuing after"), and 1999's Frederick P. Brooks, Jr. ("for landmark contributions to computer architecture, operating systems, and software engineering").

Brooks said he has a vision of using augmented reality (AR) for the purpose of training emergency teams. He asked the panel about "the state of actual use of augmented reality today? Who is using is a tool to earn their living?" Sutherland responded that the pilot of a jumbo jet, who trains in a simulator, is taking advantage of "some of the best VR (virtual reality) in use today," while Yvonne Rogers of University College London pointed out that head-up displays "are a reality for navigation." Peter Lee, of Microsoft AI and Research, said there is "a lot of belief, interest, and a growing amount of experimentation in AR, such as the ability to "teleport" (virtual visit other locations); he added, "If we can teleport, there really isn't a need for so many airplanes."

Sutherland added that the "greatest value of AR/VR is to show people things in a way that makes the underlying physics, the meaning, clear."

The full conference sessions are available at https://www.facebook.com/pg/AssociationForComputingMachinery/videos/.

—Lawrence M. Fisher

A young conference attendee takes a selfie with Ivan Sutherland (1988).

Panel discussions during the conference drew a packed house.

In Memoriam | DOI:10.1145/3125605 Lawrence M. Fisher

Charles W. Bachman: 1924–2017

An engineer best known for his work in database management systems, and in techniques of layered architecture that include Bachman diagrams.

CHARLES WILLIAM "CHARLIE" Bachman, the "father of databases" who received the ACM A.M. Turing Award for 1973 for creating the first database management system, died June 13 at the age of 92.

Born in Manhattan, KS, in 1924, Bachman earned his B.S. in mechanical engineering in 1948, as well as an M.S. in mechanical engineering from the University of Pennsylvania.

He went to work for Dow Chemical in 1950, using mechanical punched-card computing devices to solve networks of simultaneous equations representing data from Dow plants. In 1957, Bachman became head of Dow's Data Processing Department, through which he became a member of Share Inc., and a founding member of the Share Data Processing Committee.

In 1960, Bachman joined the General Electric (GE) Production Control Services Group in New York City, using a factory in Philadelphia to test designs for a system to automate factory planning, scheduling, operational control, and inventory control. The resulting MIACS was based on the Integrated Data Store (IDS), Bachman's concept of an "information inventory," and was first to adopt the "network data model" in which the system would support and enforce relationships between records.

Bachman moved to GE's Computer Department in 1964, where he helped build another management information system, the Weyerhauser Comprehensive Operating Supervisor (WEYCOS 2).

Bachman was awarded the ACM A.M. Turing Award for 1973 for his contributions to database technology. As biographer Thomas Haigh observed, "Bachman was the first Turing Award winner without a Ph.D., the first to be trained in engineering rather than science, the first to win for the application of computers to business administration, the first to win for a specific piece of software, and the first who would spend his whole career in industry."

Who inspired Bachman? "The inventors, the developers of new concepts, the solvers of previously unsolved problems."

The British Computer Society named Bachman a Distinguished Fellow in 1977 for his work in database systems.

Bachman received the U.S. National Medal of Technology and Innovation (NMTI) for 2012. The award was presented to Bachman in 2014 by President Barack Obama.

He was nominated for the NMTI by U.S. Senator Edward J. Markey (D-MA), who said, "The United States would not be the worldwide hub for technological innovation had it not been for the achievements of Charles Bachman."

Data scientist Gary Rector said Bachman was "humble, kind, generous, and a gentle soul; his entire family reflects that humanity. Charlie loved flowers and had a smile that embraced everyone. His heart connected to people more meaningfully than any database could ever do merely with data. To connect to people in this way is the greatest lesson he gave me."

George Colliat, a colleague from GE, said, "I have learned from his ability to look for solutions that transcend the problems at hand and thereby multiply the value of the solutions." He added, "Charlie's human values have influenced me as much as his creative genius. His respect for his colleagues, always looking for their positive contribution, his patience in explaining ideas to people who were not always at his level, his humility and open mind in always listening to others as an opportunity to learn something new, characterize him as a gentleman in this industry."

Haigh last saw Bachman when he was "close to 90 but still sharp and enjoying life; talking about the article he was working on and his chats with E.O. Wilson in the retirement community they shared. He never stopped trying to understand how things worked, or trying to make them work better. I feel honored to have known him."

In 2014, Bachman was named a Fellow of the ACM for his contributions to database technology.

Bachman was named a Fellow of the Computer History Museum in 2015, for his work on database management systems. Also that year, Michigan State University awarded Bachman an honorary doctorate of engineering for being "at the forefront of computer science for more than 65 years."

Bachman's son, Jon, said his father's vision of the Integrated Data Store resulted in "a high-performance direct access storage model (that) allows developers to build large efficient databases of any type of business or operational data. In fact, the first versions were so successful that they became established as the most important system software on mainframe computers of that era."

In an interview in 2008, Bachman was asked who in the IT industry "inspired you or was a role model for you?" He replied, "The inventors, the developers of new concepts, the solvers of previously unsolved problems, the assemblers of new and interesting combinations of old technologies. Take Sir Maurice Wilkes, Edsger Dijkstra, Sir Tim Berners-Lee."

PHOTO COURTESY OF BACHMAN FAMILY

DOI:10.1145/3126489

Joel R. Reidenberg

Law and Technology
Digitocracy

Considering law and governance in the digital age.

Digital technologies have unleashed profound forces changing and reshaping rule making in the democracies of the information society. Today, we are witnessing a transformative period for law and governance in the digital age. Elected representative government and democratically chosen rules vie for authority with new players who have emerged from the network environment. At the same time, network technologies have unraveled basic foundational prerequisites for the rule of law in democracy like privacy, freedom of association, and government oversight. The digital age, thus, calls for the emergence of a *Digitocracy*—a new set of more complex governance mechanisms assuring public accountability for online power held by state and nonstate actors through the creation of new checks and balances among a more diverse group of players than democracy's traditional grouping of a representative legislature, executive branch, and judiciary.

Where Google and Facebook know more than most spy agencies about the lives of millions of citizens as well as the inner workings of companies and governments, information powerhouses and platforms can establish their own rules for citizens' interactions online. Where public-sector surveillance and private-sector tracking are so pervasive, citizens lose the ability to control the disclosure of their thoughts, friends, activities, and no longer have privacy. Where lone coders wreak massive havoc for private gain or for opposition to governmental policies, they can use their information resources to reject majority rule. Where technology can protect the anonymity of wrongdoers, rule-breakers can escape accountability. In short, the modern information society destroys one of the most fundamental truths of any democracy that "the power to make the laws rests with those chosen by the people."[a]

a *King v. Burwell*, 135 S. Ct. 2480, 2496 (2015).

We are witnessing a transformative period for law and governance in the digital age.

The Internet's Promise

Without a doubt, the Internet revolutionized the dissemination of information and the ability of individuals to engage with each other. The euphoria surrounding the early days of the Internet's expansion into the public sphere predicted that technology would expand democracy and empower citizens around the world. The conventional wisdom thought citizen participation would multiply online with e-government, and the public would have better oversight of the state thanks to new capabilities for monitoring administrative and executive actions. The power of the Internet to disseminate information from one to millions and the power of the Internet to foster conversations seemed an unstoppable force for democratic discourse. Popular movements like the Arab Spring, the Occupy Movement, and the Bernie Sanders U.S. presidential campaign illustrated that information technologies could indeed significantly enhance and enable political organizing on a new, unprecedented scale. Many expected that mechanisms like open electronic proceedings for rule making and open data for government transparency would herald better representative government and decision making.

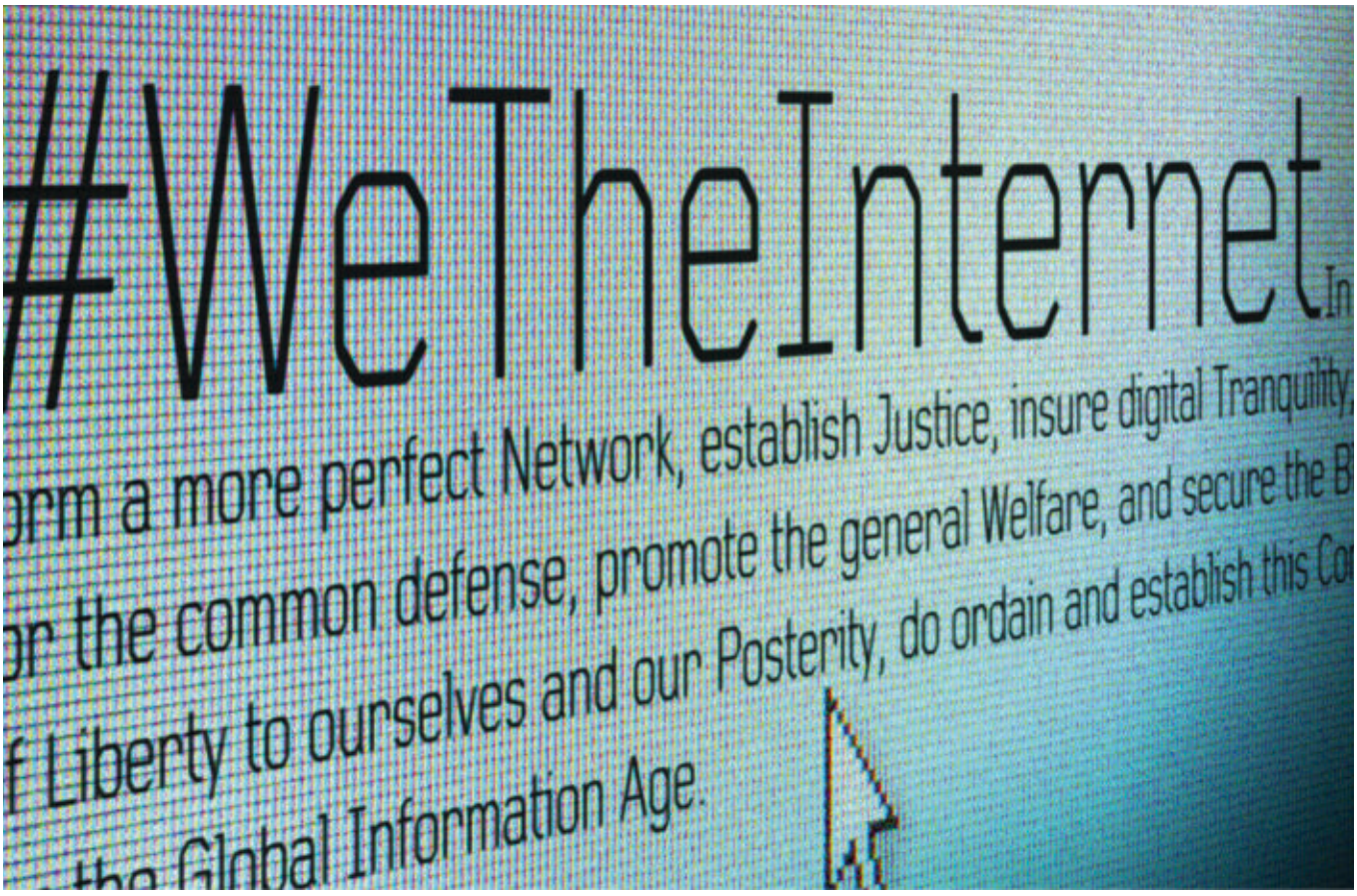


IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

The Internet's technical infrastructure turns out to challenge the promise of the political empowerment of citizens. Just as network technologies offered organizational tools for political empowerment, the technologies themselves provided the means to reverse the hope that the Internet would be a one-way pro-democracy force. Network infrastructure proved that it could be used to frustrate empowerment dreams. Egypt, for example, pulled the plug on the Internet for several days during the Arab Spring uprisings to block political organizing; Brazil shut down WhatsApp for 48 hours; local police in the U.S. used stealth Stingray technology to engage in large-scale geo-surveillance of citizens. And, at the same time, Twitter bots flooded social media in order to shut down political dialog or to falsify support for candidates, while hate and bullying flourish online. In short, the Internet has embedded the means to block political empowerment and discourse.

Undermining Democracy

In the intervening years since the early euphoria over the Internet's political potential, the embedding of the Internet in our daily lives has effectively demonstrated new vulnerabilities. The Internet's infrastructure has already displaced three key areas essential to the rule of law in democracy: sovereignty, government accountability, and respect for law. Internet technologies restructure a state's ability to prescribe and assure the enforcement of law. Governments forfeit sovereignty to networks when services like cloud computing transcend borders and enable organizations to choose rules in the blink of an eye. Network architecture enables technology developers and service providers to embed rules for online activities through infrastructure choices. For example, cloud service providers like Dropbox make determinations every day on the security of users' data. These encryption decisions determine the very capability of states to examine user data in lawful investigations.

Network infrastructure undermines the oversight and accountability of government. While open government technologies enable greater transparency of public institutions, electronic tools also empower governments to circumvent traditional political checks and balances and the public's oversight of government suffers irreparably. For example, in Oakland, CA, the police engaged in a mass-scale surveillance program to geo-locate thousands of mobile phones using stingray devices without any judicial approval and, in New York City, the police program to record drivers through traffic cams and smart city sensors also escapes judicial oversight. At the same time, technologically enabled leaks and wide dissemination of non-public activities of government through sites like WikiLeaks may jeopardize legitimate functions of government such as international relations and active law enforcement investigations. Snowden's leaks, for example, are reported to have endangered the lives of British M16 agents in Russia and China.

Laws lose their authority when governments can no longer control the use of power to enforce rules and hackers have control over weapons of mass disruption. Network infrastructure removes the state's monopoly on the use of coercive, police power to enforce rules and protect its citizens. Technol-

ogy allows lone-wolf actors unchecked by states to create and deploy weapons of mass disruption whether through malware, ransomware, or botnets. For example, hospitals across the U.S. in the spring of 2016 faced a wave of ransomware attacks that left some in a "state of emergency." ISIS uses crowd sourcing to sow terror in the U.S. and Europe. Simultaneously, the infrastructure empowers private actors to engage in vigilante actions. The underground group, Anonymous, recently illustrated such actions when they threatened an electronic attack against ISIS following the Paris massacres in November 2016. In essence, individuals and associations now have tools—outside the ability of state control—to enforce their choices and rules online in ways that are independent of the state. To be sure when a Texas college discovered in 2015 that Facebook provided better real-time information for an on-campus police emergency than 911, it becomes clear the state has even lost control over basic information it needs to protect its citizens.

Beyond undermining key aspects of the rule of law, the Internet's infrastructure has toppled critical, substantive legal pillars of democracy. Freedom of thought and association as well as public safety are essential elements of democracy and privacy is a prerequisite. Yet, the network infrastructure contradicts the basic tenents of freedom of association and privacy. Network functionality works thanks to ubiquitous data surveillance. The resulting transparency of citizens to those in the network undermine both state and citizen's respect for the rule of law. States lose important checks and balances against omnipotent acquisition of information and citizen's freedom of thought and association are undercut. Counterintuitively, public safety and security are also destabilized by the transparency when stalkers, social engineering hackers, and cyberwarriors find the informational keys to success readily accessible online.

Freedom of expression is another cornerstone of democracy. Yet, democracies have a capability problem dealing with socially destructive content like hate, threats, and cyberbullying that jeopardize public order and individual safety. Technology allows rapid and widespread dissemination of harmful content, while wrongdoers can shield their activities from accountability through encryption and anonymity tools. At the same time, freedom of expression limits the authority of states to ban nefarious online content. In the U.S., for example, there is no public recourse for the rapid growth of anti-Semitic Twitter accounts. Users must appeal to the social media firms who, in turn, then decide what to suppress or censor. By contrast, in Europe, platforms bear more legal responsibility for content, but firms are often left in the same position as an all-powerful censor. In effect, government is unable to suppress the vile and corrosive online material that threatens citizens without resorting to oppressive, anti-democratic controls.

Beyond undermining key aspects of the rule of law, the Internet infrastructure has toppled critical substantive legal pillars of democracy.

The Opportunity of Digitocracy

The information society lacks a model of governance suited to the digital age. Going forward, the digital age will need a new system of checks and balances for its political decision making—a "*Digitocracy*"—offering the opportunity to develop new governing principles that articulate *who* regulates *what* to preserve public accountability online.

Our challenge is how to construct the appropriate checks and balances. Digitocracy's dynamic will be much more complex than the analog world. Online private rule making like Twitter's decisions regarding censorship, Adobe's technical protections on digital content, and Facebook's settings for privacy have become more powerful in people's lives than rules from the democratic constitutional framework. Business organizations are likely to serve as counterweights to government power. Google's Transparency Report, Apple's defiance of an FBI request for encryption keys, and Microsoft's challenge to U.S. government access to foreign-based servers each reflect a check on the state's intrusiveness. And, individuals like Snowden may serve as counterweights to states and businesses. Individuals and associations of individuals have direct authority when they coalesce with online tools ranging from social media to hacktivism as they perceive the need to interject and amplify their end goals online. All while national government provides checks on overreaching private actors. Where each actor from a state to an individual can assure mass disruption online, fair governance will require co-existence among the rule-making actors.

At the core, the assurance of public accountability online is the key objective of Digitocracy. The mechanisms for states, private actors and citizens to co-exist as rule-makers in the networked society are likely to be defined in unexpected ways incorporating notions of federalism, multistakeholder governance, and subsidiarity. These tools will draw the boundaries of rule-making authority among the state actors, platform operators, corporate organizations, and empowered users. Each actor, whether state or non-state, has an important role to prevent overreaching by the other actors. In essence, Digitocracy constructs a more multifaceted set of interwoven checks and balances to establish limits on the powers of both state and non-state actors and a reliance on both to protect the public good. For our future, now is the time to begin the robust public discussion on our means of governance in the digital age. C

Joel R. Reidenberg (jreidenberg@law.fordham.edu) is the Stanley D. and Nikki Waxberg Chair and Professor of Law, Fordham University, Director, Fordham Center on Law and Information Policy, and Visiting Research Affiliate, Center for Information Technology Policy, Princeton University.

The author is preparing a book on this topic to be published by Yale University Press.

DOI:10.1145/3126492

Carolina Alves de Lima Salge and Nicholas Berente

Computing Ethics

Is That Social Bot Behaving Unethically?

A procedure for reflection and discourse on the behavior of bots in the context of law, deception, and societal norms.

ATTEMPTING TO ANSWER the question posed by the title of this column requires us to reflect on moral goods and moral evils—on laws, duties, and norms, on actions and their consequences. In this Viewpoint, we draw on information systems ethics[6,7] to present *Bot Ethics*, a procedure the general social media community can use to decide whether the actions of social bots are unethical. We conclude with a consideration of culpability.

Social bots are computer algorithms in online social networks.[8] They can share messages, upload pictures, and connect with many users on social media. Social bots are more common than people often think.[a] Twitter has approximately 23 million of them, accounting for 8.5% of total users; and Facebook has an estimated 140 million social bots, which are between 5.5%–1.2% total users.[b,c] Almost 27 million Instagram users (8.2%) are estimated to be social bots.[d] LinkedIn and Tumblr also have significant social bot activity.[e,f] Sometimes their activity on these networks can be innocuous or even beneficial. For example, *SF QuakeBot*[g] performs a useful service by disseminating information about earthquakes, as they happen, in the San Francisco Bay area. However, in other situations, social bots can behave quite unethically.

Items purchased by Random Darknet Shopper, an automated computer program designed as an online shopping system that would make random purchases on the deep Web. The robot would have its purchases delivered to a group of artists who then put the items in an exhibition in Switzerland; the robot was 'arrested' by Swiss police after it bought illegal drugs.

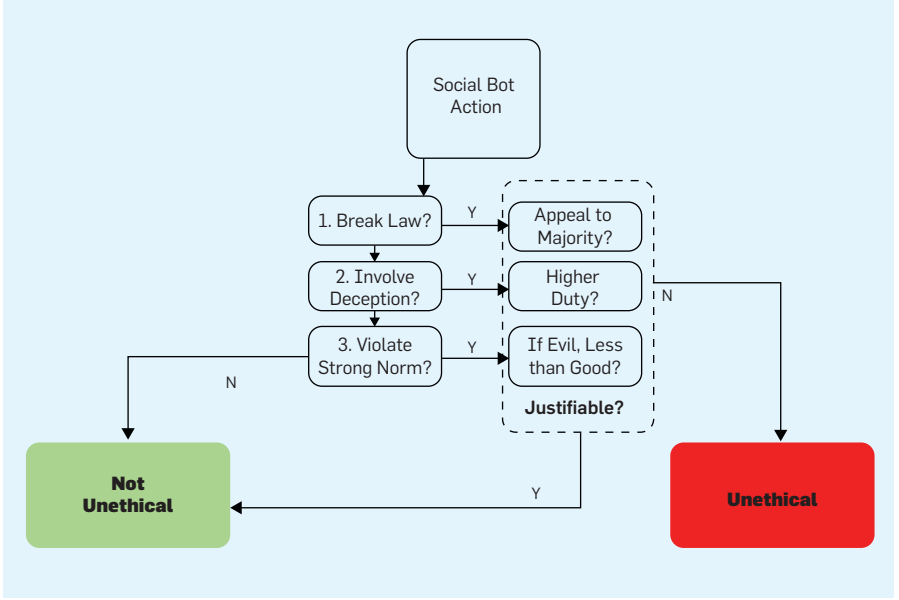
IMAGE COURTESY OF MEDIENGRUPPE BITNIK

Social Bots Behaving Unethically

LinkedIn reports that social bots on the professional networking platform are often used to "steal data about legitimate users, breaching the user agreement and violating copyright law."[h] Social bots have been reported to behave badly in a variety of ways across various contexts—everything from disseminating spam[i] and fake news[j] to limiting free speech.[k] But it is not always clear whether their undesirable activity is simply a nuisance or whether it is indeed unethical—particularly given the random nature of the logic underlying many social bots. Bad actions are not necessarily unethical—

a http://bit.ly/2uDfIbP
b http://cnnmon.ie/2uFR4XJ
c http://bit.ly/1ieIIXN
d http://read.bi/1LFQJFU
e http://bit.ly/1Ktz5kc
f http://tcrn.ch/2tKo90x
g http://bit.ly/2vneleU
h http://bit.ly/2vFRI4E
i http://ubm.io/1MbsSf3
j http://bit.ly/2ftn0It
k http://bit.ly/14bDiuN

Bot Ethics: How to determine whether social bot actions are unethical.



there are shades of gray that are difficult to judge.

For example, *Tay*,[l] a social bot created by Microsoft to conduct research on conversational understanding, went from "humans are super cool" to "Hitler was right I hate the Jews" in less than 24 hours on Twitter due to malicious humans interacting with the social bot.[m] In another case, a social bot tweeted "I seriously want to kill people" from randomly generated sentences during a fashion convention in Amsterdam.[n] Clearly such inadvertent comments violate our sensibilities and are distasteful, but are they unethical? Perhaps, but by what standard do we judge? Some social bots do more than just comment—clearly those that steal information and other misdeeds are engaging in unethical activity, but, again, it is not always so clear. For instance, the *Random Darknet Shopper*—a social bot coded to explore the dark Web in the name of art—inadvertently purchased 10 Ecstasy pills (an illegal narcotic) and a counterfeit passport.[o] So a law was broken, but was this unethical behavior? We developed a procedure, which we describe next, to help answer such questions.

l https://twitter.com/TayandYou
m http://bit.ly/14bDiuN
n http://bit.ly/2ttN5Ox
o http://bit.ly/2vFGdu9

Bot Ethics: A Procedure to Evaluate the Ethics of Social Bot Activity

Ethics in philosophy dates back thousands of years, and this Viewpoint column cannot do justice to the entire field. However, because of the increasing prominence of social bots and their potential for malicious activity, ethical judgment about their activity is necessary. The best way to guide ethical conduct in a community is to provide a procedure for reflection and discourse.[5] The procedure we created is called "*Bot Ethics*" (see the figure here) and it focuses on the behavior of social bots with respect to law, deception, and norms.

Break Law?

Many laws are developed from ethical principles.[6] Even when a law may be flawed, it is typically the ethical course of action to follow that law.[9] Therefore a natural first question is: "*Does the action of the social bot break the law?*" The objective is to assess straightforward ethical questions, such as whether algorithms plant viruses in someone else's device. This is clearly illegal and unethical. There are cases where a social bot might ethically violate the law, such as civil disobedience for a cause the creator considers just. However, civil disobedience is only ethical in very rare cases in constitutional democracies where legal recourse for unjust laws pervade.[6] Cases where a law may be broken that are not unethical require justification—compelling arguments that appeal to moral standards of the majority.[6] Only in such rare cases may illegal acts be seen as moral and therefore ethical.[6] Thus we ask "*Is the illegal act justifiable?*" Acts that are not suitably justifiable (that is, do not appeal to the morality of the majority) are unethical. Swiss authorities did not file charges against the *Random Darknet Shopper* developers.[p] They argued that social bots can buy illegal narcotics over the Internet for the purpose of art[q] and that "ecstasy in this presentation was safe." The behavior was not unethical because it was justified according to the pervading morality of the community.

Social bots have been reported to behave badly in a variety of ways across various contexts.

Involve Deception?

If a social bot's behavior does not break any laws, next evaluate for truthfulness: "*Is any deception involved?*" Social bots may act deceitfully. For example, they can misrepresent themselves as human beings[2] or spread untruthful information (such as fake news). Deceiving acts communicate false or erroneous assertions, violating the prima facie duty of fidelity. Social bots should always act truthfully.[3] However, deceitful acts can be justifiable if the duty of fidelity is superseded by a higher-order duty, such as beneficence.[r] Deceptive, satirical actions may not be unethical since they elicit pleasure, improving the life of others. Consider *Big Data Batman*[s] as an illustration.

p By "developer" we are referring to either the organization or management of the organization or the software developer involved in the creation of the social bot.
q http://bit.ly/2ud2cZC
r Beneficence is the duty to bring virtue, knowledge or pleasure to others; other duties, according to Ross 1930, include non-maleficence, self-improvement, justice, gratitude, reparation (see Mason et al.[7], p. 132–133).
s http://bit.ly/2ttNUH7

The social bot finds every tweet with the term big data, replaces "big data" with "Batman," and then tweets the message as if it were its own. It obviously substitutes its words for others' words, but the satire makes it difficult to judge its ethics. Because the social bot might insult and embarrass some big-data advocates the community must go beyond the act (deontology) to consider its consequences (teleology), and ask whether potentially bad actions (for example, insult and embarrassment) outweigh, or supersede, the good (for example, pleasure through laughter) for the involved parties. Again, is the deception justifiable? Deception in the absence of supersession is likely to be unethical.

Violate Strong Norm?

Social bots that are legal and truthful can still behave unethically by violating strong norms that create more evil than good. Moral evils inflict "limits on human beings and contracts human life."[4] Evil restrains, instead of emancipating, evil actions reduce opportunities. Let us go back to *Tay*'s racist comments on Twitter. Although not illegal (First Amendment protections apply), nor deceitful, they violated the strong norm of racial equality. Social media companies like Twitter that temporarily lock or permanently suspend accounts that "directly attack or threaten other people on the basis of race,"[t] have established that the moral evil of racism outweighs the moral good of free speech. By applying *Bot Ethics* to Twitter's norms we conclude that *Tay*'s actions were unethical. Yet, there are cases where social bots may violate strong norms and not act unethically, as with asking inappropriate questions (what is your salary?). Such violations do not create moral evils.

Culpability of Unethical Social Bot Behavior

Should the general social media community blame developers for unethical behavior of their social bots? In the example of the algorithm that randomly generated that it wanted to kill people, who is responsible for the death threat? The programmer? Who is responsible for *Tay*'s remark about

Should the general social media community blame developers for the unethical behavior of their social bots?

Hitler—Microsoft developers or those teaching the social bot to generate racist statements? Similarly, who is responsible for the social bot buying the illegal narcotics?

Aristotle[1] said we can only assign culpability if we know that individuals behaved voluntarily and knowingly. Involuntary situations likely do not apply to social bots. Developers who are coerced into doing something unethical without a choice may not be entirely culpable, but in the case of free enterprise there is always a choice. Therefore, culpability rests on the knowledge of the developers. Developers who knowingly create social bots to engage in unethical actions are clearly culpable. They should be punished if evidence of their wrongdoing is convincing—the penalty must be consistent and proportional to the harm done and those affected should be compensated.[7]

But what about situations where developers act unknowingly? In those occasions the community must determine whether developers are culpably ignorant—did they ignore industry best practices in creating and testing their algorithms? If industry guidelines were not followed and the action was unethical, developers are culpable. However, developers who followed good development practices and incorporated the current industry thinking, and yet their social bot still acted unethically, deserve our pity and pardon, but they are not culpable. They should apologize, correct immediately, learn from their experience, and communicate the occurrence to the development community. For example, Microsoft posted its learning from *Tay* in blog form.[u]

Conclusion

We do not purport to write the last word on social bot ethics and culpability. Ethics is simply too complex of a domain to deal with fully in such a format. Nevertheless, some readily accessible guidance rooted in sound ethical thinking is in order.

For example, with the recent attention to the role of social bots in spreading misinformation in the form of "fake news," other social bots, such as *Reuters News Tracer*, are being created to ferret out such deceitful activity.[v] The *Bot Ethics* procedure can help the social media community understand when these deceitful actions are indeed unethical. It further helps to expand the focus of the community beyond narrow (that is, only deceitfulness) and simplistic (that is, good or bad bot) assessments of social bot activity to attend to the complexities of ethical assessments. In short, the *Bot Ethics* procedure serves as a starting point and guide for ethics-related discussion among various participants in a social media community, as they evaluate the actions of social bots. C

t http://bit.ly/19SJwlt

u http://bit.ly/2tiPfMH

v http://bit.ly/2hIlfXG

References

1. Aristotle. *Nicomachean Ethics of Aristotle*. E.P. Dutton, NY, 1911.
2. Ferrara, E. et al. The rise of social bots. *Commun. ACM 59*, 7 (July 2016); 96–104; DOI: 10.1145/2818717
3. Gotterbarn, D., Miller, K. and Rogerson, S. Computer society and ACM approve software engineering code of ethics. *Computer Society Connection*, (1999), 84–88.
4. Grisez, G. and Shawn, R. *Beyond the New Morality: The Responsibilities of Freedom*. University of Notre Dame Press, Notre Dame, IN, 1980.
5. Habermas, J. *The Theory of Communicative Action, Volume 1: Reason and the Rationalization of Society*. 1985.
6. Kallman, E.A. and Grillo, J.P. *Ethical Decision Making and Information Technology*. McGraw-Hill, New York, NY, 1996.
7. Mason, R.O., Mason, F.M., and Culnan, M. *Ethics of Information Management*. Sage Publications, London, U.K.
8. Morstatter, F. et al. A new approach to bot detection: Striking the balance between precision and recall. ASONAM, 2016.
9. Rawls, J. The justification of civil disobedience. *Arguing about Law* (2013). 244–253.

Carolina Alves de Lima Salge (csalge@uga.edu) is a doctoral candidate at the University of Georgia.

Nicholas Berente (berente@uga.edu) is an associate professor at the University of Georgia.

DOI:10.1145/3126494

Peter J. Denning

The Profession of IT
Multitasking Without Thrashing

Lessons from operating systems teach how to do multitasking without thrashing.

Our individual ability to be productive has been hard stressed by the sheer load of task requests we receive via the Internet. In 2001, David Allen published *Getting Things Done*,[1] a best-selling book about a system for managing all our tasks to eliminate stress and increase productivity. Allen claims that a considerable amount of stress comes our way when we have too many incomplete tasks. He views tasks as loops connecting someone making a request and you as the performer who must deliver the requested results. Getting systematic about completing loops dramatically reduces stress.

Allen says that operating systems are designed to get tasks done efficiently on computers. Why not export key ideas about task management into a personal operating system? He calls his operating system GTD, for Getting Things Done. The GTD system supports you in tracking open loops and moving them toward completion. It routes incoming requests to one of these destinations in your filing system:

- Trash
- Tasks that might one day turn out to be worth doing
- Tasks that serve as potential future reference points
- Tasks delegated to someone else, awaiting their response
- Tasks that can be completed immediately in under two minutes
- Tasks accepted for processing

The first four destinations basically remove incoming tasks from your workspace, the fifth closes quick loops, and the sixth holds your incomplete loops. GTD helps you keep track of these unfinished loops.

The idea of tasks being closed loops of a conversation between a requester and a perform was first proposed in 1979 by Fernando Flores.[5] The "conditions of satisfaction" that are produced by the performer define loop completion and allow tracking the movement of the conversation toward completion. Incomplete loops have many negative consequences including accumulations of dissatisfaction, stress, and distrust.

Many people have found the GTD operating system to be very helpful at completing their loops, maintaining satisfaction with work, and reducing stress. It is a fine example of us taking lessons from technology to improve our lives.

Multitasking

Unfortunately, GTD does not eliminate another source of stress that was much less of a problem in 2001 than today. This is the problem of thrashing when you have too many tasks in progress at the same time.[2]

The term multitasking is used in operating systems to mean executing multiple computational processes simultaneously. The very first operating system do this was the Atlas supervisor, running at the University of Manchester, U.K., in 1959. IBM brought the idea to the commercial world with its OS 360 in 1965.

Operating systems implement multitasking by cycling a CPU through a list of all incomplete tasks, giving each one a time slice on the CPU. If the task does not complete by the end of its time slice, the OS interrupts it and puts it on the end of the list. To switch the CPU context, the OS saves all the CPU registers of the current task and loads the registers of the new task. The designers set the time slice length long enough to keep the total context switch time insignificant. However, if the time slice is too short, the system can significantly slow down due to rapidly accumulating context-switching time.

When main memory was small, multitasking was implemented by loading only one task at a time. Thus, each context switch forced a memory swap: the pages of the running task were saved to disk, and then the pages of the new task loaded. Page swapping is extremely expensive. The 1965 era OSs eliminated this problem by combining multitasking with multiprogramming: the pages of all active tasks stay loaded in main memory and context switching involves no swapping. However, if too many tasks were activated, their allocations would be too small and they would page excessively, causing system throughput to collapse. Engineers called this thrashing, a shorthand for "paging to death."

Eventually researchers discovered the root cause of thrashing and built control systems to eliminate it—I will return to this shortly.

Figure 1. In this memory map of a Firefox Browser in Linux, the colored pixels indicate that a page (vertical axis) is used during a fixed size execution interval (horizontal axis). The locality sets (pages used) are small compared to the whole address space and their use persists over extended intervals.

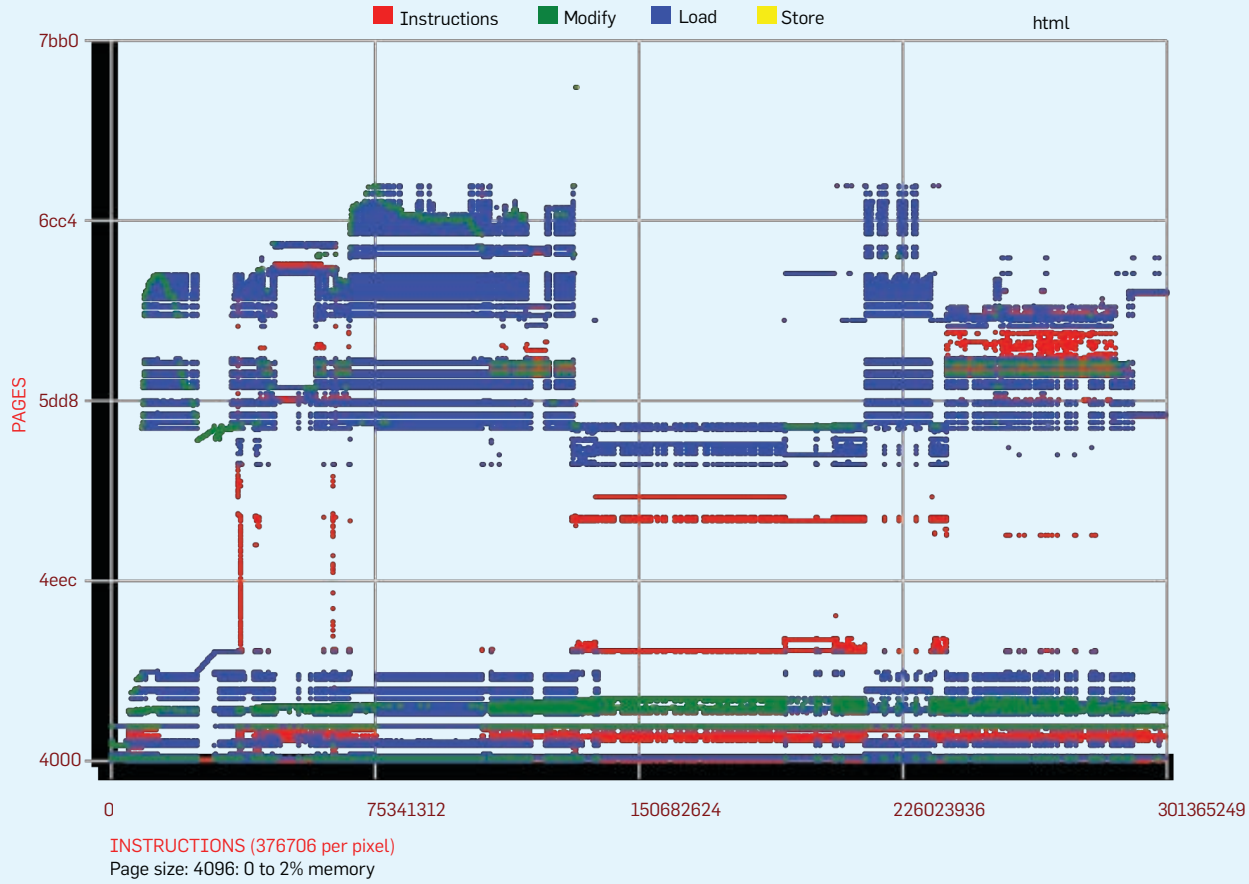


FIGURE 1 COURTESY OF ANDRIAN MCMENAMIN

Human Multitasking

Humans multitask too by juggling several incomplete tasks at once. Cognitive scientists and psychologists have studied human multitasking for almost two decades. Their main finding is that humans do not switch tasks well. Psychologist Nancy Napier illustrates with a simple do-it-yourself test.[7] Write "I am a great multitasker" on line 1 and the series of numbers 1, 2, 3, ..., 20 on line 2. Time how long it takes to do this. Now do it again, alternating one letter from line 1 and one numeral from line 2. Time how long it takes. For most people, the fine-grained multitasking in the second run takes over twice as long as the one-task-at-a-time first run. Moreover, you are likely to make more errors while multitasking. This test reveals just how slow our brains are at context switching. You can try the test a third time using time-slicing, for example writing five letters and then switching to write five numerals. With fewer context switches, time-slicing is faster than fine-grained multitasking but still slower than one-at-a-time processing.

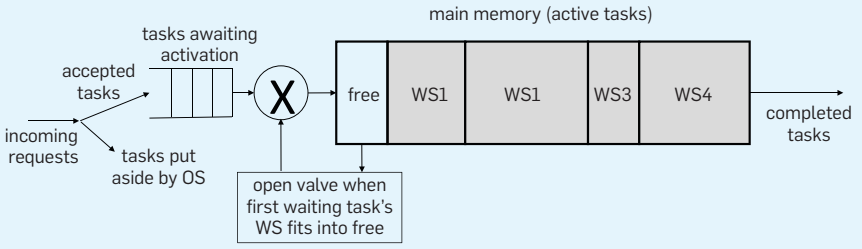
Human context switching is more complicated than computer context switching. Whereas the computer context switch replaces a fixed number of bytes in a few CPU registers, the human has to recall what was "on the mind" at the time of the switch and, if the human was interrupted with no opportunity to choose a "clean break," the human has to reconstruct lost short term memory.

Context switching is not the only problem. Whereas a computer picks the next task from the head of a queue, your brain has to consider all the tasks and select one, such as the most urgent or the most important. The time to choose a next task goes up faster than linear with the number of tasks. Moreover, if you have several urgent important tasks, your brain can get stuck in a decision process that can take quite a long time to decide—a situation known as the choice uncertainty problem.[4]

A third factor that slows human multitasking is gathering the resources necessary to continue with a task. Some resources are physical such as books, equipment, and tools. Some are digital such as files, images, sounds, Web pages, and remote databases. And some are mental, things you have to remember about where you were in the task and what approach you were taking to perform it. All these resources must be close at hand so that you can access them quickly.

These three problems plague multitaskers of all age groups. Many studies report considerable evidence of negative effects—multitasking seems to reduce productivity, increase errors, increase stress, and exhaust us. Some researchers report that multitaskers are less likely to develop expertise in a topic because they do not get enough inten-

Figure 2. OS control system to maximize throughput with variable partition of main memory determined by task working sets.



sive focused practice with it. Some fret that if we do not learn to manage our multitasking well, we may wind up becoming a world of dilettantes with few experts to keep our technology running.

Thrashing happens to human multitaskers when they have too many incomplete tasks. They fall into a mood of "overwhelm" in which they experience considerable stress, cannot choose a next task to work on, and cannot stay focused on the chosen task. It can be a difficult state to recover from.

Let us now take a look at what OSs do to avoid thrashing and see what lessons we can take to avoid it ourselves.

Locality, Working Sets, and Thrashing

The OS seeks to allocate memory among multiple tasks so as to maximize system throughput—the number of completed tasks per second.[3]

The accompanying Figure 1 is strong graphical evidence of the principle of locality—computations concentrate their memory accesses to relatively small locality sets over extended intervals. Locality should be no surprise—it reflects the way human designers approach tasks.

We use the term working set for OS's estimate of a task's locality set. The formal definition is that working set is the pages used in a backward-looking window of a fixed size T memory references. In Figure 1, T is the length of the sampling interval and the working set equals the locality set 97% of the time.

Each task needs a workspace—its own area of memory in which to load its pages. There are at least two ways to divide the total memory among the active tasks. In fixed partitioning, the OS gives each task a fixed workspace. In working-set partitioning, the OS gives each task a variable workspace that tracks its locality sets. Fixed partitioning is susceptible to thrashing as the number of tasks sharing memory increases because each gets a smaller workspace and, when the workspaces are smaller than the working sets, every task is quickly interrupted by a page fault.

Under working-set partitioning the OS sizes the workspaces to hold each task's measured working set. As shown in Figure 2, it loads tasks into memory until the unused free space is too small to hold the next task's working set; the remaining tasks are held aside in a queue until there is room for their working sets. When a task has a page fault, the new page is added to its workspace by taking a free page; when any page has not been used for T memory references, it is evicted from the task's workspace and placed in the free space. Thus, the OS divides the memory among the active tasks such that each task's workspace tracks its locality sets. Page faults do not steal pages from other working sets. This strategy automatically adjusts the load (number of active tasks) to keep throughput near its maximum and to avoid thrashing.

Context switching is *not* the cause of thrashing. The cause of thrashing is the failure to give every active task enough space for its working set, thereby causing excessive movement of pages between secondary and main memory.

Translation to Human Multitasking

Although the analogy with OSs is not perfect, there are some lessons:

- Recognize that each task needs a variable working set of resources (physical, digital, and mental), which must be easily accessible in your workspace. Analog: the working set of pages.
- Your capacity to deal with a task is the resources and time needed to get it done. Analog: the memory and CPU time needed for a task.
- Some tasks need to be held aside in an inactive status until you have the capacity to deal with them. Analog: the waiting tasks queue.
- When a task's working set is in your workspace, protect it from being unloaded as long as the task is active. Analog: protect working sets of active tasks and do not steal from other tasks.
- You will thrash if you activate too many tasks so that the total demand is beyond your capacity. Analog: insufficient CPU and memory for active tasks.
- If you are able to choose moments of context switch, select a moment of "clean break" that requires little mental reacquisition time when you return to the task. If you cannot defer an interruption to such a moment, you will need more reacquisition time because you will have to reconstruct short-term memory lost at the interruption. Analog: ill-timed interrupts can cause loss of part of a working set.

You are likely to find that you cannot accommodate more than a few active tasks at once without thrashing. However, with the precautions described here, thrashing is unlikely. If it does occur you will feel overwhelmed and your processing efficiency will be badly impaired. To exit the thrashing state, you need to reduce demand or increase your capacity. You can do this by reaching out to other people—making requests for help, renegotiating deadlines, acquiring more resources, and in some cases canceling less important tasks. C

References

1. Allen, D. *Getting Things Done*. Penguin. 2001.
2. Christian, B. and Griffiths. T. *Algorithms to Live By: The Computer Science of Human Decisions*. Henry Holt and Company, 2016.
3. Denning, P. Working sets past and present. *IEEE Trans Software Engineering SE-6*, 1 (Jan. 1980), 64–84.
4. Denning, P. and Martell, C. *Great Principles of Computing*. MIT Press, 2015.
5. Flores, F. *Conversations for Action and Collected Essays*. CreateSpace Independent Publishing Platform, 2012.
6. McMenamin, A. Applying working set heuristics to the Linux kernel. Masters Thesis, Birkbeck College, University of London, 2011; http://bit.ly/2vFSgY8
7. Napier, N. The myth of multitasking, 2014; http://bit.ly/1vuBGcC

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

DOI:10.1145/3126156

Gregorio Convertino and Nancy Frishberg

Viewpoint
Why Agile Teams Fail Without UX Research

Failures to involve end users or to collect comprehensive data representing user needs are described and solutions to avoid such failures are proposed.

LESSONS LEARNED BY two user researchers in the software industry point to recurrent failures to incorporate user experience (UX) research or design research. This leads agile teams to miss the mark with their products because they neglect or mischaracterize the target users' needs and environment. While the reported examples focus on software, the lessons apply equally well to the development of services or tangible products.

Even when customers are involved, sometimes the teams may still fail to involve the actual end users.

Why It Matters to the ACM Community

Over the past 15 years, agile and lean product development practices have increasingly become the norm in the IT industry.[3] At the same time, two synergistic trends have also emerged.

▸ End users' demand for good user experience has increased significantly, with wide adoption of mobile devices. Any new application needs to do something useful or fun, plus it needs to do it well and fast enough. In 2013, technology analysts found that only 16% of people tried a new mobile app more than twice, suggesting that users have low tolerance for poor user experience (UX) (where UX is the totality of user's interactions supported by the app to accomplish a goal).[9]

▸ With growing emphasis on good UX design, UX professionals, both designers and researchers, are gradually being incorporated as required roles in software development, alongside product managers and software developers. A 2014 Forrester survey of 112 companies found that organizations in which there was systematic investment in UX design process and user research self-evaluated as having greater impact than those with more limited scope of investment.

These trends describe a new context that often finds agile teams unprepared for two main reasons. First, while the agile process formally values the principle of collaboration with customers to define the product vision, we and our colleagues in industry too often observe this princi-

ple not being put into practice: teams do not validate requirements systematically in the settings of use. Second, even when customers are involved, sometimes the teams may still fail to involve actual end users. As Rosenberg puts it, when user requirements are not validated but are still called "user stories," it creates "the illusion of user requirements" that fools the team and the executives, who are then mystified when the product fails in the marketplace.[10]

In this Viewpoint, we illustrate five classic examples of failures to involve actual end users or to gather sufficiently comprehensive data to represent their needs. Then we propose how these failures can be avoided.

Five Cases of Neglect or Mischaracterizations of the User

We identified five classic cases of failures to involve actual end users.

The Wild West case. The first and most obvious case occurs when the team does not do regular testing with the users along the development process. Thus the team fails to evaluate how well the software built fits target users, their tasks, and their environments. A real-life example of this failure is the development and deployment of Healthcare.org, where the team, admittedly, did not fully test the online health insurance marketplace until two weeks before it opened to the public on October 1, 2013. Then the site ran into major failures.[8]

Chooser ≠ target user. The second case is neither new nor unique to agile. The term "customer" conflates the chooser with the user. Let's unpack these words:

- A customer is often an organization (the target buyer of enterprise software, that is, product chooser) as represented by the purchasing officer, an executive or committee that makes a buying decision.
- A customer is the target user only for consumer-facing products. For enterprise software, target users may be far from the process of choosing a product, and have no input about products the organization selects.

Agile terminology adds to the confusion: product teams write *user stories* from the perspective of the person who uses the software, not the one who chooses it. Then a *customer demo* (or stakeholder review) at the end of an iteration confirms that each *user story* is satisfied. Here is when the terms customer and user are conflated. For enterprise software and large systems, practice teaches us that often the "end-of-iteration customer" is someone representing the product chooser rather than the end user.

Agile teams without user research are prone to building the wrong product.

So the end-of-iteration demo cannot be the sole form of feedback to predict user adoption and satisfaction. In addition, the software development team should also leverage user research to answer questions such as:

- What are the classes of users (personas)?
- Have we validated that the intended users have the needs specified in the user stories?
- What are the current user practices before the introduction of the product and the impact afterward?
- How would we extend the tool to support new personas or future use cases?

Internal proxies ≠ target user. The third case is about bias. Some teams work with their in-house professional services or sales support staff (that is, experts thought to represent large groups of customers) as proxies for end users. While we appreciate the expertise and knowledge these resources bring, we are wary of two common types of misrepresentation in these situations.

First, internal proxies are unrepresentative as end users because they have multiple unfair advantages: they know the software inside out, including the work-arounds; have access to internal tools unavailable to external customers; and do not need to use the product within the target users' time constraints or digital environment.

Second, the evidence internal proxies bring to the team is also biased. Professional sales and support staff are more likely to channel the needs of the largest or most strategic existing customers in the marketplace. They are more likely to focus on pain points of existing customers and less on what works well. Also, they may ignore new requirements that are not yet addressed by the current tool or market.

Therefore internal staff cannot be the sole representative of "users"—as shown in the "Dilbert" comic strip at the beginning of this column. User research welcomes their comments about competitive analysis, current insights about information architecture or other issues, which complement customer support data, UX research, and other sources of user feedback.

Executives liking sales demos ≠ target users adopting product. Enterprise software companies, during their annual customer conferences, use a sales demo to portray features and functions intended to excite the audience of buyers, investors, and the market analysts about the company strategy. However, positive responses to the sales demos should not be taken as equivalent to assertions about a product's user requirements. Instead, these requirements need confirmation via a careful validation cycle. Let sales demos open a door toward users with the help of choosers and influencers.

Similarly, Customer Advisory Boards (which draw from customers who have large installations, or who represent a specific or important segment of the market) stand in for all customers and offer additional opportunities to showcase future features or strategy. However, a basic law for success in the software industry is "Build Once, Sell Many."[7] This principle creates an inherent tension between satisfying current customers and attracting new ones. Therefore, a software company needs to constantly rethink their tiered offerings to include new market segments or customer classes as these emerge, and avoid one-off development efforts.

Confusing business leaders with users or the sales demo with the product prototype leads companies to build products based on what sales and product managers believe is awesome (for example, see Loranger[6]). Instead, we advocate validating the designs with actual end users during the product development.

Big data (What? When?) < The full picture (... How? Why?). Collecting and analyzing big data about digital product use is popular among product managers and even software developers, who can now learn what features get traction with users. We support the use of big data techniques *as part of* user research and user-centered design, but not as a substitute for qualitative user research. Let's review two familiar ways to use big data on usage: user data analytics and A/B testing.

User data analytics can quickly answer questions about current usage: quantity and most frequent patterns, such as How many? How often? When? Where? Once a product team has worked out most of the design (interaction patterns, page layouts, and more), A/B testing compares design alternatives, such as "which image on a page produces more clickthroughs"? In vivo experiments with sufficient traffic can generate large amount of useful data. Thus, A/B testing is very helpful for small incremental adjustments.

Every software company is in the business of finding and keeping new customers. Suppose the logs show the subscribers of an online dating application are not renewing. Should the company rejoice or despair? If people are getting good matches, and thus are satisfied, non-renewal implies success. If they are hopelessly disappointed by not getting dates, non-renewal implies failure. Big data won't tell you which, but observing and listening to even a handful of non-renewing individuals will.

In brief, quantitative data is useful but has two limitations: First, it will not tell the team why the current features are or are not used.[5] Different classes of users can have different reasons. Second, it will not identify what additional or alternative features appeal to a new class of users unfamiliar with the product. To answer these questions the team needs to rely on qualitative research with existing and proposed classes of users.

Market Research ≠ User Research

Finally, we point to the growing and worrisome tendency in industry to mix up user research with market research.

Market research groups make great partners for user research. While user research and market research have a few techniques in common (for example, surveys and focus groups), the goals and variables they focus on are different.

▸ Market research seeks to understand attitudes toward products, categories, or brands, and tries to predict the *likelihood of purchase*, engagement, or subscription.

▸ User research aims at improving the user experience by understanding the relation between actual usage behaviors and the properties of the design. To this end, it measures the behavior and attitudes of users thereby learning whether the product (or service) is usable, useful and delightful, *including after decision to purchase*.

We urge organizations to act strategically and connect market research, user research, and customer success functions. This requires aligning goals and sharing data among Marketing, Sales, Customer Success, and the UX Team (typically in Product or R&D).[1,4]

The Way Forward: Educate Managers and Agile Development Teams

We have shown five different ways that agile teams without user research are prone to building the wrong product. To avoid such failures, we invite software managers and product teams to assess and fill the current gap in a team's competencies. The closing table gives short-term and longer-term action items to address the gaps.

Actions to address gaps in UX competencies.

Short term

1. **Analyze the current skills of the team and flag the gap.** A functional product team needs several key skill sets or UX competencies: UX research, UX design, UI software development and prototyping.[11] These might be filled by training the current team members or hiring UX professionals full-time or part-time.
2. **Support product managers (or product owners) with investment in UX.** Too often, product managers find their role is a sort of "kitchen sink" for any task that is not software development. We encourage product managers to find additional resources in the UX competencies, to benefit both product and their workload.

Longer term

3. **Integrate UX competencies**
   a. Teams need UX research competencies as well as UX design skills (interaction, visual). Other related skill sets include content development and documentation; accessibility; globalization and localization.
4. **Collect and prioritize findings from user research**
   a. Seek user feedback early and often.
   b. Create channels to learn from end users and appropriate surrogates.
   c. Prioritize UX issues during backlog grooming; remove friction and measure delight.
   d. Build new features only after steps 4.a.–c. are done for each key version of the product.

References

1. Buley, L. The modern UX organization. *Forrester Report*. (2016); https://vimeo.com/121037431
2. Grudin J. *From Tool to Partner: The Evolution of Human-Computer Interaction*. Morgan & Claypool, 2017.
3. HP report. *Agile Is the New Normal: Adopting Agile Project Management*. 4AA5-7619ENW, May 2015.
4. Kell, E. Interview by Steve Portigal. *Portigal* blog. Podcast and transcript. (Mar. 1, 2016); http://www.portigal.com/podcast/10-elizabeth-kell-of-comcast/
5. Klein, L. *UX for Lean Startups: Faster, Smarter User Experience Research and Design*. O'Reilly, 2013.
6. Loranger, H. UX Without User Research Is Not UX. (Aug. 10, 2014) *Nielsen Norman Group* blog. http://www.nngroup.com/articles/ux-without-user-research/
7. Mironov, R. *Four Laws Of Software Economics*. Part 2: Law of Build Once, Sell Many. (Sept. 14, 2015); http://www.mironov.com/4law2/
8. Pear, R. Contractors Describe Limited Testing of Insurance Web Site. *New York Times* (Oct. 24, 2013); http://nyti.ms/292NryG
9. Perez, S. Users have low tolerance for buggy apps. *Techcrunch*. (Mar 12, 2013);[http://tcrn.ch/Y80ctA
10. Rosenberg, D. Introducing the business of UX. *Interactions*. Forums. XXI.1 Jan.–Feb. 2014.
11. Spool, J.M. Assessing your team's UX skills. *UIE*. (Dec. 10, 2007); https://www.uie.com/articles/assessing_ux_teams/

Gregorio Convertino (gconvertino@informatica.com) is a UX manager and principal user researcher at Informatica LLC.

Nancy Frishberg (nancyf@acm.org) is a UX researcher and strategist, in private practice, and a 25+-year member of the local SIGCHI Chapter BayCHI.org.

DOI:10.1145/3012006

Andrew Conway and Peter Eckersley

Viewpoint

When Does Law Enforcement's Demand to Read Your Data Become a Demand to Read Your Mind?

On cryptographic backdoors and prosthetic intelligence.

IMAGE COLLAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

THE RECENT DISPUTE between the FBI and Apple has raised a potent set of questions about companies' right to design strong cryptographic protections for their customers' data. The real stakes in these questions are not just whether the security of our devices should be weakened to facilitate FBI investigations, but ultimately, the ability of law enforcement and intelligence agencies to read our minds and most intimate private thoughts.

In the U.S. and other countries, there have been many legal cases in recent years pitting the demands of law enforcement against the concerns of technology companies and privacy advocates over access to new, technologically generated, information about people. The disputed topics have included spy agencies' bulk collection of Internet traffic and mobile phone metadata; law enforcement use of location-tracking devices, malware, and fake cellphone towers; the constitutionality of "gag orders" that make it a crime for individuals and companies to ever discuss certain requests they receive for others' data.

In some sense, this is not a new debate; the Fourth Amendment to the U.S. constitution, for instance, has engendered a long history of litigation about the boundaries between types of information that the police can obtain about people simply by demanding it with letters called *subpoenas*, and information for which a court-issued warrant is necessary. What has changed are the stakes of these disputes.

As the law has operated in the past, almost any information was theoretically fair game for law enforcement to demand if it had probable cause and obtained a warrant. But there was not nearly as much to collect: people did not carry recording and tracking devices with them everywhere, and they did not turn over the most intimate details of their lives to multinational technology companies. There were

also legal limits: the private thoughts of defendants were largely protected by rights to remain silent and against self-incrimination—historical legal protections that sprang up as shields against religious persecution. Unfortunately, changes to our lifestyles, to our relationship with technology, and to the very process of human cognition are making these protections so impractical that they may cease to exist at all.

So, what do we mean by changes to the process of human cognition?

Pens and paper are wonderful things. "Hang on. Let me write that down," or "I need a pen and paper to work this out," are the kinds of utterances that reveal our dependence. It is intelligence that makes us human, and a pen and paper magnifies our intelligence.

If you doubt this, consider any reasonable method of measuring intelligence. A human with a pen and paper will perform at least as well as, and often much, much better than, the same human without a pen and paper. So it would be reasonable to state that the pen and paper constituted a prosthetic component of our intelligence, or at least a prosthetic aid for our imperfect memory.

Furthermore, to read someone else's notes is often described as a window into their mind. Reading someone else's diary without their permission seems not only to be a violation of privacy but perhaps a form of taboo mind reading.

Now consider the same human having access to Google, Wikipedia, GPS, a calculator, a mobile phone to communicate with friends and colleagues, and indeed the whole Internet. As long as cat videos are not too much of a distraction, this well-resourced human can answer hard questions and perform many difficult tasks much more quickly than people even two decades earlier.

As hunters, weapons were prosthetic claws. As gatherers, baskets were prosthetic arms. After the development of agriculture, horses and plows were huge prosthetic muscles. Later the industrial revolution made us physically strong to a level unimaginable beforehand. And looking back, the invention of writing was the first step on the road to a modern existence built on prosthetic intelligence, one where the states we share through the Internet and the financial system are becoming more important than the biological and physical environment around us.

We have no choice but to pour our minds out if we want to exist and perform at the same level as the humans around us.

But this has come at a complicated price. You can think faster and more accurately, but your electronic devices know where you are, where you have been, who you have talked to, what you said, what your heart rate was at the time, what you have looked at on the Web, what medication you are taking, what you have bought, what maps you have looked up, what spelling mistakes you make, and it is only accelerating. With virtual reality and augmented reality looking imminent, gadgets will begin to log almost every action we take. And we have no choice but to pour our minds out if we want to exist and perform at the same level as the humans around us.

Ignoring arguments about precise definitions of words, it is clear that many humans in the developed world have a lot of their thoughts happening, or at least observable, outside of their brain, and this is only likely to increase in the future. It is through this lens that we need to understand the importance of Apple's fight to use encryption to protect some (presently very small) portions of its customers' data so that Apple (and transitively, the FBI) cannot read it. The FBI wants to be able to turn over literally every digital stone in its investigation. But in the era of prosthetic intelligence, that is equivalent to outlawing strong privacy for any corners of the modern human mind.

Calendar of Events

September 2
APSys '17: 8th Asia-Pacific Workshop on Systems, Mumbai, India, Sponsored: ACM/SIG, Contact: Purushottam Kulkarni, Email: puru@cse.iitb.ac.in

September 3–9
ICFP '17: ACM SIGPLAN International Conference on Functional Programming, Oxford, U.K., Sponsored: ACM/SIG, Contact: Jeremy Gibbons, Email: jeremy.gibbons@cs.ox.ac.uk

September 4–7
DocEng '17: ACM Symposium on Document Engineering 2017, Valletta, Malta, Sponsored: ACM/SIG, Contact: Kenneth P. Camilleri, Email: kenneth.camilleri@um.edu.mt

September 4–7
MobileHCI '17: 19th International Conference on Human-Computer Interaction with Mobile Devices and Services, Vienna, Austria, Sponsored: ACM/SIG

September 4–8
ESEC/FSE'17: Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, Paderborn, Germany, Sponsored: ACM/SIG, Contact: Wilhelm Schaefer, Email: wilhelm@uni-paderborn.de

September 6–8
WomEncourage '17: ACM-W Europe womENcourage Celebration of Women in Computing, Barcelona, Spain, Sponsored: ACM/SIG, Contact: Núria Castell Ariño, Email: castell@lsi.upc.edu

Where is this heading? Consider a future technological innovation—a brain reader. It is a little device that you attach to your skull that lets someone read your thoughts. This could be a great boon to law enforcement. Trials could be conducted more accurately by reading the thoughts of the defendant. Even better, everyone could be required to daily attend a mind reading to make sure they are not plotting any criminal acts. This would significantly cut down on premeditated crime, making our lives safer. Then we can concentrate on unpremeditated crime. Possibly there are some thoughts that people who are likely to commit unpremeditated crimes might think. We can proscribe those thoughts, and then preemptively arrest people for thought crime. While we are at it, the morality police can put in laws against thinking racist, sexist, extremist, sacrilegious, offensive, or fattening thoughts.

While such an extreme society may have a low crime rate, some people (including us) may think this police state would not actually be a better society to live in. Even ignoring the horrors that would result from imperfect readings, who doesn't feel guilty about something? As attributed to Cardinal Richelieu, "If you give me six lines written by the hand of the most honest of men, I will find something in them which will hang him." Such devices do not exist yet, although the demand has been strong enough that polygraphs, notorious for unreliability, are widely used in the U.S. Other technologies like fMRI are already being used and may turn out to be slightly more accurate than polygraphs, but we are still some distance from having to worry about the societal effects of active mind-reading machines.

What we have instead is a society moving toward prosthetic brains that can be monitored at all times by the state, without the inconvenience of having to have everyone check in each day at the police station. It may feel less invasive to have one's eye movements recorded by your augmented reality glasses when an attractive member of the opposite sex walks past than to have a daily visit to the mind reader. The former is certainly more convenient than the latter. But practically speaking, the effects are the same.

With access to a vast store of reference information massive deductions can be made.

The available information is not complete, and there will be gaps. But you can inference an awful amount with limited data. Think about how well you know your friends, and how you can often predict what decisions they will make, with only the small view of their world that you get from your interactions with them. With access to a vast store of reference information massive deductions can be made.

Conversely, the possibility of faulty deductions is itself a threat to individuals. You would not want to have performed Internet searches for pressure cookers and backpacks just before the Boston marathon bombings.

Dedicated, well-meaning people in law enforcement naturally want to be able to do their jobs better and make the world a safer, and thus better, place. They see the new data as a boon, and law enforcement agencies select extremely unphotogenic criminals and terrorists as the test cases that will set the rules for millions of other people. Unfortunately, while this surveillance apparatus may occasionally be useful, it also poses a structural threat to democracy.

Even beyond the threat of police states in the Western world and elsewhere, there is a fundamental issue with cryptography that mathematics works the same regardless of whether you are naughty or nice. So if the state can break cryptography then so can other actors. There are obvious direct applications to crime—knowing when someone is away from home; knowing who is worth kidnapping and what their movements are; identity theft, bank fraud, and so forth. But ineffective cryptography also strengthens the black market for industrial espionage—many people would pay to know the thoughts of their competitors, people they are negotiating with, or even people they are considering going on a date with.

Of course the state is not the only institution that wants to read your mind. There is great value to corporations in knowing about you. They collect this data from phone apps and operating systems, credit cards, and web browsers; they use it to help design their products, but also for targeted advertising, differential pricing, and other debatable purposes. People joke, semi-seriously, that Google knows you better than you know yourself. As well as being a threat in their own right, corporations provide an additional target of attack for an intrusive state: as Snowden's leaks revealed, the NSA didn't try to track the location of every cellphone on the planet directly: they let advertisements and tracking code in apps collect the data for them.

Ultimately, the question of what to do about the data accumulated by technology companies is different from the question of what to do about the FBI, but it should also be understood that we have largely given these companies the power to read our minds, and might want to find alternatives to that arrangement.

We fear we are slowly moving toward the era of universal mind monitoring without having recognized and considered it in those terms. And those are the terms in which we should understand battles about the right to use effective cryptography. That wonderful gadget in your pocket is not a phone. It is a prosthetic part of your mind—which happens to also be able to make telephone calls. We need to think of it as such, and ask again which parts of our thoughts should be categorically shielded against prying by the state. C

Andrew Conway (andrewed@greatcactus.org) is an engineer and mostly retired entrepreneur. He founded and ran Silicon Genetics.

Peter Eckersley (pde@eff.org) is Chief Computer Scientist for the Electronic Frontier Foundation, San Francisco, CA.

DESIGN • INNOVATE • SUSTAIN

idc2018

INTERACTION DESIGN AND CHILDREN

JUNE 19–22

TRONDHEIM NORWAY

The IDC Conference Series has brought together researchers and practitioners seeking to study how best to develop and design interactive technologies for children.

idc.acm.org

bit.ly/idc2018

@IDC_ACM

Association for Computing Machinery

SIGCHI

NTNU
Norwegian University of Science and Technology

DOI:10.1145/3080202

Article development led by acmqueue
queue.acm.org

You're only as available as the sum of your dependencies.

BY BEN TREYNOR, MIKE DAHLIN, VIVEK RAU, AND BETSY BEYER

The Calculus of Service Availability

AS DETAILED IN *Site Reliability Engineering: How Google Runs Production Systems*[1] (hereafter referred to as the SRE book), Google products and services seek high-velocity feature development while maintaining aggressive service-level objectives (SLOs) for availability and responsiveness. An SLO says that the service should *almost always* be up, and the service should *almost always* be fast; SLOs also provide precise numbers to define what "almost always" means for a particular service. SLOs are based on the following observation:

The vast majority of software services and systems should aim for almost-perfect reliability rather than perfect reliability—that is, 99.999% or 99.99% rather than 100%—because users cannot tell the difference between a service being 100% available and less than "perfectly" available. There are many other systems in the path between user and service (laptop, home WiFi, ISP, the power grid ...), and those systems collectively are far less than 100% available. Thus, the marginal difference between 99.99% and 100% gets lost in the noise of other unavailability, and the user receives no benefit from the enormous effort required to add that last fractional percent of availability. Notable exceptions to this rule include antilock brake control systems and pacemakers!

For a detailed discussion of how SLOs relate to SLIs (service-level indicators) and SLAs (service-level agreements), see the "Service Level Objectives" chapter in the SRE book. That chapter also details how to choose metrics that are meaningful for a particular service or system, which in turn drives the choice of an appropriate SLO for that service.

This article expands upon the topic of SLOs to focus on service dependencies. Specifically, we look at how the availability of critical dependencies informs the availability of a service, and how to design in order to mitigate and minimize critical dependencies.

Most services offered by Google aim to offer 99.99% (sometimes referred to as the "four 9s") availability to users. Some services contractually commit to a lower figure externally but set a 99.99% target internally. This more stringent target accounts for situations in which users become unhappy with service performance well before a contract violation occurs, as the number one aim of an SRE team is to keep users happy. For many services, a 99.99% internal target represents the sweet spot that balances cost, complexity, and availability. For some services, notably global cloud services, the internal target is 99.999%.

99.99% Availability: Observations And Implications

Let's examine a few key observations about and implications of designing and operating a 99.99% service and then move to a practical application.

Observation 1. Sources of outages. Outages originate from two main sources: problems with the service it-

IMAGE BY PLING/SHUTTERSTOCK

self and problems with the service's critical dependencies. A critical dependency is one that, if it malfunctions, causes a corresponding malfunction in the service.

Observation 2. The mathematics of availability. Availability is a function of the frequency and the duration of outages. It is measured through:

- Outage *frequency*, or the inverse: MTTF (mean time to failure).
- *Duration*, using MTTR (mean time to repair). Duration is defined as it is experienced by users: lasting from the start of a malfunction until normal behavior resumes.

Thus, availability is mathematically defined as MTTF/(MTTF+MTTR), using appropriate units.

Implication 1. Rule of the extra 9. A service cannot be more available than the intersection of all its critical dependencies. If your service aims to offer 99.99% availability, then all of your critical dependencies must be significantly more than 99.99% available.

Internally at Google, we use the following rule of thumb: critical dependencies must offer one additional 9 relative to your service—in the example case, 99.999% availability—because any service will have several critical dependencies, as well as its own idiosyncratic problems. This is called the "rule of the extra 9."

If you have a critical dependency that does not offer enough 9s (a relatively common challenge!), you must employ mitigation to increase the effective availability of your dependency (for example, via a *capacity cache*, *failing open*, *graceful degradation* in the face of errors, and so on.)

Implication 2. The math vis-à-vis frequency, detection time, and recovery time. A service cannot be more available than its incident frequency multiplied by its detection and recovery time. For example, three complete outages per year that last 20 minutes each result in a total of 60 minutes of outages. Even if the service worked perfectly the rest of the year, 99.99% availability (no more than 53 minutes of downtime per year) would not be feasible.

This implication is just math, but it is often overlooked, and can be very inconvenient.

Corollary to implications 1 and 2. If your service is relied upon for an availability level you cannot deliver, you should make energetic efforts to correct the situation—either by increasing the availability level of your service or by adding mitigation as described earlier. Reducing expectations (that is, the published availability) is also an option, and often it is the correct choice: make it clear to the dependent service that it should either reengineer its system to compensate for your service's availability or reduce its own tar-

Key Definitions

Some of the terms and concepts used throughout this article may not be familiar to readers who don't specialize in operations.

Capacity cache: A cache that serves precomputed results for API calls or queries to a service, generating cost savings in terms of compute/IO resource needs by reducing the volume of client traffic hitting the underlying service.

Unlike the more typical performance/latency cache, a capacity cache is considered critical to service operation. A drop in the cache hit rate or cache ratio below the SLO is considered a capacity loss. Some capacity caches may even sacrifice performance (for example, redirecting to remote sites) or freshness (for example, CDNs) in order to meet hit rate SLOs.

Customer isolation: Isolating customers from each other may be advantageous so that the behavior of one customer doesn't impact other customers. For example, you might isolate customers from one another based on their global traffic. When a given customer sends a surge of traffic beyond what they're provisioned for, you can start throttling or rejecting this excess traffic without impacting traffic from other customers.

Failing safe/failing open/failing closed: Strategies for gracefully tolerating the failure of a dependency. The "safe" strategy depends on context: failing open may be the safe strategy in some scenarios, while failing closed may be the safe strategy in others.

Failing open: When the trigger normally required to authorize an action fails, failing open means to let some action happen, rather than making a decision. For example, a building exit door that normally requires badge verification "fails open" to let you exit without verification during a power failure.

Failing closed is the opposite of falling open. For example, a bank vault door denies all attempts to unlock it if its badge reader cannot contact the access-control database.

Failing safe means whatever behavior is required to prevent the system from falling into an unsafe mode when expected functionality suddenly doesn't work. For example, a given system might be able to *fail open* for a while by serving cached data, but then *fail closed* when that data becomes stale (perhaps because past a certain point, the data is no longer useful).

Failover: A strategy that handles failure of a system component or service instance by automatically routing incoming requests to a different instance. For example, you might route database queries to a replica database, or route service requests to a replicated server pool in another datacenter.

Fallback: A mechanism that allows a tool or system to use an alternative source for serving results when a given component is unavailable. For example, a system might fall back to using an in-memory cache of previous results. While the results may be slightly stale, this behavior is better than outright failure. This type of fallback is an example of graceful degradation.

Geographic isolation: You can build additional reliability into your service by isolating particular geographic zones to have no dependencies on each other. For example, if you separate North America and Australia into separate serving zones, an outage that occurs in Australia because of a traffic overload won't also take out your service in North America. Note that geographic isolation does come at increased cost: isolating these geographic zones also means that Australia cannot borrow spare capacity in North America.

Graceful degradation: A service should be "elastic" and not fail catastrophically under overload conditions and spikes—that is, you should make your applications do something reasonable even if not all is right. It is better to give users limited functionality than an error page.

Integration testing: The phase in software testing in which individual software modules are combined and tested as a group to verify that they function correctly together. These "parts" may be code modules, individual applications, client and server applications on a network, among others. Integration testing is usually performed after unit testing and before final validation testing.

Operational readiness practice: Exercises designed to ensure the team supporting a service knows how to respond effectively when an issue arises, and that the service is resilient to disruption. For example, Google performs disaster-recovery test drills continuously to make sure that its services deliver continuous uptime even if a large-scale disaster occurs.

Rollout policy: A set of principles applied during a service rollout (a deployment of any sort of software component or configuration) to reduce the scope of an outage in the early stages of the rollout. For example, a rollout policy might specify that rollouts occur progressively, on a 5%/20%/100% timeline, so that a rollout proceeds to a larger portion of customers only when it passes the first milestone without problems. Most problems will manifest when the service is exposed to a small number of customers, allowing you to minimize the scope of the damage. Note that for a rollout policy to be effective in minimizing damage, you must have a mechanism in place for rapid rollback.

Rollback: This is the ability to revert a set of changes that have been previously rolled out (fully or not) to a given service or system. For example, you can revert configuration changes or run a previous version of a binary that's known to be good.

Sharding: Splitting a data structure or service into shards is a management strategy based on the principle that systems built for a single machine's worth of resources don't scale. Therefore, you can distribute resources such as CPU, memory, disk, file handles, and so on across multiple machines to create smaller, faster, more easily managed parts of a larger whole.

Tail latency: When setting a target for the latency (response time) of a service, it is tempting to measure the average latency. The problem with this approach is that an average that looks acceptable can hide a "long tail" of very large outliers, where some users may experience terrible response times. Therefore, the SRE best practice is to measure and set targets for 95th- and/or 99th-percentile latency, with the goal of reducing this tail latency, not just average latency.

get. If you do not correct or address the discrepancy, an outage will inevitably force the need to correct it.

Practical Application

Let's consider an example service with a target availability of 99.99% and work through the requirements for both its dependencies and its outage responses.

The numbers. Suppose your 99.99% available service has the following characteristics:

- One major outage and three minor outages of its own per year. Note that these numbers sound high, but a 99.99% availability target implies a 20- to 30-minute widespread outage and several short partial outages per year. (The math makes two assumptions: that a failure of a single shard is not considered a failure of the entire system from an SLO perspective, and that the overall availability is computed with a weighted sum of regional/shard availability.)
- Five critical dependencies on other, independent 99.999% services.
- Five independent shards, which cannot fail over to one another.
- All changes are rolled out progressively, one shard at a time.

The availability math plays out as follows.

Dependency requirements.

- The total budget for outages for the year is 0.01% of 525,600 minutes/year, or 53 minutes (based on a 365-day year, which is the worst-case scenario).
- The budget allocated to outages of critical dependencies is five independent critical dependencies, with a budget of 0.001% each = 0.005%; 0.005% of 525,600 minutes/year, or 26 minutes.
- The remaining budget for outages caused by your service, accounting for outages of critical dependencies, is 53 - 26 = 27 minutes.

Outage response requirements.

- Expected number of outages: 4 (1 full outage, 3 outages affecting a single shard only)
- Aggregate impact of expected outages: (1 x 100%) + (3 x 20%) = 1.6
- Time available to detect and recover from an outage: 27/1.6 = 17 minutes
- Monitoring time allotted to detect and alert for an outage: 2 minutes
- Time allotted for an on-call responder to start investigating an alert: five minutes. (*On-call* means that a technical person is carrying a pager that receives an alert when the service is having an outage, based on a monitoring system that tracks and reports SLO violations. Many Google services are supported by an SRE on-call rotation that fields urgent issues.)
- Remaining time for an effective mitigation: 10 minutes

Implication. Levers to make a service more available. It's worth looking closely at the numbers just presented because they highlight a fundamental point: there are three main levers to make a service more reliable.

- Reduce the frequency of outages—via *rollout policy*, testing, design reviews, and other tactics.
- Reduce the scope of the average outage—via *sharding*, *geographic isolation*, *graceful degradation*, or *customer isolation*.
- Reduce the time to recover—via monitoring, one-button safe actions (for example, *rollback* or adding emergency capacity), *operational readiness practice*, and so on.

You can trade among these three levers to make implementation easier. For example, if a 17-minute MTTR is difficult to achieve, instead focus your efforts on reducing the scope of the average outage. Strategies for minimizing and mitigating critical dependencies are discussed in more depth later in this article.

Clarifying the "Rule of the Extra 9" for Nested Dependencies

A casual reader might infer that each additional link in a dependency chain calls for an additional 9, such that second-order dependencies need two extra 9s, third-order dependencies need three extra 9s, and so on.

This inference is incorrect. It is based on a naive model of a dependency hierarchy as a tree with constant fan-out at each level. In such a model, as shown in Figure 1, there are 10 unique first-order dependencies, 100 unique second-order dependencies, 1,000 unique third-order dependencies, and so on, leading to a total of 1,111 unique services even if the architecture is limited to four layers. A highly available service ecosystem with that many independent critical dependencies is clearly unrealistic.

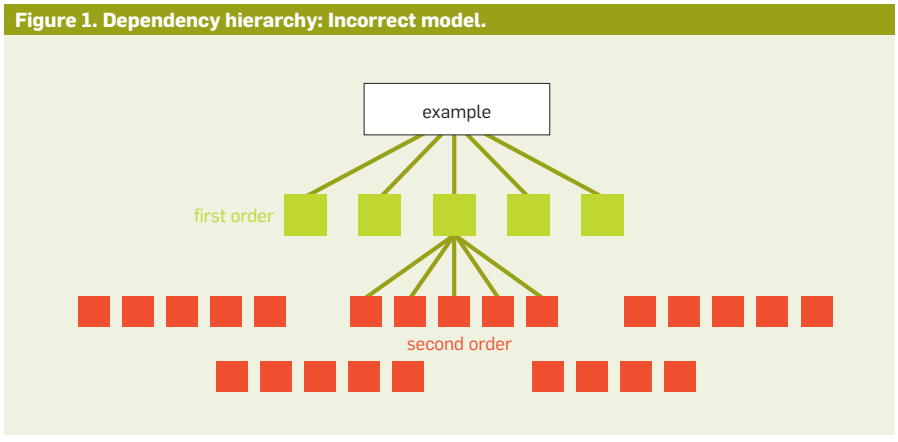
A critical dependency can by itself cause a failure of the entire service (or service shard) no matter where it appears in the dependency tree. Therefore, if a given component *X* appears as a dependency of several first-order dependencies of a service, *X* should be counted only once because its failure will ultimately cause the service to fail no matter how many intervening services are also affected.

The correct rule is as follows:

- If a service has *N* unique critical dependencies, then each one contributes 1/*N* to the dependency-induced unavailability of the top-level service, regardless of its depth in the dependency hierarchy.
- Each dependency should be counted only once, even if it appears multiple times in the dependency hierarchy (in other words, count only unique dependencies). For example, when counting dependencies of Service A in Figure 2, count Service B only once toward the total *N*.

For example, consider a hypothetical Service A, which has an error

Figure 1. Dependency hierarchy: Incorrect model.



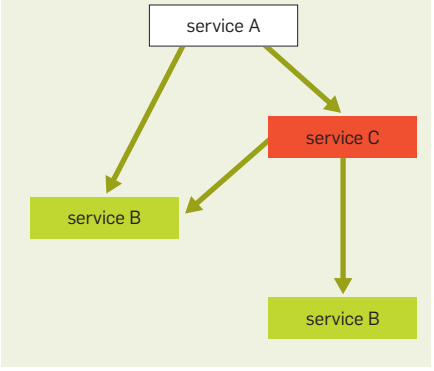
budget of 0.01%. The service owners are willing to spend half that budget on their own bugs and losses, and half on critical dependencies. If the service has *N* such dependencies, each dependency receives 1/Nth of the remaining error budget. Typical services often have about five to 10 critical dependencies, and therefore each one can fail only one-tenth or one-twentieth as much as Service A. Hence, as a rule of thumb, a service's critical dependencies must have one extra 9 of availability.

Error Budgets

The concept of error budgets is covered quite thoroughly in the SRE book,[1] but bears mentioning here. Google SRE uses error budgets to balance reliability and the pace of innovation. This budget defines the acceptable level of failure for a service over some period of time (often a month). An error budget is simply 1 minus a service's SLO, so the previously discussed 99.99% available service has a 0.01% "budget" for unavailability. As long as the service hasn't spent its error budget for the month, the development team is free (within reason) to launch new features, updates, and so on.

If the error budget is spent, the service freezes changes (except for urgent security fixes and changes addressing what caused the violation in the first place) until either the service earns back room in the budget, or the month resets. Many services at Google use sliding windows for SLOs, so the error budget grows back gradually. For mature services with an SLO greater than 99.99%, a quarterly rather than monthly budget reset is appropriate, because the amount of allowable downtime is small.

Figure 2. Multiple dependencies in the dependency hierarchy.



Error budgets eliminate the structural tension that might otherwise develop between SRE and product development teams by giving them a common, data-driven mechanism for assessing launch risk. They also give both SRE and product development teams a common goal of developing practices and technology that allow faster innovation and more launches without "blowing the budget."

Strategies for Minimizing and Mitigating Critical Dependencies

Thus far, this article has established what might be called the "Golden Rule of Component Reliability." This simply means that any critical component must be 10 times as reliable as the overall system's target, so that its contribution to system unreliability is noise. It follows that in an ideal world, the aim is to make as many components as possible noncritical. Doing so means the components can adhere to a lower reliability standard, gaining freedom to innovate and take risks.

The most basic and obvious strategy to reduce critical dependencies is to eliminate single points of failure (SPOFs) whenever possible. The larger system should be able to operate acceptably without any given component that's not a critical dependency or SPOF.

In reality, you likely cannot get rid of all critical dependencies, but you can follow some best practices around system design to optimize reliability. While doing so isn't always possible, it is easier and more effective to achieve system reliability if you plan for reliability during the design and planning phases, rather than after the system is live and impacting actual users.

Conduct architecture/design reviews. When you are contemplating a new system or service, or refactoring or improving an existing system or service, an architecture or design review can identify shared infrastructure and internal vs. external dependencies.

Shared infrastructure. If your service is using shared infrastructure—for example, an underlying database service used by multiple user-visible products—think about whether or not that infrastructure is being used correctly. Be explicit in identifying the owners of shared infrastructure as additional stakeholders. Also, beware of overloading your dependencies—coordinate launches carefully with the owners of these dependencies.

Internal vs. external dependencies. Sometimes a product or service depends on factors beyond company control—for example, code libraries, or services or data provided by third parties. Identifying these factors allows you to mitigate the unpredictability they entail.

Engage in thoughtful system planning and design. Design your system with the following principles in mind.

Redundancy and isolation. You can seek to mitigate your reliance upon a critical dependency by designing that dependency to have multiple independent instances. For example, if storing data in one instance provides 99.9% availability for that data, then storing three copies in three widely distributed instances provides a theoretical availability level of 1 - 0.013, or nine 9s, if instance failures are independent with zero correlation.

In the real world, the correlation is never zero (consider network backbone failures that affect many cells concurrently), so the actual availability will be nowhere close to nine 9s but is much higher than three 9s. Also note that if a system or service is "widely distributed," geographic separation is not always a good proxy for uncorrelated failures. You may be better off using more than one system in nearby locations than the same system in distant locations.

Similarly, sending an RPC (remote procedure call) to one pool of servers in one cluster may provide 99.9% availability for results, but sending three concurrent RPCs to three different server pools and accepting the first response that arrives helps increase availability to well over three 9s (noted earlier). This strategy can also reduce *tail latency* if the server pools are approximately equidistant from the RPC sender. (Since there is a high cost to sending three RPCs concurrently, Google often stages the timing of these calls strategically: most of our systems wait a fraction of the allotted time before sending the second RPC,

and a bit more time before sending the third RPC.)

Failover and fallback. Pursue software rollouts and migrations that *fail safe* and are automatically isolated should a problem arise. The basic principle at work here is that by the time you bring a human online to trigger a *failover*, you have likely already exceeded your error budget.

Where concurrency/voting is not possible, automate failover and *fallback*. Again, if the issue needs a human to check what the problem is, the chances of meeting your SLO are slim.

Asynchronicity. Design dependencies to be asynchronous rather than synchronous where possible so that they don't accidentally become critical. If a service waits for an RPC response from one of its noncritical dependencies and this dependency has a spike in latency, the spike will unnecessarily hurt the latency of the parent service. By making the RPC call to a noncritical dependency asynchronous, you can decouple the latency of the parent service from the latency of the dependency. While asynchronicity may complicate code and infrastructure, this trade-off will be worthwhile.

Capacity planning. Make sure that every dependency is correctly provisioned. When in doubt, overprovision if the cost is acceptable.

Configuration. When possible, standardize configuration of your dependencies to limit inconsistencies among subsystems and avoid one-off failure/error modes.

Detection and troubleshooting. Make detecting, troubleshooting, and diagnosing issues as simple as possible. Effective monitoring is a crucial component of being able to detect issues in a timely fashion. Diagnosing a system with deeply nested dependencies is difficult. Always have an answer for mitigating failures that doesn't require an operator to investigate deeply.

Fast and reliable rollback. Introducing humans into a mitigation plan substantially increases the risk of missing a tight SLO. Build systems that are easy, fast, and reliable to roll back. As your system matures and you gain confidence in your monitoring to detect problems, you can lower MTTR by engineering the system to automatically trigger safe rollbacks.

Systematically examine all possible failure modes. Examine each component and dependency and identify the impact of its failure. Ask yourself the following questions:

- Can the service continue serving in degraded mode if one of its dependencies fails? In other words, design for graceful degradation.
- How do you deal with unavailability of a dependency in different scenarios? Upon startup of the service? During runtime?

Conduct thorough testing. Design and implement a robust testing environment that ensures each dependency has its own test coverage, with tests that specifically address use cases that other parts of the environment expect. Here are a few recommended strategies for such testing:

- Use *integration testing* to perform fault injection—verify that your system can survive failure of any of its dependencies.
- Conduct disaster testing to identify weaknesses or hidden/unexpected dependencies. Document follow-up actions to rectify the flaws you uncover.
- Don't just load test. Deliberately overload your system to see how it degrades. One way or another, your system's response to overload *will* be tested; better to perform these tests yourself than to leave load testing to your users.

Plan for the future. Expect changes that come with scale: a service that begins as a relatively simple binary on a single machine may grow to have many obvious and nonobvious dependencies when deployed at a larger scale. Every order of magnitude in scale will reveal new bottlenecks—not just for your service, but for your dependencies as well. Consider what happens if your dependencies cannot scale as fast as you need them to.

Also be aware that system dependencies evolve over time and that your list of dependencies may very well grow over time. When it comes to infrastructure, Google's typical design guideline is to build a system that will scale to 10 times the initial target load without significant design changes.

Conclusion

While readers are likely familiar with some or many of the concepts this article has covered, assembling this information and putting it into concrete terms may make the concepts easier to understand and teach. Its recommendations are uncomfortable but not unattainable. A number of Google services have consistently delivered better than four 9s of availability, not by superhuman effort or intelligence, but by thorough application of principles and best practices collected and refined over the years (see SRE's Appendix B: A Collection of Best Practices for Production Services).

Acknowledgments

Thank you to Ben Lutch, Dave Rensin, Miki Habryn, Randall Bosetti, and Patrick Bernier for their input.

Related articles on queue.acm.org

There's Just No Getting Around It: You're Building a Distributed System
Mark Cavage
http://queue.acm.org/detail.cfm?id=2482856

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

A Conversation with Wayne Rosing
David J. Brown
http://queue.acm.org/detail.cfm?id=945162

Reference

1. Beyer, B., Jones, C., Petoff, J., Murphy, N.R. *Site Reliability Engineering: How Google Runs Production Systems.* O'Reilly Media, 2016; https://landing.google.com/sre/book.html.

Ben Treynor started programming at age six and joined Oracle as a software engineer at age 17. He has also worked in engineering management at E.piphany, SEVEN, and Google (2003-present). His current team of approximately 4,200 at Google is responsible for Site Reliability Engineering, networking, and datacenters worldwide.

Mike Dahlin is a distinguished engineer at Google, where he has worked on Google's Cloud Platform since 2013. Prior to joining Google, he was a professor of computer science at the University of Texas at Austin.

Vivek Rau is an SRE manager at Google and a founding member of the Launch Coordination Engineering sub-team of SRE. Prior to joining Google, he worked at Citicorp Software, Versant, and E.piphany. He currently manages various SRE teams tasked with tracking and improving the reliability of Google's Cloud Platform.

Betsy Beyer is a technical writer for Google, specializing in Site Reliability Engineering. She has previously written documentation for Google's Data Center and Hardware Operations Teams. She was formerly a lecturer on technical writing at Stanford University.

DOI:10.1145/3080008

Article development led by acmqueue
queue.acm.org

The approximate approach is often faster and more efficient.

BY GRAHAM CORMODE

Data Sketching

DO YOU EVER feel overwhelmed by an unending stream of information? It can seem like a barrage of new email and text messages demands constant attention, and there are also phone calls to pick up, articles to read, and knocks on the door to answer. Putting these pieces together to keep track of what is important can be a real challenge.

The same information overload is a concern in many computational settings. Telecommunications companies, for example, want to keep track of the activity on their networks, to identify overall network health and spot anomalies or changes in behavior. Yet, the scale of events occurring is huge: many millions of network events per hour, per network element. While new technologies allow the scale and granularity of events being monitored to increase by orders of magnitude, the capacity of computing elements (processors, memory, and disks) to make sense of these is barely increasing. Even on a small scale, the amount of information may be too large to store in an impoverished setting (say, an embedded device) or to keep conveniently in fast storage.

In response to this challenge, the model of streaming data processing has grown in popularity. The aim is no longer to capture, store, and index every minute event, but rather to process each observation quickly in order to create a summary of the current state. Following its processing, an event is dropped and is no longer accessible. The summary that is retained is often referred to as a *sketch* of the data.

Coping with the vast scale of information means making compromises: The description of the world is approximate rather than exact; the nature of queries to be answered must be decided in advance rather than after the fact; and some questions are now insoluble. The ability to process vast quantities of data at blinding speeds with modest resources, however, can more than make up for these limitations.

As a consequence, streaming methods have been adopted in a number of domains, starting with telecommunications but spreading to search engines, social networks, finance, and time-series analysis. These ideas are also finding application in areas using traditional approaches, but where the rough-and-ready sketching approach is more cost effective. Successful applications of sketching involve a mixture of algorithmic tricks, systems know-how, and mathematical insight, and have led to new research contributions in each of these areas.

This article introduces the ideas behind sketching, with a focus on algorithmic innovations. It describes some algorithmic developments in the abstract, followed by the steps needed to put them into practice, with examples. The article also looks at four novel algorithmic ideas and discusses some emerging areas.

Simply Sampling

When faced with a large amount of information to process, there may be

PHOTO BY TAFFPIXTURE

a strong temptation just to ignore it entirely. A slightly more principled approach is just to ignore *most of it*—that is, take a small number of examples from the full dataset, perform the computation on this subset, and then try to extrapolate to the full dataset. To give a good estimation, the examples must be randomly chosen. This is the realm of sampling.

There are many variations of sampling, but this article uses the most basic: uniform random sampling. Consider a large collection of customer records. Randomly selecting a small number of records provides the sample. Then various questions can be answered accurately by looking only at the sample: for example, estimating what fraction of customers live in a certain city or have bought a certain product.

The method. To flesh this out, let's fill in a few gaps. First, how big should the sample be to supply good answers? With standard statistical results, for questions like those in the customer records example, the standard error of a sample of size s is proportional to $1/\sqrt{s}$. Roughly speaking, this means that in estimating a proportion from the sample, the error would be expected to look like $\pm 1/\sqrt{s}$. Therefore, looking at the voting intention of a subset of 1,000 voters produces an opinion poll whose error is approximately 3%—providing high confidence (but not certainty) that the true answer is within 3% of the result on the sample, assuming the sample was drawn randomly and the participants responded honestly. Increasing the size of the sample causes the error to decrease in a predictable, albeit expensive, way: reducing the margin of error of an opinion poll to 0.3% would require contacting 100,000 voters.

Second, how should the sample be drawn? Simply taking the first s records is not guaranteed to be random; there may be clustering through the data. You need to ensure every record has an equal chance of being included in the sample. This can be achieved by using standard random-number generators to pick which records to include in the sample. A common trick is to attach a random number to each record, then sort the data based on this random tag and take the first s records in the sorted order. This works fine, as long as sorting the full dataset is not too costly.

Finally, how do you maintain the sample as new items are arriving? A simple approach is to pick every record with probability p, for some chosen value of p. When a new record comes, pick a random fraction between 0 and 1, and if it is smaller than p, put the record in the sample. The problem with this approach is that you do not know in advance what p should be. In the

previous analysis a fixed sample size s was desired, and using a fixed sampling rate p means there are too few elements initially, but then too many as more records arrive.

Presented this way, the question has the appearance of an algorithmic puzzle, and indeed this was a common question in technical interviews for many years. One can come up with clever solutions that incrementally adjust p as new records arrive. A simple and elegant way to maintain a sample is to adapt the idea of random tags. Attach to each record a random tag, and define the sample to be the s records with the smallest tag values. As new records arrive, the tag values decide whether to add the new record to the sample (and to remove an old item to keep the sample size fixed at s).

Discussion and applications. Sampling methods are so ubiquitous that there are many examples to consider. One simple case is within database systems. It is common for the database management system to keep a sample of large relations for the purpose of query planning. When determining how to execute a query, evaluating different strategies provides an estimate of how much data reduction may occur at each step, with some uncertainty of course. Another example comes from the area of data integration and linkage, in which a subproblem is to test whether two columns from separate tables can relate to the same set of entities. Comparing the columns in full can be time consuming, especially when you want to test all pairs of columns for compatibility. Comparing a small sample is often sufficient to determine whether the columns have any chance of relating to the same entities.

Entire books have been written on the theory and practice of sampling, particularly around schemes that try to sample the more important elements preferentially, to reduce the error in estimating from the sample. For a good survey with a computational perspective, see *Synopses for Massive Data: Samples, Histograms, Wavelets and Sketches*.[11]

Given the simplicity and generality of sampling, why would any other method be needed to summarize data? It turns out that sampling is not well suited for some questions. Any question that requires detailed knowledge of individual records in the data cannot be answered by sampling. For example, if you want to know whether one specific individual is among your customers, then a sample will leave you uncertain. If the customer is not in the sample, you do not know whether this is because that person is not in the data or because he or she did not happen to be sampled. A question like this ultimately needs all the presence information to be recorded and is answered by highly compact encodings such as the Bloom filter (described later).

A more complex example is when the question involves determining the cardinality of quantities. In a dataset that has many different values, how many distinct values of a certain type are there? For example, how many distinct surnames are in a particular customer dataset? Using a sample does not reveal this information. Let's say in a sample size of 1,000 out of one million records, 900 surnames occur just once among the sampled names. What can you conclude about the popularity of these names in the rest of the dataset? It might be that almost every other name in the full dataset is also unique. Or it might be that each of the unique names in the sample reoccurs tens or hundreds of times in the remainder of the data. With the sampled information there is no way to distinguish between these two cases, which leads to huge confidence intervals on these kinds of statistics. Tracking information about cardinalities, and omitting duplicates, is addressed by techniques such as HyperLogLog, addressed later.

Finally, there are quantities that samples can estimate, but for which better special-purpose sketches exist. Recall that the standard error of a sample of size s is $1/\sqrt{s}$. For problems such as estimating the frequency of a particular attribute (such as city of residence), you can build a sketch of size s so the error it guarantees is proportional to $1/s$. This is considerably stronger than the sampling guarantee and only improves as we devote more space s to the sketch. The Count-Min sketch described later in this article has this property. One limitation is that the attribute of interest must be specified in advance of setting up the sketch, while a sample allows you to evaluate a query for any recorded attribute of the sampled items.

Because of its flexibility, sampling is a powerful and natural way of building a sketch of a large dataset. There are many different approaches to sampling that aim to get the most out of the sample or to target different types of queries that the sample may be used to answer.[11] Here, more information is presented about less flexible methods that address some of these limitations of sampling.

Summarizing Sets with Bloom Filters

The Bloom filter is a compact data structure that summarizes a set of items. Any computer science data-structures class is littered with examples of "dictionary" data structures, such as arrays, linked lists, hash tables, and many esoteric variants of balanced tree structures. The common feature of these structures is that they can all answer "membership questions" of the form: Is a certain item stored in the structure or not? The Bloom filter can also respond to such membership questions. The answers given by the structure, however, are either "the item has definitely not been stored" or "the item has *probably* been stored." This introduction of uncertainty over the state of an item (it might be thought of as introducing potential false positives) allows the filter to use an amount of space that is much smaller than its exact relatives. The filter also does not allow listing the items that have been placed into it. Instead, you can pose membership questions only for specific items.

The method. To understand the filter, it is helpful to think of a simple exact solution to the membership problem. Suppose you want to keep track of which of a million possible items you have seen, and each one is helpfully labeled with its ID number (an integer between one and a million). Then you can keep an array of one million bits, initialized to all 0s. Every time you see an item i, you just set the ith bit in the array to 1. A lookup query for item j is correspondingly straightforward: just see whether bit j is a 1 or a 0. The structure is very compact: 125KB will suffice if you pack the bits into memory.

Real data, however, is rarely this nicely structured. In general, you might have a much larger set of possible inputs—think again of the names of customers, where the number of possible name strings is huge. You can nevertheless adapt your bit-array approach by borrowing from a different dictionary structure. Imagine the bit array is a hash table: you will use a hash function h to map from the space of inputs onto the range of indices for your table. That is, given input i, you now set bit hi to 1. Of course, now you have to worry about hash collisions in which multiple entries might map onto the same bit. A traditional hash table can handle this, as you can keep information about the entries in the table. If you stick to your guns and keep the bits only in the bit array, however, false positives will result: if you look up item i, it may be that entry h_i is set to 1, but i has not been seen; instead, there is some item j that was seen, where h(i) = h(j).

Can you fix this while sticking to a bit array? Not entirely, but you can make it less likely. Rather than just hashing each item i once, with a single hash function, use a collection of k hash functions $h_1, h_2, \ldots h_k$, and map i with each of them in turn. All the bits corresponding to $h_1(i), h_2(i) \ldots h_k(i)$ are set to 1. Now to test membership of j, check all the entries it is hashed to, and say no if any of them are 0.

There's clearly a trade-off here: Initially, adding extra hash functions reduces the chances of a false positive as more things need to "go wrong" for an incorrect answer to be given. As more and more hash functions are added, however, the bit array gets fuller and fuller of 1 values, and therefore collisions are more likely. This trade-off can be analyzed mathematically, and the sweet spot found that minimizes the chance of a false positive. The analysis works by assuming that the hash functions look completely random (which is a reasonable assumption in practice), and by looking at the chance that an arbitrary element not in the set is reported as present.

If n distinct items are being stored in a Bloom filter of size m, and k hash functions are used, then the chance of a membership query that should receive a negative answer yielding a false positive is approximately exp(k ln(1 exp(kn/m))).[4] While extensive study of this expression may not be rewarding in the short term, some simple analysis shows that this rate is minimized by picking k = (m/n) ln 2. This corresponds to the case when about half the bits in the filter are 1 and half are 0.

For this to work, the number of bits in the filter should be some multiple of the number of items that you expect to store in it. A common setting is m = 10n and k = 7, which means a false positive rate below 1%. Note that there is no magic here that can compress data beyond information-theoretical limits: under these parameters, the Bloom filter uses about 10 bits per item and must use space proportional to the number of different items stored. This is a modest savings when representing integer values but is a considerable benefit when the items stored have large descriptions—say, arbitrary strings such as URLs. Storing these in a traditional structure such as a hash table or balanced search tree would consume tens or hundreds of bytes per item. A simple example is shown in Figure 1, where an item i is mapped by k = 3 hash functions to a filter of size m = 12, and these entries are set to 1.

Discussion and applications. The possibility of false positives needs to be handled carefully. Bloom filters are at their most attractive when the consequence of a false positive is not the introduction of an error in a computation, but rather when it causes some additional work that does not adversely impact the overall performance of the system. A good example comes in the context of browsing the Web. It is now common for Web browsers to warn users if they are attempting to visit a site that is known to host malware. Checking the URL against a database of "bad" URLs does this. The database is large enough, and URLs are long enough, that keeping the full database, as part of the browser would be unwieldy, especially on mobile devices.

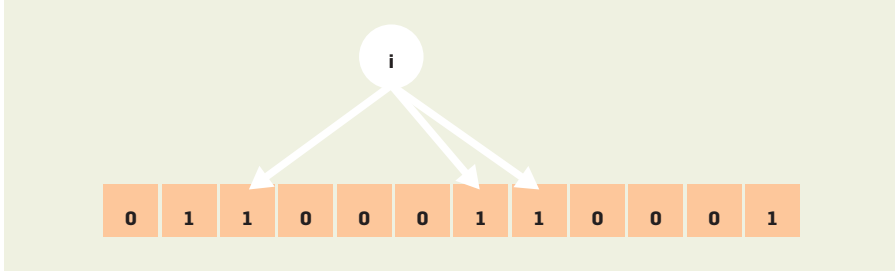
Instead, a Bloom filter encoding of the database can be included with the browser, and each URL visited can be checked against it. The consequence of a false positive is that the browser may believe that an innocent site is on the bad list. To handle this, the browser can contact the database authority and check whether the full URL is on the list. Hence, false positives are removed at the cost of a remote database lookup.

Notice the effect of the Bloom filter: it gives the all clear to most URLs and incurs a slight delay for a small fraction (or when a bad URL is visited). This is preferable both to the solution of keeping a copy of the database with the browser and to doing a remote lookup for every URL visited. Browsers such as Chrome and Firefox have adopted this concept. Current versions of Chrome use a variation of the Bloom filter based on more directly encoding a list of hashed URLs, since the local copy does not have to be updated dynamically and more space can be saved this way.

The Bloom filter was introduced in 1970 as a compact way of storing a dictionary, when space was really at a premium.[3] As computer memory grew, it seemed that the filter was no longer needed. With the rapid growth of the Web, however, a host of applications for the filter have been devised since around the turn of the century.[4] Many of these applications have the flavor of the preceding example: the filter gives a fast answer to lookup queries, and positive answers may be double-checked in an authoritative reference.

Bloom filters have been widely used to avoid storing unpopular items in caches. This enforces the rule that an item is added to the cache only if it has

Figure 1. Bloom filter with K=3, M=12.



been seen before. The Bloom filter is used to compactly represent the set of items that have been seen. The consequence of a false positive is that a small fraction of rare items might also be stored in the cache, contradicting the letter of the rule. Many large distributed databases (Google's Bigtable, Apache's Cassandra and HBase) use Bloom filters as indexes on distributed chunks of data. They use the filter to keep track of which rows or columns of the database are stored on disk, thus avoiding a (costly) disk access for nonexistent attributes.

Counting with Count-Min Sketch

Perhaps the canonical data summarization problem is the most trivial: to count the number of items of a certain type that have been observed, you do not need to retain each item. Instead, a simple counter suffices, incremented with each observation. The counter has to be of sufficient bit depth in order to cope with the magnitude of events observed. When the number of events gets truly huge, ideas such as Robert Morris's approximate counter can be used to provide such a counter in fewer bits[12] (another example of a sketch).

When there are different types of items, and you want to count each type, the natural approach is to allocate a counter for each item. When the number of item types grows huge, however, you encounter difficulties. It may not be practical to allocate a counter for each item type. Even if it is, when the number of counters exceeds the capacity of fast memory, the time cost of incrementing the relevant counter may become too high. For example, a social network such as Twitter may wish to track how often a tweet is viewed when displayed via an external website. There are billions of Web pages, each of which could in principle link to one or more tweets, so allocating counters for each is infeasible and unnecessary. Instead, it is natural to look for a more compact way to encode counts of items, possibly with some tolerable loss of fidelity.

The Count-Min sketch is a data structure that allows this trade-off to be made. It encodes a potentially massive number of item types in a small array. The guarantee is that large counts will be preserved fairly accurately, while small counts may incur greater (relative) error. This means it is good for applications where you are interested in the head of a distribution and less so in its tail.

The method. At first glance, the sketch looks quite like a Bloom filter, as it involves the use of an array and a set of hash functions. There are significant differences in the details, however. The sketch is formed by an array of counters and a set of hash functions that map items into the array. More precisely, the array is treated as a sequence of rows, and each item is mapped by the first hash function into the first row, by the second hash function into the second row, and so on (note that this is in contrast to the Bloom filter, which allows the hash functions to map onto overlapping ranges). An item is processed by mapping it to each row in turn via the corresponding hash function and incrementing the counters to which it is mapped.

Given an item, the sketch allows its count to be estimated. This follows a similar outline to processing an update: inspect the counter in the first row where the item was mapped by the first hash function, and the counter in the second row where it was mapped by the second hash, and so on. Each row has a counter that has been incremented by every occurrence of the item. The counter was also potentially incremented by occurrences of other items that were mapped to the same location, however, since collisions are expected. Given the collection of counters containing the desired count, plus noise, the best guess at the true count of the desired item is to take the smallest of these counters as your estimate.

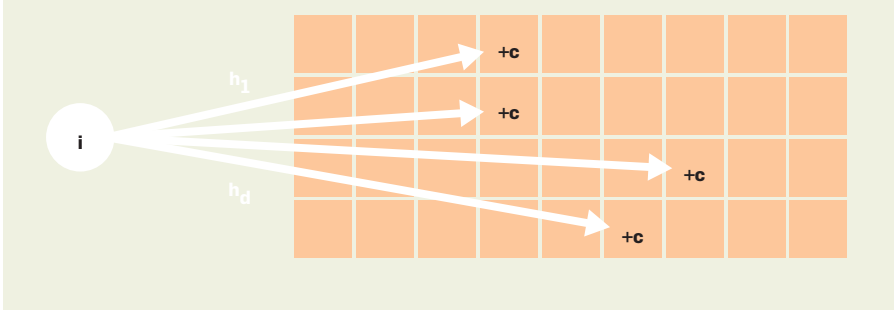
Figure 2 shows the update process: an item `i` is mapped to one entry in each row `j` by the hash function h_j, and the update of `c` is added to each entry. It can also be seen as modeling the query process: a query for the same item `i` will result in the same set of locations being probed, and the smallest value returned as the answer.

Discussion and applications. As with the Bloom filter, the sketch achieves a compact representation of the input, with a trade-off in accuracy. Both provide some probability of an unsatisfactory answer. With a Bloom filter, the answers are binary, so there is some chance of a false positive response; with a Count-Min sketch, the answers are frequencies, so there is some chance of an inflated answer.

What may be surprising at first is that the obtained estimate is very good. Mathematically, it can be shown that there is a good chance that the returned estimate is close to the correct value. The quality of the estimate depends on the number of rows in the sketch (each additional row halves the probability of a bad estimate) and on the number of columns (doubling the number of columns halves the scale of the noise in the estimate). These guarantees follow from the random selection of hash functions and do not rely on any structure or pattern in the data distribution that is being summarized. For a sketch of size `s`, the error is proportional to $1/s$. This is an improvement over the case for sampling where, as noted earlier, the corresponding behavior is proportional to $1/\sqrt{s}$.

Just as Bloom filters are best suited for the cases where false positives can be tolerated and mitigated, Count-Min sketches are best suited for handling a slight inflation of frequency. This means, in particular, they do not apply to cases where a Bloom filter might be used: if it matters a lot whether an item has been seen or not, then the uncertainty that the Count-Min sketch

Figure 2. Count-min sketch data structure with four rows, nine columns.



introduces will obscure this level of precision. The sketches are very good for tracking which items exceed a given popularity threshold, however. In particular, while the size of a Bloom filter must remain proportional to the size of the input it is representing, a Count-Min sketch can be much more compressive: its size can be considered to be independent of the input size, depending instead on the desired accuracy guarantee only (that is, to achieve a target accuracy of ε, fix a sketch size of s proportional to $1/\varepsilon$ that does not vary over the course of processing data).

The Twitter scenario mentioned previously is a good example. Tracking the number of views that a tweet receives across each occurrence in different websites creates a large enough volume of data to be difficult to manage. Moreover, the existence of some uncertainty in this application seems acceptable: the consequences of inflating the popularity of one website for one tweet are minimal. Using a sketch for each tweet consumes only moderately more space than the tweet and associated metadata, and allows tracking which venues attract the most attention for the tweet. Hence, a kilobyte or so of space is sufficient to track the percentage of views from different locations, with an error of less than one percentage point, say.

Since their introduction over a decade ago,[7] Count-Min sketches have found applications in systems that track frequency statistics, such as popularity of content within different groups—say, online videos among different sets of users, or which destinations are popular for nodes within a communications network. Sketches are used in telecommunications networks where the volume of data passing along links is immense and is never stored. Summarizing network traffic distribution allows hotspots to be detected, informing network-planning decisions and allowing configuration errors and floods to be detected and debugged.[6] Since the sketch compactly encodes a frequency distribution, it can also be used to detect when a shift in popularities occurs, as a simple example of anomaly detection.

Successful applications of sketching involve a mixture of algorithmic tricks, systems know-how, and mathematical insight, and have led to new research contributions in each of these areas.

Counting Distinct Items with HyperLogLog

Another basic problem is keeping track of how many different items have been seen out of a large set of possibilities. For example, a Web publisher might want to track how many different people have been exposed to a particular advertisement. In this case, you would not want to count the same viewer more than once. When the number of possible items is not too large, keeping a list, or a binary array, is a natural solution. As the number of possible items becomes very large, the space needed by these methods grows proportional to the number of items tracked. Switching to an approximate method such as a Bloom filter means the space remains proportional to the number of distinct items, although the constants are improved.

Could you hope to do better? If you just counted the total number of items, without removing duplicates, then a simple counter would suffice, using a number of bits that is proportional to the logarithm of the number of items encountered. If only there were a way to know which items were new, and count only those, then you could achieve this cost.

The HyperLogLog (HLL) algorithm promises something even stronger: the cost needs to depend only on the logarithm of the logarithm of the quantity computed. Of course, there are some scaling constants that mean the space needed is not quite so tiny as this might suggest, but the net result is that quantities can be estimated with high precision (say, up to a 1%–2% error) with a couple of kilobytes of space.

The method. The essence of this method is to use hash functions applied to item identifiers to determine how to update counters so that duplicate items are treated identically. A Bloom filter has a similar property: attempting to insert an item already represented within a Bloom filter means setting a number of bits to 1 that are already recording 1 values. One approach is to keep a Bloom filter and look at the final density of 1s and 0s to estimate the number of distinct items represented (taking into account collisions under hash functions). This still requires space proportional to the number of items and is the basis of early approaches to this problem.[15]

To break this linearity, a different approach to building a binary counter is needed. Instead of adding 1 to

the counter for each item, you could add 1 with a probability of one-half, 2 with a probability of one-fourth, 4 with a probability of 1/8th, and so on. This use of randomness decreases the reliability of the counter, but you can check that the expected count corresponds to the true number of items encountered. This makes more sense when using hash functions. Apply a hash function g to each item i, with the same distribution: g maps items to j with probability 2^{-j} (say, by taking the number of leading zero bits in the binary expansion of a uniform hash value). You can then keep a set of bits indicating which j values have been seen so far. This is the essence of the early Flajolet-Martin approach to tracking the number of distinct items.[8] Here a logarithmic number of bits is needed, as there are only this many distinct j values expected.

The HLL method reduces the number of bits further by retaining only the highest j value that has been seen when applying the hash function. This might be expected to be correlated to the cardinality, although with high variation for example, there might be only a single item seen, which happens to hash to a large value. To reduce this variation, the items are partitioned into groups using a second hash function (so the same item is always placed in the same group), and information about the largest hash in each group is retained. Each group yields an estimate of the local cardinality; these are all combined to obtain an estimate of the total cardinality.

A first effort would be to take the mean of the estimates, but this still allows one large estimate to skew the result; instead, the harmonic mean is used to reduce this effect. By hashing to s separate groups, the standard error is proportional to $1/\sqrt{s}$. A small example is shown in Figure 3. The figure shows a small example HLL sketch with s = 3 groups. Consider five distinct items a, b, c, d, e with their related hash values. From this, the following array is obtained:

3	2	1

The estimate is obtained by taking 2 to the power of each of the array entries and computing the sum of the reciprocals of these values, obtaining 1/8 + 1/4 + 1/2 = 7/8 in this case. The final estimate is made by multiplying $\alpha_s s^2$ by the reciprocal of this sum. Here, α_s is a scaling constant that depends on s. $\alpha_3 = 0.5305$, so 5.46 is obtained as the estimate—close to the true value of 5.

The analysis of the algorithm is rather technical, but the proof is in the deployment: the algorithm has been widely adopted and applied in practice.

Discussion and applications. One example of HLL's use is in tracking the viewership of online advertising. Across many websites and different advertisements, trillions of view events may occur every day. Advertisers are interested in the number of "uniques:" how many different people (or rather, browsing devices) have been exposed to the content. Collecting and marshaling this data is not infeasible, but rather unwieldy, especially if it is desired to do more advanced queries (say, to count how many uniques saw both of two particular advertisements). Use of HLL sketches allows this kind of query to be answered directly by combining the two sketches rather than trawling through the full data. Sketches have been put to use for this purpose, where the small amount of uncertainty from the use of randomness is comparable to other sources of error, such as dropped data or measurement failure.

Approximate distinct counting is also widely used behind the scenes in Web-scale systems. For example, Google's Sawzall system provides a variety of sketches, including count distinct, as primitives for log data analysis.[13] Google engineers have described some of the implementation modifications made to ensure high accuracy of the HLL across the whole range of possible cardinalities.[10]

Figure 3. Example of HyperLogLog in action.

x	a	b	c	d	e
h(x)	1	2	3	1	3
g(x)	0001	0011	1010	1101	0101

A last interesting application of distinct counting is in the context of social network analysis. In 2016, Facebook set out to test the "six degrees of separation" claim within its social network. The Facebook friendship graph is sufficiently large (more than a billion nodes and hundreds of billions of edges) that maintaining detailed information about the distribution of long-range connections for each user would be infeasible. Essentially, the problem is to count, for each user, how many friends they have at distance 1, 2, 3, and so on. This would be a simple graph exploration problem, except that some friends at distance 2 are reachable by multiple paths (via different mutual friends). Hence, distinct counting is used to generate accurate statistics on reachability without double counting and to provide accurate distance distributions (the estimated number of degrees of separation in the Facebook graph is 3.57).[2]

Advanced Sketching

Roughly speaking, the four examples of sketching described in this article cover most of the current practical applications of this model of data summarization. Yet, unsurprisingly, there is a large body of research into new applications and variations of these ideas. Just around the corner are a host of new techniques for data summarization that are on the cusp of practicality. This section mentions a few of the directions that seem most promising.

Sketching for dimensionality reduction. When dealing with large high-dimensional numerical data, it is common to seek to reduce the dimensionality while preserving fidelity of the data. Assume the hard work of data wrangling and modeling is done and the data can be modeled as a massive matrix, where each row is one example point, and each column encodes an attribute of the data. A common technique is to apply PCA (principal components analysis) to extract a small number of "directions" from the data. Projecting each row of data along each of these directions yields a different representation of the data that captures most of the variation of the dataset.

One limitation of PCA is that finding the direction entails a substantial amount of work. It requires finding eigenvectors of the covariance matrix,

which rapidly becomes unsustainable for large matrices. The competing approach of random projections argues that rather than finding "the best" directions, it suffices to use (a slightly larger number of) random vectors. Picking a moderate number of random directions captures a comparable amount of variation, while requiring much less computation.

The random projection of each row of the data matrix can be seen as an example of a sketch of the data. More directly, close connections exist between random projections and the sketches described earlier. The Count-Min sketch can be viewed as a random projection of sorts; moreover, the best constructions of random projections for dimensionality reduction look a lot like Count-Min sketches with some twists (such as randomly multiplying each column of the matrix by either -1 or 1). This is the basis of methods for speeding up high-dimensional machine learning, such as the Hash Kernels approach.[14]

Randomized numerical linear algebra. A grand objective for sketching is to allow arbitrary complex mathematical operations over large volumes of data to be answered approximately and quickly via sketches. While this objective appears quite a long way off, and perhaps infeasible because of some impossibility results, a number of core mathematical operations can be solved using sketching ideas, which leads to the notion of randomized numerical linear algebra. A simple example is matrix multiplication: given two large matrices `A` and `B`, you want to find their product `AB`. An approach using sketching is to build a dimensionality-reducing sketch of each row of `A` and each column of `B`. Combining each pair of these provides an estimate for each entry of the product. Similar to other examples, small answers are not well preserved, but large entries are accurately found.

Other problems that have been tackled in this space include regression. Here the input is a high-dimensional dataset modeled as matrix `A` and column vector `b`: each row of `A` is a data point, with the corresponding entry of `b` the value associated with the row. The goal is to find regression coefficients `x` that minimize $||Ax-b||_2$. An exact solution to this problem is possible but costly in terms of time as a function of the number of rows. Instead, applying sketching to matrix `A` solves the problem in the lower-dimensional sketch space.[5] David Woodruff provides a comprehensive mathematical survey of the state of the art in this area.[16]

Rich data: Graphs and geometry. The applications of sketching so far can be seen as summarizing data that might be thought of as a high-dimensional vector, or matrix. These mathematical abstractions capture a large number of situations, but, increasingly, a richer model of data is desired—say, to model links in a social network (best thought of as a graph) or to measure movement patterns of mobile users (best thought of as points in the plane or in 3D). Sketching ideas have been applied here also.

For graphs, there are techniques to summarize the adjacency information of each node, so that connectivity and spanning tree information can be extracted.[1] These methods provide a surprising mathematical insight that much edge data can be compressed while preserving fundamental information about the graph structure. These techniques have not found significant use in practice yet, perhaps because of high overheads in the encoding size.

For geometric data, there has been much interest in solving problems such as clustering.[9] The key idea here is that clustering part of the input can capture a lot of the overall structural information, and by merging clusters together (clustering clusters) you can retain a good picture of the overall point density distribution.

Why Should You Care?

The aim of this article has been to introduce a selection of recent techniques that provide approximate answers to some general questions that often occur in data analysis and manipulation. In all cases, simple alternative approaches can provide exact answers, at the expense of keeping complete information. The examples shown here have illustrated, however, that in many cases the approximate approach can be faster and more space efficient. The use of these methods is growing. Bloom filters are sometimes said to be one of the core technologies that "big data experts" must know. At the very least, it is important to be aware of sketching techniques to test claims that solving a problem a certain way is the only option. Often, fast approximate sketch-based techniques can provide a different trade-off. C

Related articles on queue.acm.org

It Probably Works
Tyler McMullen
http://queue.acm.org/detail.cfm?id=2855183

Statistics for Engineers
Heinrich Hartmann
http://queue.acm.org/detail.cfm?id=2903468

References

1. Ahn, K.J., Guha, S., McGregor, A. Analyzing graph structure via linear measurements. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, (2012).
2. Bhagat, S., Burke, M., Diuk, C., Filiz, I.O., Edunov, S. Three-and-a-half degrees of separation. Facebook Research, 2016; https://research.fb.com/three-and-a-half-degrees-of-separation/.
3. Bloom, B. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM 13*, 7 (July 1970), 422–426.
4. Broder, M., Mitzenmacher, A. Network applications of Bloom filters: a survey. *Internet Mathematics 1*, 4 (2005), 485–509.
5. Clarkson, K.L., Woodruff, D.P. Low rank approximation and regression in input sparsity time. In *Proceedings of the ACM Symposium on Theory of Computing*, (2013), 81–90.
6. Cormode, G., Korn, F., Muthukrishnan, S., Johnson, T., Spatscheck, O., Srivastava, D. 2004. Holistic UDAFs at streaming speeds. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, (2004), 35–46.
7. Cormode, G., Muthukrishnan, S. An improved data stream summary: the Count-Min sketch and its applications. *J. Algorithms 55*, 1 (2005), 58–75.
8. Flajolet, P., Martin, G.N. 1985. Probabilistic counting. In *Proceedings of the IEEE Conference on Foundations of Computer Science*, 1985, 76–82. Also in *J. Computer and System Sciences 31*, 182–209.
9. Guha, S., Mishra, N., Motwani, R., O'Callaghan, L. Clustering data streams. In *Proceedings of the IEEE Conference on Foundations of Computer Science*, 2000.
10. Heule, S., Nunkesser, M., Hall, A. HyperLogLog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm. In *Proceedings of the International Conference on Extending Database Technology*, 2013.
11. Jermaine, C. Sampling techniques for massive data. Synopses for massive data: samples, histograms, wavelets and sketches. *Foundations and Trends in Databases 4*, 1–3 (2012). G. Cormode, M. Garofalakis, P. Haas, and C. Jermaine, Eds. NOW Publishers.
12. Morris, R. Counting large numbers of events in small registers. *Commun. ACM 21*, 10 (Oct. 1977), 840–842.
13. Pike, R., Dorward, S., Griesemer, R., Quinlan, S. Interpreting the data: Parallel analysis with Sawzall. *Dynamic Grids and Worldwide Computing 13*, 4 (2005), 277–298.
14. Weinberger, K.Q., Dasgupta, A., Langford, J., Smola, A.J., Attenberg, J. Feature hashing for large-scale multitask learning. In *Proceedings of the International Conference on Machine Learning*, 2009.
15. Whang, K.Y., Vander-Zanden, B.T., Taylor, H.M. A linear-time probabilistic counting algorithm for database applications. *ACM Trans. Database Systems 15*, 2 (1990, 208.
16. Woodruff, D. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science 10*, 1–2 (2014), 1–157.

Graham Cormode is a professor of computer science at the University of Warwick, U.K. Previously, he was a researcher at Bell Labs and AT&T on algorithms for data management. He received the 2017 Adams Prize for his work on data analysis.

DOI:10.1145/3106631

Article development led by acmqueue
queue.acm.org

Plan ahead to make the interview a successful one.

BY KATE MATSUDAIRA

10 Ways to Be a Better Interviewer

IN MANY WAYS interviewing is an art. You have one hour (more if you count the cumulative interview time) to determine if the candidate has the desired skills, and, more importantly, if you would enjoy working with this person. That is a lot of ground to cover.

As if finding out all that information isn't a daunting enough task, you also need to make sure that the candidate has a positive experience while visiting your company (after all, people talk and you want them to be saying good things—since this candidate may not be your next hire, but someone he or she meets may be).

As an interviewer, the key to your success is preparation. Planning will help ensure the success of the interview (both in terms of getting the information you need and giving the candidate a good impression).

The following list is advice to consider prior to stepping into that room with two chairs and a whiteboard.

1. Review the Candidate's Résumé

Read every line of every résumé (and this goes for the really long ones that go on for four pages). Where have these candidates worked? How long did they stay in a role and did their positions change? These questions make for interesting conversation topics. Hopefully there will be something in a candidate's background that piques your interest and can be great fodder for starting the interview with some common ground. This can put candidates at ease, giving them their greatest chance of success.

2. Review Feedback from Previous Interviews

Most software companies have a longer interview process that can start with phone-screen or homework problems and evolve from there. If the candidate has done homework problems, or your teammates have taken the time to type up feedback, do your due diligence and read it. These can also be a great source of material for questions, but more importantly, it is unprofessional to ask the same questions that have already been posed to the candidate. This is partly because you will not learn as much from repeated questions, but also because the candidate will be bored or unimpressed going over the same ground. Great candidates want to be challenged, and an interview team where people are asking the same questions makes the candidate think the team is disorganized or unimaginative.

3. Use Calibrated Questions

Interviews are not the time to try something new. Take the time to do new problems on your own or test them on your peers. Come to the interview with questions that you were given in your interview (since you certainly will know how well you did) or that you have already given to others. Testing new material can really hurt a candidate's chances for success or, worse, give him or her a bad impression of the company when you

PHOTO BY DRAGON IMAGES

are not prepared to answer clarifying questions. You get the most from interviews when you can compare the results of one with another, particularly with the results of a successful hire or peer—so try to come to the interview with questions that will help you make this comparison.

4. Test New Questions on Yourself and Your Peers

If you do have a new question you want to give a dry run, have someone ask you to answer it. Where do you get hung up? How long does it take you? If the problem is too familiar to you to assess it, ask one of your teammates to be your guinea pig (as a manager I often offer to be the interviewee for my team to test out new questions; after all, isn't it fun to turn the tables and interview your manager?). Seeing where the people you know and respect get stuck, or how long they take to solve it, will give you a good baseline for comparison with future candidates.

5. Create a Timeline for the Interview

You should walk into every interview with a schedule: what questions you plan to ask and how long each should take. Each question should have clear goals and focus on specific competencies for the position. Ideally, the questions should be different from one another and give you a feel for multiple areas of the candidate's experience and background. I like to ask about five questions, so a typical agenda might look like this:

- Warm-up question about the candidate's background (or common interest): 5–10 minutes
- Problem-solving question that involves coding of some sort: 10–20 minutes

- Design question: 10–15 minutes
- Two to three cultural or situational questions: 5–10 minutes
- Time to answer the candidate's questions

6. Head In With a Positive Attitude

You want the candidate to have a good experience with the company and your process. If you are upbeat, it is much more likely a qualified candidate will accept the position. If you are not, people talk and it is a small world. You want candidates to think well of the company and feel they were treated fairly. It's like karma—what goes around comes around. To ensure this happens, try to make your questions and hints feel collaborative, and whatever you do, do not insult any candidates or make them feel stupid. They are probably nervous and you already have the job—there is nothing to prove, so make an effort to give them a fair shot.

7. Take Notes

Seems obvious, but so many people don't take notes. Even if you have a photographic memory, taking the time to write down a few things here and there will indicate to the candidate you are paying attention and are genuinely interested in what he or she has to say. As an avid note-taker, here are some of my favorite tips:

- Try not to use a laptop. Yes, it is probably faster and more efficient, but it can be a physical divider between you and the candidate, not to mention off-putting. When an interviewer uses a computer during an interview, it is easy to think that he or she is not paying attention to what the candidate has to say.
- Instead of writing code/drawings on a whiteboard, try paper. This may be more comfortable for most people than standing up at a whiteboard, and you can take the paper with you, which is better than any copied whiteboard code.
- Don't write notes on the résumé. Someone once told me that in some cultures, business cards and résumés are considered a reflection of the person, and writing on them can be insulting. While I personally haven't encountered anyone who felt this way, I am sure never to do this (and bring my own paper) just in case.

Don't write notes on the résumé. Someone once told me that in some cultures, business cards and résumés are considered a reflection of the person, and writing on them can be insulting.

8. Bring a List of Questions to the Interview

No candidate will think less of you for coming in with written questions, and in fact some may appreciate that you prepared the same way they did. This will also help you establish your game plan and agenda so you don't forget. Another one of my favorite tips is always to have spare questions for really good interviews (that get through all the material quickly) or for bad interviews (where you don't want to ask your prepared questions because they are too hard).

9. Be Collaborative

You want the candidate to be successful, so try to approach a problem together. I know many other managers who have moved to a pair-programming model where the interviewer and the candidate code a problem together in an editor or Google doc.

10. Try To Make the Problems Feel As Real-World As Possible

Smart people want to be challenged. They also would love to get a taste of what it is like to work at your company. Do your best to come up with questions that at least hint at some of the problems you might solve (or problems that relate to the underlying theory of the work you do).

Of course, there is no right way to do an interview, but you can always be better. Make an effort to make your candidates as comfortable as possible so they have the greatest chance for success. Happy hiring!

Related articles on queue.acm.org

Interviewing Techniques
George Neville-Neil
http://queue.acm.org/detail.cfm?id=1998475

Nine Things I Didn't Know I Would Learn Being an Engineer Manager
Kate Matsudaira
http://queue.acm.org/detail.cfm?id=2935693

10 Optimizations on Linear Search
Thomas A. Limoncelli
http://queue.acm.org/detail.cfm?id=2984631

Kate Matsudaira (katemats.com) is the founder of her own company, Popforms. Previously she worked at Microsoft and Amazon as well as startups like Decide, Moz, and Delve Networks.

ACM Europe Conference

Barcelona, Spain | 7 – 8 September 2017

The ACM Europe Conference, hosted in Barcelona by the Barcelona Supercomputing Center, aims to bring together computer scientists and practitioners interested in exascale high performance computing and cybersecurity.

The High Performance Computing track includes a panel discussion of top world experts in HPC to review progress and current plans for the worldwide roadmap toward exascale computing. The Cybersecurity track will review the latest trends in this very hot field. High-level European Commission officials and representatives of funding agencies are participating.

Keynote Talk by ACM 2012 Turing Award Laureate Silvio Micali, "ALGORAND: A New Distributed Ledger"

Co-located events:

- ACM Europe Celebration of Women in Computing: WomENcourage 2017 (Requires registration, https://womencourage.acm.org/)
- EXCDI, the European Extreme Data & Computing Initiative (https://exdci.eu/)
- Eurolab-4-HPC (https://www.eurolab4hpc.eu/)
- HiPEAC, the European Network on High Performance and Embedded Architecture and Compilation (https://www.hipeac.net/)

Conference Chair: Mateo Valero, Director of the Barcelona Supercomputing Center

Registration to the ACM Europe Conference is free of charge for ACM members and attendees of the co-located events.

acm Europe Council

http://acmeurope-conference.acm.org

DOI:10.1145/3122814

Answering questions correctly from standardized eighth-grade science tests is itself a test of machine intelligence.

BY CARISSA SCHOENICK, PETER CLARK, OYVIND TAFJORD, PETER TURNEY, AND OREN ETZIONI

Moving Beyond the Turing Test with the Allen AI Science Challenge

THE FIELD OF artificial intelligence has made great strides recently, as in AlphaGo's victories in the game of Go over world champion South Korean Lee Sedol in March 2016 and top-ranked Chinese Go player Ke Jie in May 2017, leading to great optimism for the field. But are we really moving toward smarter machines, or are these successes restricted to certain classes of problems, leaving others untouched? In 2015, the Allen Institute for Artificial Intelligence (AI2) ran its first Allen AI Science Challenge, a competition to test machines on an ostensibly difficult task—answering eighth-grade science questions. Our motivations were to encourage the field to set its sights more broadly by exploring a problem that appears to require modeling,

» key insights

- Determining whether a system truly displays artificial intelligence is difficult and complex, and well-known assessments like the Turing Test are not suited to the task.
- The Allen Institute for Artificial Intelligence suggests that answering science exam questions successfully is a better measure of machine intelligence and designed a global competition to engage the research community in this approach.
- The outcome of the Allen AI Science Challenge highlights the current limitations of AI research in language understanding, reasoning, and commonsense knowledge; the highest scores are still limited to the capabilities of information-retrieval methods.

PHOTO BY PANITAN PHOTO. WITH ROBOT ILLUSTRATION BY PETER CROWTHER ASSOCIATES

reasoning, language understanding, and commonsense knowledge in order to probe the state of the art while sowing the seeds for possible future breakthroughs.

Challenge problems have historically played an important role in motivating and driving progress in research. For a field striving to endow machines with intelligent behavior (such as language understanding and reasoning), challenge problems that test such skills are essential.

In 1950, Alan Turing proposed the now well-known Turing Test as a possible test of machine intelligence: If a system can exhibit conversational behavior that is indistinguishable from that of a human during a conversation, that system could be considered intelligent.[11] As the field of AI has grown, the test has become less meaningful as a challenge task for several reasons. First, in its details, it is not well defined (such as Who is the person giving the test?). A computer scientist would likely know good distinguishing questions to ask, while a random member of the general public may not. What constraints are there on the interaction? What guidelines are provided to the judges? Second, recent Turing Test competitions have shown that, in certain formulations, the test itself is gameable; that is, people can be fooled by systems that simply retrieve sentences and make no claim of being intelligent.[2,3] John Markoff of *The New York Times* wrote that the Turing Test is more a test of human gullibility than machine intelligence. Finally, the test as originally conceived is pass/fail rather than scored, thus providing no measure of progress toward a goal, something essential for any challenge problem.[a,b]

Machine intelligence today is viewed less as a binary pass/fail attribute and

a Turing himself did not conceive of the Turing Test as a challenge problem to drive the field forward but rather as a thought experiment to explore a useful alternative to the question Can machines think?

b Although one can imagine metrics that quantify performance on the Turing Test, the imprecision of the task definition and human variability make it difficult to define metrics that are reliably reproducible.

more as a diverse collection of capabilities associated with intelligent behavior. Rather than a single test, cognitive scientist Gary Marcus of New York University and others have proposed the notion of series of tests—a Turing Olympics of sorts—that could assess the full gamut of AI, from robotics to natural language processing.[9,12]

Our goal with the Allen AI Science Challenge was to operationalize one such test—answering science-exam questions. Clearly, the Science Challenge is not a full test of machine intelligence but does explore several capabilities strongly associated with intelligence—capabilities our machines need if they are to reliably perform the smart activities we desire of them in the future, including language understanding, reasoning, and use of commonsense knowledge. Doing well on the challenge appears to require significant advances in AI technology, making it a potentially powerful way to advance the field. Moreover, from a practical point of view, exams are accessible, measurable, understandable, and compelling.

One of the most interesting and appealing aspects of science exams is their graduated and multifaceted nature; different questions explore different types of knowledge, varying substantially in difficulty, especially for a computer. There are questions that are easily addressed with a simple fact lookup, like this

How many chromosomes does the human body cell contain?
(A) 23
(B) 32
(C) 46
(D) 64

Then there are questions requiring extensive understanding of the world, like this

City administrators can encourage energy conservation by
(A) lowering parking fees
(B) building larger parking lots
(C) decreasing the cost of gasoline
(D) lowering the cost of bus and subway fares

This question requires the knowledge that certain activities and incentives result in human behaviors that in turn result in more or less energy being consumed. Understanding the question also requires the system being able to recognize that "energy" in this context refers to resource consumption for the purposes of transportation, as opposed to other forms of energy one might find in a science exam (such as electrical and kinetic/potential).

AI vs. Eighth Grade

To put this approach to the test, AI2 designed and hosted The Allen AI Science Challenge, a four-month-long competition in partnership with Kaggle (https://www.kaggle.com/) that began in October 2015 and concluded in February 2016.[7] Researchers worldwide were invited to build AI software that could answer standard eighth-grade multiple-choice science questions. The competition aimed to assess the state of the art in AI systems utilizing natural language understanding and knowledge-based reasoning; how accurately the participants' models could answer the exam questions would serve as an indicator of how far the field has come in these areas.

Participants. A total of 780 teams participated during the model-building phase, with 170 of them eventually submitting a final model. Participants were required to make the code for their models available to AI2 at the close of the competition to validate model performance and confirm they followed contest rules. At the conclusion of the competition, the winners were also expected to make their code open source. The three teams achieving the highest scores on the challenge's test set received prizes of $50,000, $20,000, and $10,000, respectively.

Data. AI2 licensed a total of 5,083 eighth-grade multiple-choice science questions from providing partners for the purposes of the competition. All questions were standard multiple-choice format, with four answer options, as in the earlier examples. From this collection, we provided participants with a set of 2,500 training questions to train their models. We used a validation set of 8,132 questions during the course of the competition for confirming model performance. Only 800 of the validation questions were legitimate; we artificially generated the rest to disguise the real questions in order to prevent cheating via manual question answering or unfair advantage of additional training examples. A week before the end of the competition, we provided the final test set of 21,298 questions (including the validation set) to participants to use to produce a final score for their models, of which 2,583 were legitimate. We licensed the data for the competition from private assessment-content providers that did not wish to allow the use of their data beyond the constraints of the competition, though AI2 made some subsets of the questions available on its website http://allenai.org/data.

Baselines and scores. As these questions were all four-way multiple choice, a standard baseline score using random guessing was 25%. AI2 also generated a baseline score using a Lucene search over the Wikipedia corpus, producing scores of 40.2% on the training set and 40.7% on the final test set. The final results of the competition was quite close, with the top three teams achieving scores with a spread of only 1.05%. The highest score was 59.31%.

First Place

Top prize went to Chaim Linhart of Hod HaSharon, Israel (Kaggle data science website https://www.kaggle.com username Cardal). His model achieved a final score of 59.31% correct on the test question set of 2,583 questions using a combination of 15 gradient-boosting models, each with a different subset of features. Unlike the other winners' models, Linhart's model predicted the correctness of each answer option individually. Linhart used two general categories of features to make these predictions; the first consisted of information-retrieval-based features, applied by searching over corpora he compiled from various sources (such as study-guide or quiz-building websites, open source textbooks, and Wikipedia). His searches used various weightings and stemmed words to optimize performance. The other flavor of features used in his ensemble of 15 models was based on properties of the questions themselves (such as length of question and answer, form of answer like numeric answer options, answers containing referential clauses like "none of the above" as an option, and relationships among answer options).

Linhart explained that he used several smaller gradient-boosting models instead of one big model to maximize diversity. One big model tends to ignore some important features because it requires a very large training set to ensure it pays attention to all potentially useful features present. Linhart's use of several small models required that the learning algorithm use features it would otherwise ignore, an advantage, given the relatively limited training data available in the competition.

The information-retrieval-based features alone could achieve scores as high as 55% by Linhart's estimation. His question-form features filled in some remaining gaps to bring the system up to approximately 60% correct. He combined his 15 models using a simple weighted average to yield the final score for each choice. He credited careful corpus selection as one of the primary elements driving the success of his model.

In the end, each of the winning models gained from information-retrieval-based methods, indicative of the state of AI technology in this area of research.

Second Place

The second-place team, with a score of 58.34%, was from a social-media-analytics company based in Luxembourg called Talkwalker (https://www.talkwalker.com), led by Benedikt Wilbertz (Kaggle username poweredByTalkwalker).

The Talkwalker team built a relatively large corpus compared to other winning models, using 180GB of disk space after indexing with Lucene. Feature types included information-retrieval-based features, vector-based features (scoring question-answer similarity by comparing vectors from word2vec, a two-layer neural net that processes text, and GloVe, an unsupervised learning algorithm (for obtaining vector representations for words), pointwise mutual information features (measured between the question and target answer, calculated on the team's large corpus), and string hashing features in which term-definition pairs were hashed and a supervised learner was then trained to classify pairs as correct or incorrect. A final model used them to learn pairwise ranking between the answer options using the XGBoost library, an implementation of gradient-boosted decision trees.

Wilbertz's use of string hashing features was unique, not tried by either of the other two winners nor currently used in AI2's Project Aristo. His team used a corpus of terms and definitions obtained from an educational-flashcard-building site, then created negative examples by mixing terms with random definitions. A supervised classifier was trained on these incorrect pairs, and the output was used to generate features for input to XGBoost.

Third Place

The third-place winner was Alejandro Mosquera from Reading, U.K. (Kaggle username Alejandro Mosquera), with a score of 58.26%. Mosquera approached the challenge as a three-way classification problem for each pair of answer options. He transformed answer choices A, B, C, and D to all 12 possible pairs (A,B), (A,C), ..., (D,C) he labeled with three classes: left-pair element is correct; right is correct; or neither is correct. He then classified the pairs using logistic regression. This three-way classification is easier for supervised learning algorithms than the more natural two-way (correct vs. incorrect) classification with four choices, because the two-way classification requires an absolute decision about a choice, whereas the three-way classification requires only a relative ranking of the choices. Mosquera made use of three types of features: information-retrieval-based features based on scores from Elastic Search using Lucene over a corpus; vector-based features that measured question-answer similarity by comparing vectors from word2vec; and question-form features that considered such aspects of the data as the structure of a question, length of a question, and answer choices. Mosquera also noted that careful corpus selection was crucial to his model's success.

Lessons

In the end, each of the winning models gained from information-retrieval-based methods, indicative of the state of AI technology in this area of research. AI researchers intent on creating a machine with human-like intelligence are unable to ace an eighth-grade science exam because they do not currently have AI systems able to go beyond surface text to a deeper understanding of the meaning underlying each question, then use reasoning to find the appropriate answer. All three winners said it was clear that applying a deeper, semantic level of reasoning with scientific knowledge to the questions and answers would be the

key to achieving scores of 80% and higher and demonstrating what might be considered true artificial intelligence.

A few other example questions each of the top three models got wrong highlight the more interesting, complex nuances of language and chains of reasoning an AI system must be able to handle in order to answer the following questions correctly and for which information-retrieval methods are not sufficient:

What do earthquakes tell scientists about the history of the planet?

(A) Earth's climate is constantly changing.

(B) The continents of Earth are continually moving.

(C) Dinosaurs became extinct about 65 million years ago.

(D) The oceans are much deeper today than millions of years ago.

This involves the causes behind earthquakes and the larger geographic phenomena of plate tectonics and is not easily solved by looking up a single fact. Additionally, other true facts appear in the answer options ("Dinosaurs became extinct about 65 million years ago.") but must be intentionally identified and discounted as incorrect in the context of the question.

Which statement correctly describes a relationship between the distance from Earth and a characteristic of a star?

(A) As the distance from Earth to the star decreases, its size increases.

(B) As the distance from Earth to the star increases, its size decreases.

(C) As the distance from Earth to the star decreases, its apparent brightness increases.

(D) As the distance from Earth to the star increases, its apparent brightness increases.

This requires general commonsense-type knowledge of the physics of distance and perception, as well as the semantic ability to relate one statement to another within each answer option to find the right directional relationship.

Other Attempts

While numerous question-answering systems have emerged from the AI community, none has addressed the challenges of scientific and commonsense reasoning required to successfully answer these example questions. Question-answering systems developed for the message-understanding conferences[6] and text-retrieval conferences[13] have historically focused on retrieving answers from text, the former from newswire articles, the latter from various large corpora (such as the Web, microblogs, and clinical data). More recent work has focused on answer retrieval from structured data (such as "In which city was Bill Clinton born?" from FreeBase, a large publicly available collaborative knowledgebase).[4,5,15] However, these systems rely on the information being stated explicitly in the underlying data and are unable to perform the reasoning steps that would be required to conclude this information from indirect supporting evidence.

A few systems attempt some form of reasoning: Wolfram Alpha[14] answers mathematical questions, providing they are stated either as equations or with relatively simple English; Evi[10] is able to combine facts to answer simple questions (such as "Who is older: Barack or Michelle Obama?"); and START,[8] which likewise is able to answer simple inference questions (such as "What South American country has the largest population?") using Web-based databases. However, none of them attempts the level of complex question processing and reasoning that is indeed required to successfully answer many of the science questions in the Allen AI Challenge.

Looking Forward

As the 2015 Allen AI Science Challenge demonstrated, achieving a high score on a science exam requires a system that can do more than sophisticated information retrieval. Project Aristo at AI2 is focused on the problem of successfully demonstrating artificial intelligence using standardized science exams, developing an assortment of approaches to address the challenge. AI2 plans to release additional datasets and software for the wider AI research community in this effort.[1]

References

1. Allen Institute for Artificial Intelligence. Datasets; http://allenai.org/data
2. Aron, J. Software tricks people into thinking it is human. *New Scientist 2829* (Sept. 6, 2011).
3. BBC News. Computer AI passes Turing Test in 'world first.' *BBC News* (June 9, 2014); http://www.bbc.com/news/technology-27762088
4. Berant, J., Chou, A., Frostig, R., and Liang, P. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (Seattle, WA, Oct. 18–21). Association for Computational Linguistics, Stroudsburg, PA, 2013, 6.
5. Fader, A., Zettlemoyer, L., and Etzioni, O. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, Aug. 24–27). ACM Press, New York, 2014.
6. Grishman, R. and Sundheim, B. Message understanding Conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics* (Copenhagen, Denmark, Aug. 5–9). Association for Computational Linguistics, Stroudsburg, PA, 1996, 466–471.
7. Kaggle. *The Allen AI Science Challenge*; https://www.kaggle.com/c/the-allen-ai-science-challenge
8. Katz, B., Borchardt, G., and Felshin, S. Natural language annotations for question answering. In *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference* (Melbourne Beach, FL, May 11–13). AAAI Press, Menlo Park, CA, 2006.
9. Marcus, G., Rossi, F., and Veloso, M., Eds. Beyond the Turing Test. *AI Magazine (Special Edition) 37*, 1 (Spring 2016).
10. Simmons, J. True Knowledge: The natural language question answering Wikipedia for facts. *Semantic Focus* (Feb. 26, 2008); http://www.semanticfocus.com/blog/entry/title/true-knowledge-the-natural-language-question-answering-wikipedia-for-facts/
11. Turing, A.M. Computing machinery and intelligence. *Mind 59*, 236 (Oct. 1950), 433–460.
12. Turk, V. The plan to replace the Turing Test with a 'Turing Olympics.' *Motherboard* (Jan. 28, 2015); https://motherboard.vice.com/en_us/article/the-plan-to-replace-the-turing-test-with-a-turing-olympics
13. Voorhees, E. and Ellis, A., Eds. In *Proceedings of the 24th Text REtrieval Conference* (Gaithersburg, MD, Nov. 17–20). Publication SP 500-319, National Institute of Standards and Technology, Gaithersburg, MD, 2015.
14. Wolfram, S. Making the world's data computable. *Stephen Wolfram Blog* (Sept. 24, 2010); http://blog.stephenwolfram.com/2010/09/making-the-worlds-data-computable/
15. Yao, X. and Van Durme, B. Information extraction over structured data: Question answering with Freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Baltimore, MD, June 22–27). Association for Computational Linguistics, Stroudsburg, PA, 2014, 956–966.

Carissa Schoenick (carissas@allenai.org) is the senior program manager for Project Aristo at the Allen Institute for Artificial Intelligence in Seattle, WA.

Peter Clark (peterc@allenai.org) is the senior research manager for Project Aristo at the Allen Institute for Artificial Intelligence in Seattle, WA.

Oyvind Tafjord (oyvindt@allenai.org) is a senior research scientist and engineer at the Allen Institute for Artificial Intelligence in Seattle, WA.

Peter Turney (peter.turney@gmail.com) was a senior research scientist for Project Aristo at the Allen Institute for Artificial Intelligence in Seattle, WA, and is now retired.

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

Watch the authors discuss their work in this exclusive *Communications* video. https://cacm.acm.org/videos/moving-beyond-the-turing-test

DOI:10.1145/3122803

Even when checked by fact checkers, facts are often still open to preexisting bias and doubt.

BY PETTER BAE BRANDTZAEG AND ASBJØRN FØLSTAD

Trust and Distrust in Online Fact-Checking Services

WHILE THE INTERNET has the potential to give people ready access to relevant and factual information, social media sites like Facebook and Twitter have made filtering and assessing online content increasingly difficult due to its rapid flow and enormous volume. In fact, 49% of social media users in the U.S. in 2012 received false breaking news through social media.[8] Likewise, a survey by Silverman[11] suggested in 2015 that false rumors and misinformation disseminated further and faster than ever before due to social media. Political analysts continue to discuss misinformation and fake news in social media and its effect on the 2016 U.S. presidential election.

» key insights

- **Though fact-checking services play an important role countering online disinformation, little is known about whether users actually trust or distrust them.**
- **The data we collected from social media discussions—on Facebook, Twitter, blogs, forums, and discussion threads in online newspapers—reflects users' opinions about fact-checking services.**
- **To strengthen trust, fact-checking services should strive to increase transparency in their processes, as well as in their organizations, and funding sources.**

Such misinformation challenges the credibility of the Internet as a venue for authentic public information and debate. In response, over the past five years, a proliferation of outlets has provided fact checking and debunking of online content. Fact-checking services, say Kriplean et al.,[6] provide "... evaluation of verifiable claims made in public statements through investigation of primary and secondary sources." An international

Figure 1. Categorization of fact-checking services based on areas of concern.

Fact-checking services' areas of concern

Online rumors and hoaxes	Political and public claims	Specific topics or controversies
Snopes.com	FactCheck.org	StopeFake
Hoax-Slayer	PolitiFact	TruthBeTold
ThruthOrFiction.com	The Washington Post Fact Checker	#RefugeeCheck
HoaxBusters	CNN Reality Check	Climate Feedback
Viralgranskaren - Metro	Full Fact	Brown Moses Blog (continued as Bellingcat)

Figure 3. Outline of our research approach; posts collected October 2014 to March 2015.

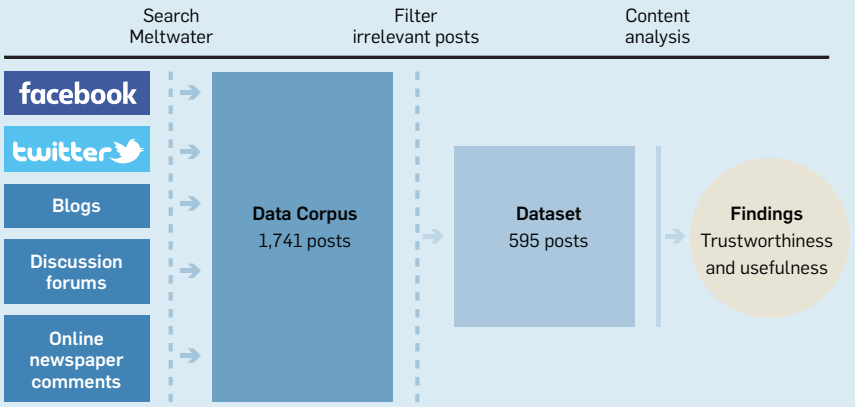


Table 1. Coding scheme we used to analyze the data.

Theme	Sentiment	Service described as
Usefulness	Positive	Useful, serving the purpose of fact checking
	Negative	Not as useful, often derogatory
Ability	Positive	Reputable, expert, or acclaimed
	Negative	Lacking expertise or credibility
Benevolence	Positive	Aiming for greater (social) good
	Negative	Suspected of (social) ill will (such as through conspiracy, propaganda, or fraud)
Integrity	Positive	Independent or impartial
	Negative	Dependent or partially or politically biased

Figure 2. Example of Snopes debunking a social media rumor on Twitter (March 6, 2016); https://twitter.com/snopes/status/706545708233396225

census from 2017 counted 114 active fact-checking services, a 19% increase over the previous year.[12] To benefit from this trend, Google News in 2016 let news providers tag news articles or their content with fact-checking information "... to help readers find fact checking in large news stories."[3] Any organization can use the fact-checking tag, if it is non-partisan, transparent, and targets a range of claims within an area of interest and not just one single person or entity.

However, research into fact checking has scarcely paid attention to the general public's view of fact checking, focusing instead on how people's beliefs and attitudes change in response to facts that contradict their own preexisting opinions. This research suggests fact checking in general may be unsuccessful at reducing misperceptions, especially among the people most prone to believe them.[9] People often ignore facts that contradict their current beliefs,[2,13] particularly in politics and controversial social issues.[9] Consequently, the more political or controversial issues a fact-checking service covers, the more it needs to build a reputation for usefulness and trustworthiness.

Research suggests the trustworthiness of fact-checking services depends on their origin and ownership, which may in turn affect integrity perceptions[10] and the transparency of their fact-checking process.[4] Despite these observations, we are unaware of any other research that has examined users' perceptions of these services. Addressing the gap in current knowledge, we investigated the research question: How do social media users perceive the trustworthiness and usefulness of fact-checking services?

Fact-checking services differ in terms of their organizational aim and funding,[10] as well as their areas of concern,[11] that in turn may affect their trustworthiness. As outlined in Figure 1, the universe of fact-checking services can be divided into three general categories based on their area(s) of concern: political and public statements in general, corresponding to the fact checking of politicians, as discussed by Nyhan and Reifler;[9] online rumors and hoaxes, reflecting the need for debunking services, as discussed by Silverman;[11]

and specific topics or controversies or particular conflicts or narrowly scoped issues or events (such as the ongoing Ukraine conflict).

We have focused on three services—Snopes, FactCheck.org, and StopFake—all included in the Duke Reporters' Lab's online overview of fact checkers (http://reporterslab.org/fact-checking/). They represent three categories of fact checkers, from online rumors to politics to a particular topic, as in Figure 1, and differences in organization and funding. As a measure of their popularity, as of June 20, 2017, Snopes had 561,650 likes on Facebook, FactCheck.org 806,814, and StopFake 52,537.

We study Snopes because of its aim to debunk online rumors, fitting the first category in Figure 1. This aim is shared by other such services, including HoaxBusters and the Swedish service Viralgranskaren. Snopes is managed by a small volunteer organization that has emerged from a single-person initiative and funded through advertising revenue.

We study FactCheck.org because it monitors the factual accuracy of what is said by major political figures. Other such services include PolitiFact (U.S.) and Full Fact (U.K.) in the second category in Figure 1. FactCheck.org is a project of the Annenberg Public Policy Center of the Annenberg School for Communication at the University of Pennsylvania, Philadelphia, PA. FactCheck.org is supported by university funding and individual donors and has been a source of inspiration for other fact-checking projects.

We study StopFake because it addresses one highly specific topic—the ongoing Ukraine conflict. It thus resembles other highly focused fact-checking initiatives (such as #Refugeecheck, which fact checks reports on the refugee crises in Europe). StopFake is an initiative by the Kyiv Mohyla Journalism School in Kiev, Ukraine, and is thus a European-based service. Snopes and FactCheck.org are U.S. based, as are more than a third of the fact-checking services identified by Duke Reporters' Lab.[12]

All three provide fact checking through their own websites, as well as through Facebook and Twitter. Figure 2 is an example of a Twitter post with content checked by Snopes.

Consequently, the more political or controversial issues a fact-checking service covers, the more it needs to build a reputation for usefulness and trustworthiness.

Analyzing Social Media Conversations

To explore how social media users perceive the trustworthiness and usefulness of these services, we applied a research approach designed to take advantage of unstructured social media conversations (see Figure 3).

While investigations of trust and usefulness often rely on structured data from questionnaire-based surveys, social media conversations represent a highly relevant data source for our purpose, as they arguably reflect the raw, authentic perceptions of social media users. Xu et al.[16] claim it is beneficial to listen to, analyze, and understand citizens' opinions through social media to improve societal decision-making processes and solutions. They wrote, for example, "Social media analytics has been applied to explain, detect, and predict disease outbreaks, election results, macroeconomic processes (such as crime detection), (...) and financial markets (such as stock price)."[16] Social media conversations take place in the everyday context of users likely to be engaged in fact-checking services. This approach may provide a more unbiased view of people's perceptions than, say, a questionnaire-based approach. The benefit of gathering data from users in their specific social media context does not imply that our data is representative. Our data lacks important information about user demographics, limiting our ability to claim generality for the entire user population. Despite this potential drawback, however, our data does offer new insight into how social media users view the usefulness and trustworthiness of various categories of fact-checking services.

For data collection, we used Meltwater Buzz, an established service for social media monitoring. crawling data from social media conversations in blogs, discussion forums, online newspaper discussion threads, Twitter, and Facebook. Meltwater Buzz crawls all blogs (such

Figure 4. Positive and negative posts related to trustworthiness and usefulness per fact-checking service (in %); "other" refers to posts not relevant for the research categories (N = 595 posts).

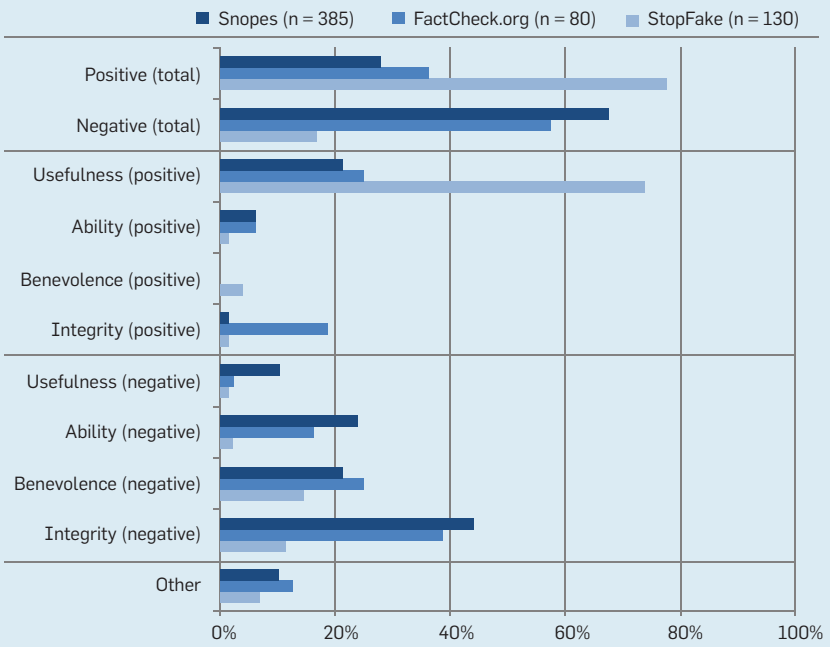


Table 2. Snopes and themes we analyzed (n = 385).

Theme	Sentiment	Example
Usefulness	Positive (21%)	Snopes is a wonderful Website for verifying things seen online; it is at least a starting point for research.
	Negative (10%)	Snopes is a joke. Look at its Boston bombing debunking failing to debunk the worst hoax ever ...
Ability	Positive (6%)	[...] Snopes is a respectable source for debunking wives' tales, urban legends, even medical myths ...
	Negative (24%)	Heh ... Snopes is a man and a woman with no investigative background or credentials who form their opinions solely on Internet research; they don't interview anyone. [...]
Benevolence	Positive (0%)	No posts
	Negative (21%)	You show your Ignorance by using Snopes ... Snopes is a NWO Disinformation System designed to fool the Masses ... SORRY. I Believe NOTHING from Snopes. Snopes is a Disinformation vehicle of the Elitist NWO Globalists. Believe NOTHING from them ... [...]
Integrity	Positive (2%)	Snopes is a standard, rather dull fact-checking site, nailing right and left equally. [...]
	Negative (44%)	Snopes is a leftist outlet supported with money from George Soros. Whatever Snopes says I take with a grain of salt ...

as https://wordpress.com/), discussion forums (such as https://offtopic.com/), and online newspapers (such as https://www.washingtonpost.com/) requested by Meltwater customers, thus representing a large, though convenient, sample. It collects various amounts of data from each platform; for example, it crawls all posts on Twitter but only the Facebook pages with 3,500 likes or groups of more than 500 members. This limitation in Facebook data partly explains why the overall number of posts we collected—1,741—was not more than it was.

To collect opinions about social media user perceptions of Snopes and FactCheck.org, we applied the search term "[service name] is," as in "Snopes is," "FactCheck.org is," and "FactCheck is." We intended it to reflect how people start a sentence when formulating their opinions. StopFake is a relatively less-known service. We thus selected a broader search string—"StopFake"—to be able to collect enough relevant opinions. The searches returned a data corpus of 1,741 posts over six months—October 2014 to March 2015—as in Figure 3. By "posts," we mean written contributions by individual users. To create a sufficient dataset for analysis, we removed all duplicates, including a small number of non-relevant posts lacking personal opinions about fact checkers. This filtering process resulted in a dataset of 595 posts.

We then performed content analysis, coding all posts to identify and investigate patterns within the data[1] and reveal the perceptions users express in social media about the three fact-checking services we investigated. We analyzed their perceptions of the usefulness of fact-checking services through a usefulness construct similar to the one used by Tsakonas et al.[14] "Usefulness" concerns the extent the service is perceived as beneficial when doing a specific fact-checking task, often illustrated by positive recommendations and characterizations (such as the service is "good" or "great"). Following Mayer et al.'s theoretical framework,[7] we categorized trustworthiness according to the perceived ability, benevolence, and integrity of the services. "Ability" concerns the extent a service is perceived as having available the needed skills and expertise, as well as being reputable and well regarded. "Benevolence" refers to the extent a service is perceived as intending to do good, beyond what would be expected from an egocentric motive. "Integrity" targets the extent a service is generally viewed as adhering to an acceptable set of principles, in particular being independent, unbiased, and fair.

Since we found posts typically reflect rather polarized perceptions of the studied services, we also grouped the codes manually according to sentiment, positive or negative. Some posts described the services in a plain and objective manner. We thus coded them using a positive sentiment (see Table 1) because they refer to the

service as a source for fact checking, and users are likely to reference fact-checking sites because they see them as useful.

For reliability, both researchers in the study did the coding. One coded all the posts, and the second then went through all the assigned codes, a process repeated twice. Finally, both researchers went through all comments for which an alternative code had been suggested to decide on the final coding, a process that recommended an alternative coding for 153 posts (or 26%).

A post could include more than one of the analytical themes, so 30% of the posts were thus coded as addressing two or more themes.

Results

Despite the potential benefits of fact-checking services, Figure 4 reports the majority of the posts on the two U.S.-based services expressed negative sentiment, with Snopes at 68% and FactCheck.org at 58%. Most posts on the Ukraine-based StopFake (78%) reflected positive sentiment.

The stated reasons for negative sentiment typically concerned one or more of the trustworthiness themes rather than usefulness. For example, for Snopes and FactCheck.org, the negative posts often expressed concern over lack in integrity due to perceived bias toward the political left. Negative sentiment pertaining to the ability and benevolence of the services were also common. The few critical comments on usefulness were typically aimed at discrediting a service, by, say, characterizing it as "satirical" or as "a joke."

Positive posts were more often related to usefulness. For example, the stated reasons for positive sentiment toward StopFake typically concerned the service's usefulness in countering pro-Russian propaganda and trolling and in the information war associated with the ongoing Ukraine conflict.

In line with a general notion of an increasing need to interpret and act on information and misinformation in social media,[6,11] some users included in the study discussed fact-checking sites as important elements of an information war.

Snopes. The examples in Table 2 reflect how negative sentiment in the posts we analyzed on Snopes was rooted in issues pertaining to trustworthiness. Integrity issues typically involved a perceived "left-leaning" political bias in the people behind the service. Pertaining to benevolence, users in the study said Snopes is part of a larger left-leaning or "liberal" conspiracy often claimed to be funded by George Soros, whereas comments on ability typically targeted lack of expertise in the people running the service. Some negative comments on trustworthiness may be seen as a rhetorical means of discrediting a service. Posts expressing positive sentiment mainly argue for the usefulness of the service, claiming that Snopes is, say, a useful resource for checking up on the veracity of Internet rumors.

FactCheck.org. The patterns in the posts we analyzed for FactCheck.org resemble those for Snopes. As in Table 3, the most frequently mentioned trustworthiness concerns related to service integrity; as for Snopes, users said the service is politically biased toward the left. Posts concerning benevolence and ability were also relatively frequent, reflecting user

Table 3. FactCheck.org and themes we analyzed (n = 80).

Theme	Sentiment	Example
Usefulness	Positive (25%)	[...] You obviously haven't listened to what they say. Also, I hate liars. FactCheck is a great tool.
	Negative (3%)	Anyway, "FactCheck" is a joke [...]
Ability	Positive (6%)	The media sources I use must pass a high credibility bar. FactCheck.org is just one of the resources I use to validate what I read ...
	Negative (16%)	[...] FactCheck is NOT a confidence builder; see its rider and sources, Huffpo articles ... REALLY?
Benevolence	Positive (0%)	No posts
	Negative (25%)	FactCheck studies the factual correctness of what major players in U.S. politics say in TV commercials, debates, talks, interviews, and news presentations, then tries to present the best possible fictional and propaganda-like version for its target [...]
Integrity	Positive (19%)	When you don't like the message, blame the messenger. FactCheck is nonpartisan. It's just that conservatives either lie or are mistaken more ...
	Negative (39%)	FactCheck is left-leaning opinion. It doesn't check facts ...

Table 4. StopFake and themes we analyzed (n = 130); note * also coded as integrity/positive.

Theme	Sentiment	Example
Usefulness	Positive (72%)	Don't forget a strategic weapon of the Kremlin is the "web of lies" spread by its propaganda machine; see antidote http://www.stopfake.org/en/news
	Negative (2%)	[...] StopFake! HaHaHa. You won, I give up. Next time I will quote "Saturday Night Live"; there is more truth:)) ...
Ability	Positive (2%)	[...] by the way, the website StopFake.org is a very objective and accurate source exposing Russian propaganda and disinformation techniques. [...]*
	Negative (2%)	[...] Ha Ha ... a flow of lies is constantly sent out from the Kremlin. Really. If so, StopFake needs updates every hour, but the best way it can do that is to find low-grade blog content and make it appear as if it was produced by Russian media [...]
Benevolence	Positive (4%)	[...] StopFake is devoted to exposing Russian propaganda against the Ukraine. [...]
	Negative (14%)	So now you acknowledge StopFake is part of Kiev's propaganda. I guess that answers my question [...]
Integrity	Positive (2%)	[...] by the way, the website StopFake.org is a very objective and accurate source exposing Russian propaganda and disinformation techniques. [...]
	Negative (11%)	[...] Why should I give any credence to StopFake.org? Does it ever criticize the Kiev regime, in favor of the Donbass position? [...]

concern regarding the service as a contributor to propaganda or doubts about its fact-checking practices.

StopFake. As in Table 4, the results for StopFake show more posts expressing positive sentiment than we found for Snopes and FactCheck.org. In particular, the posts included in the study pointed out that StopFake helps debunk rumors seen as Russian propaganda in the Ukraine conflict.

Nevertheless, the general pattern in the reasons users gave us for positive and negative sentiment for Snopes and FactCheck.org also held for StopFake. The positive posts were typically motivated by usefulness, whereas the negative posts reflected the sentiment that StopFake is politically biased ("integrity"), a "fraud," a "hoax," or part of the machinery of Ukraine propaganda ("benevolence").

Discussion

We found users with positive perceptions typically extoled the usefulness of fact-checking services, whereas users with negative opinions cited concerns over trustworthiness. This pattern emerged across all three services. In the following sections, we discuss how these findings provide new insight into trustworthiness as a key challenge when countering online rumors and misinformation[2,9] and why ill-founded beliefs may have such online reach, even though the beliefs are corrected by prominent fact checkers, including Snopes, FactCheck.org, and StopFake.

Usefulness. Users in our sample with a positive view of the services mainly pointed to their usefulness. While everyone should exercise caution when comparing the various services, topic-specific StopFake is perceived as more useful than Snopes and FactCheck.org. One reason might be that a service targeting a specific topic faces less criticism because it attracts a particular audience that seeks facts supporting its own view. For example, StopFake users target anti-Russian, pro-Ukrainian readers. Another, more general, reason might be that positive perceptions are motivated by user needs pertaining to a perceived high load of misinformation, as in the case of the Ukraine conflict, where media reports and social media are seen as overflowing with propaganda. Others highlighted the general ease information may be filtered or separated from misinformation through sites like Snopes and FactCheck.org, as expressed like this:

"As you pointed out, it doesn't take that much effort to see if something on the Internet is legit, and Snopes is a great place to start. So why not take that few seconds of extra effort to do that, rather than creating and sharing misleading items."

This finding suggests there is increasing demand for fact-checking services,[6] while at the same time a substantial proportion of social media users who would benefit from such services do not use them sufficiently. The services should thus be even more active on social media sites like Facebook and Twitter, as well as in online discussion forums, where greater access to fact checking is needed.

Trustworthiness. Negative perceptions and opinions about fact-checking services seem to be motivated by basic distrust rather than rational argument. For some users in our sample, lack of trust extends beyond a particular service to encompass the entire social and political system. Users with negative perceptions thus seem trapped in a perpetual state of informational disbelief.

While one's initial response to statements reflecting a state of informational disbelief may be to dismiss them as the uninformed paranoia of a minority of the public, the statements should instead be viewed as a source of user insight. The reason the services are often unsuccessful in reducing ill-founded perceptions[9] and people tend to disregard fact checking that goes against their preexisting beliefs[2,13] may be a lack of basic trust rather than a lack of fact-based arguments provided by the services.

We found such distrust is often highly emotional. In line with Silverman,[11] fact-checking sites must be able to recognize how debunking and fact checking evoke emotion in their users. Hence, they may benefit from rethinking the way they design and present themselves to strengthen trust among users in a general state of informational disbelief. Moreover, users of online fact-checking sites should compensate for the lack of physical evidence online by being, say, demonstrably independent, impartial, and able to clearly distinguish fact from opinion. Rogerson[10] wrote that fact-checking sites exhibit varying levels of rigor and effectiveness. The fact-checking process and even what are considered "facts" may in some cases involve subjective interpretation, especially when actors with partial ties aim to provide the service. For example, in the 2016 U.S. presidential campaign, the organization "Donald J. Trump for President" invited Trump's supporters to join a fact-check initiative, similar to the category "topics or controversies," urging "fact checking" the presidential debates on social media. However, the initiative was criticized as mainly promoting Trump's views and candidacy.[5]

Users of fact-checking sites ask: Who actually does the fact checking and how do they do it? What organizations are behind the process? And how does the nature of the organiza-

Table 5. Challenges and our related recommendations for fact-checking services.

		Challenges	Recommendations
Usefulness		Unrealized potential in public use of fact-checking services	Increase presence in social media and discussion forums
Trustworthiness	Ability	Critique of expertise and reputation	Provide nuanced but simple overview of the fact-checking process where relevant sources are included
	Benevolence	Suspicion of conspiracy and propaganda	Establish open policy on fact checking and open spaces for collaboration on fact checking
	Integrity	Perception of bias and partiality	Ensure transparency on organization and funding. and demonstrable impartiality in fact-checking process

tion influence the results of the fact checking? Fact-checking sites must thus explicate the nuanced, detailed process leading to the presented result while keeping it simple enough to be understandable and useful.[11]

Need for transparency. While fact-checker trustworthiness is critical, fact checkers represent but one set of voices in the information landscape and cannot be expected to be benevolent and unbiased just because they check facts. Rather, they must strive for transparency in their working process, as well as in their origins, organization, and funding sources.

To increase transparency in its processes, a service might try to take a more horizontal, collaborative approach than is typically seen in the current generation of services. Following Hermida's recommendation[4] to social media journalists, fact checkers could be set up as a platform for collaborative verification and genuine fact checking, relying less on centralized expertise. Forming an interactive relationship with users might also help build trust.[6,7]

Conclusion

We identified a lack of perceived trustworthiness and a state of informational disbelief as potential obstacles to fact-checking services reaching social media users most critical to such services. Table 5 summarizes our overall findings and discussions, outlining related key challenges and our recommendations for how to address them.

Given the exploratory nature of this study, we cannot conclude our findings are valid for all services. In addition, more research is needed to be able to make definite claims on systematic differences among the various fact checkers based on their "areas of concern." Nevertheless, the consistent pattern in opinions we found across three prominent services suggests challenges and recommendations that can provide useful guidance for future development in this important area.

Users with negative perceptions thus seem trapped in a perpetual state of informational disbelief.

Acknowledgments

This work was supported by the European Commission co-funded FP 7 project REVEAL (Project No. FP7-610928, http://www.revealproject.eu/) but does not necessarily represent the views of the European Commission. We also thank Marika Lüders of the University of Oslo and the anonymous reviewers for their insightful comments.

References

1. Ezzy, D. *Qualitative Analysis.* Routledge, London, U.K., 2013.
2. Friesen, J.P., Campbell, T.H., and Kay, A.C. The psychological advantage of unfalsifiability: The appeal of untestable religious and political ideologies. *Journal of Personality and Social Psychology 108*, 3 (Nov. 2014), 515–529.
3. Gingras, R. Labeling fact-check articles in Google News. *Journalism & News* (Oct. 13, 2016); https://blog.google/topics/journalism-news/labeling-fact-check-articles-google-news/
4. Hermida, A. Tweets and truth: Journalism as a discipline of collaborative verification. *Journalism Practice 6*, 5-6 (Mar. 2012), 659–668.
5. Jamieson, A. 'Big League Truth Team' pushes Trump's talking points on social media. *The Guardian* (Oct. 10, 2016); https://www.theguardian.com/us-news/2016/oct/10/donald-trump-big-league-truth-team-social-media-debate
6. Kriplean, T., Bonnar, C., Borning, A., Kinney, B., and Gill, B. Integrating on-demand fact-checking with public dialogue. In *Proceedings of the 17th ACM Conference on Computer-Supported Cooperative Work & Social Computing* (Baltimore, MD, Feb. 15–19). ACM Press, New York, 2014, 1188–1199.
7. Mayer, R.C., Davis, J.H., and Schoorman, F.D. An integrative model of organizational trust. *Academy of Management Review 20*, 3 (1995), 709–734.
8. Morejon, R. How social media is replacing traditional journalism as a news source. *Social Media Today Report* (June 28, 2012); http://www.socialmediatoday.com/content/how-social-media-replacing-traditional-journalism-news-source-infographic
9. Nyhan, B. and Reifler, J. When corrections fail: The persistence of political misperceptions. *Political Behavior 32*, 2 (June 2010), 303–330.
10. Rogerson, K.S. Fact checking the fact checkers: Verification Web sites, partisanship and sourcing. In *Proceedings of the American Political Science Association* (Chicago, IL, Aug. 29–Sept. 1). American Political Science Association, Washington, D.C., 2013.
11. Silverman, C. *Lies, Damn Lies, and Viral Content. How News Websites Spread (and Debunk) Online Rumors, Unverified Claims, and Misinformation.* Tow Center for Digital Journalism, Columbia Journalism School, New York, 2015; http://towcenter.org/wp-content/uploads/2015/02/LiesDamnLies_Silverman_TowCenter.pdf
12. Stencel, M. International fact checking gains ground, Duke census finds. Duke Reporters' Lab, Duke University, Durham, NC, Feb. 28, 2017; https://reporterslab.org/international-fact-checking-gains-ground/
13. Stroud, N.J. Media use and political predispositions: Revisiting the concept of selective exposure. *Political Behavior 30*, 3 (Sept. 2008), 341–366.
14. Tsakonas, G. and Papatheodorou, C. Exploring usefulness and usability in the evaluation of open-access digital libraries. *Information Processing & Management 44*, 3 (May 2008), 1234–1250.
15. Van Mol, C. Improving web survey efficiency: The impact of an extra reminder and reminder content on Web survey response. *International Journal of Social Research Methodology 20*, 4 (May 2017), 317–327.
16. Xu, C., Yu, Y., and Hoi, C.K. Hidden in-game intelligence in NBA players' tweets. *Commun. ACM 58*, 11 (Nov. 2015), 80–89.

Petter Bae Brandtzaeg (pbb@sintef.no) is a senior research scientist at SINTEF in Oslo, Norway.

Asbjørn Følstad (asf@sintef.no) is a senior research scientist at SINTEF in Oslo, Norway.

DOI:10.1145/3096742

Exploring the many distinctive elements that make securing HPC systems much different than securing traditional systems.

BY SEAN PEISERT

Security in High-Performance Computing Environments

HOW IS COMPUTER security different in a high-performance computing (HPC) context from a typical IT context? On the surface, a tongue-in-cheek answer might be, "just the same, only faster." After all, HPC facilities are connected to networks the same way any other computer is, often run the same, typically Linux-based operating systems as are many other common computers, and have long been subject to many of the same styles of attacks, be they compromised credentials, system misconfiguration, or software flaws. Such attacks have ranged from the "wily hacker" who broke into U.S. Department of Energy (DOE) and U.S. Department of Defense (DOD) computing systems in the mid-1980s,[42] to the "Stakkato" attacks against NCAR, DOE, and NSF-funded supercomputing centers in

» key insights

- High-performance computing systems have some similarities and some differences with traditional IT computing systems, which present both challenges and opportunities.
- One challenge is that HPC systems are "high-performance" by definition, and so many traditional security techniques are not effective because they cannot keep up with the system or reduce performance.
- Many opportunities also exist: HPC systems tend to be used for very distinctive purposes, have much more regular and predictable activity, and contain highly custom hardware/software stacks. Each of these elements can provide a toehold for leveraging some aspect of the HPC platform to improve security.

PHOTO BY GORODENKOFF VISUALS

the mid-2000s,[24,39] to the thousands of probes, scans, brute-force login attempts, and buffer overflow vulnerabilities that continue to plague high-performance computing facilities today.

On the other hand, some HPC systems run highly exotic hardware and software stacks. In addition, HPC systems have very different purposes and modes of use than most general-purpose computing systems, of either the desktop or server variety. This fact means that aside from all of the normal reasons that any network-connected computer might be attacked, HPC computers have their own distinct systems, resources, and assets that an attacker might target, as well as their own distinctive attributes that make securing such systems somewhat distinct from securing other types of computing systems.

The fact that computer security is context- and mission-dependent should not be surprising to security professionals—"security policy is a statement of what is, and what is not, allowed,"[7]—and each organization, will therefore have a somewhat distinctive security policy. For example, a mechanism designed to enforce a particular policy considered essential for security by one site might be considered a denial of service to legitimate users of another site, or how a smartphone is protected is distinct from a desktop computer. Thus, for HPC systems, we must ask what is the desired functioning of the system so that we can establish what the security policies are and better understand the mechanisms with which those policies can be enforced.

On the other hand, historically, security for HPC systems has not necessarily been treated as distinct from general-purpose computing, except, typically, making sure that security does not get in the way of performance or usability. While laudable, this article argues that this assessment of HPC's distinctiveness is incomplete.

This article focuses on four key themes surrounding this issue:

The first theme is that HPC systems are optimized for high performance by definition. Further, they tend to be used for very distinctive purposes, notably mathematical computations.

The second theme is that HPC systems tend to have very distinctive modes of operation. For example, compute nodes in an HPC system may be accessed exclusively through some kind of scheduling system on a login node in which it is typical for a single program or common set of programs to run in sequence. And, even on that login node, from which the computation is submitted to the scheduler, it may be the case that an extremely narrow range of programs exist compared to those commonly found on general-use computing systems.

The third theme is that while some HPC systems use standard operating systems, some use highly exotic stacks. And even the ones that use standard operating systems, very often have custom aspects to their software stacks, particularly at the I/O and network driver levels, and also at the application layer. And, of course, while the systems may use commodity CPUs, the CPUs and other hardware system components are often integrated in HPC systems in a way (for example, by Cray or IBM) that may well exist nowhere else in the world.

The fourth theme, which follows from the first three themes, is that HPC systems tend to have a much more regular and predictable mode of operation, which changes the way security can be enforced.

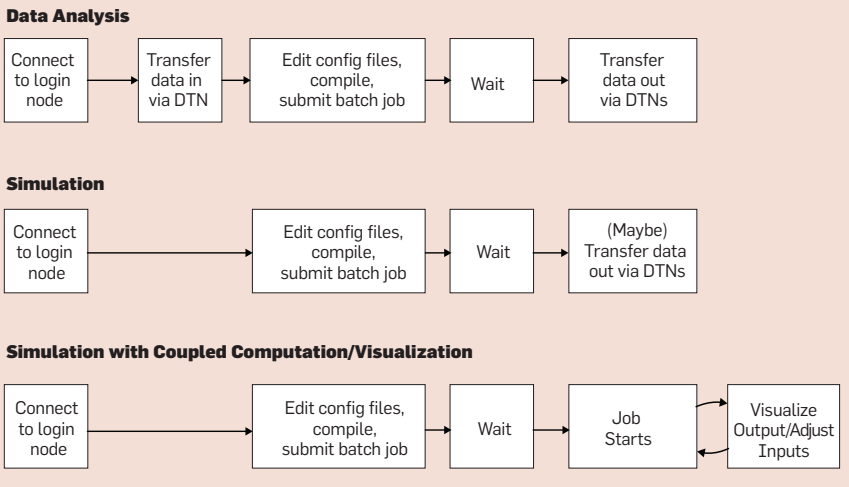
As a final aside, many, but by no means all HPC systems are often extremely open systems from a security standpoint, and may be used by scientists worldwide whose identities have never been validated. Increasingly, we are also starting to see HPC systems in which computation and visualization are more tightly coupled and, a human manipulates the inputs to the computation itself in near-real time.

This distinctiveness presents both opportunities and challenges. This article discusses the basis for these themes and the conclusions for security for these systems.

Scope and threat model. I have spent most of my career in or near "open science:" National Science Foundation and Department of Energy Office of Science-funded high-performance computing centers, and so the lens through which this article is discussed tends to focus on such environments. The challenges in "closed" environments, such as those used by the National Security Agency (NSA), Department of Defense (DoD), or National Nuclear Security Administration (NNSA) National Labs, or commercial industry, shares some, but not all of the attributes discussed in this article. As a result, although I discuss confidentiality, a typical component of the "C-I-A" triad, because even in open science, data leakage is certainly an issue and a threat, this article focuses more on integrity related threats,[31,32] including alteration of code or data, or misuse of computing cycles, and availability related threats, including disruption or denial of service against HPC systems or networks that connect them.

Computations that are incorrect for non-malicious reasons, including flaws in application code, such as general logic errors, round-off errors, nondeterminism in parallel algorithms, unit conversion errors,[20] as well as incorrect assumptions by users about the hardware they are running on, are vital issues, but beyond the scope of this article, due to length and the fact those issues are well-covered elsewhere.[4,5,6,8,36]

Figure 1. Three typical high-level workflow diagrams of scientific computing. The diagram at top shows a typical workflow for data analysis in HPC; the middle diagram shows a typical workflow for modeling and simulation; and the bottom diagram shows a coupled, interactive compute-visualization workflow.



High-Performance Computing Environments

Distinctive purposes. The first theme of the distinctiveness of security for HPC systems is that these systems are high-performance by definition, and are made that way for a reason. They are typically used for automated computation of some kind, typically performing some set of mathematical operations. Historically, this has often been for the purpose of modeling and simulation, and increasingly today, for data analysis as well. Given the primary purpose of HPC systems is therefore high-performance, and given that such systems themselves are both few in number, and therefore also that computing time on such systems is quite valuable, there is a reluctance by the major stakeholders—the funding agencies that support HPC systems as well as the users who run computations on them—to agree to any solution that might impose overhead on the system. Those stakeholders might well regard such a solution as a waste of cycles at worst, and an unacceptable delay of scientific results at best. This is an important detail, because it frames the types of security solutions that at least historically might have been considered acceptable to use.

Distinctive modes of operation. The second theme of the distinctiveness of security for HPC systems is that these systems tend to have distinctive modes of operation. The typical mode of operation for using a scientific high-perfor-

mance machine involves connecting through a login node of some kind. In parallel, at least for data analysis tasks, data that a user wishes to analyze may be copied to the machine via a data transfer node or DTN, and software that a user wishes to install may be copied to the login node as well.

The user is then likely to edit some configuration files, compile their software, and write a "batch script" that defines what programs should be run, along with parameters of how those programs should be run. This is because most significant jobs are not run on the login nodes themselves, because the login nodes have very limited resources. Rather, many institutions use compute nodes, which cannot be logged into directly, but rather have a batch scheduler that determines when jobs should run based on analyzing the batch scripts that have been submitted according to a given optimization policy for the site in question. Thus, after writing their batch script, the user will probably submit their job to a batch queue using a submission program, and then log out and wait for the job to run on the compute nodes.

Following that, the user may run some kind of additional analysis or visualization on the data that was output. This may happen on the HPC system, or the output of the HPC computation may be downloaded to a non-HPC system for analysis in a separate environment such as using Jupyter/IPython.[33] This additional analysis or visualization might happen serially, following the completed execution on the HPC system, or, alternatively, may happen in an interactive, tightly-coupled fashion such that the user visualizing the output of the computation can manipulate the computation as it is taking place.[37,45] It should be noted that the "coupled" computation/analysis model could involve network connections external to the HPC facility, or, and particularly as envisioned by the "superfacility" model for data-intensive science,[50] may involve highly specialized and optimized network connections within a single HPC center. Examples of all three workflows are shown in Figure 1.

These use cases are often in stark contrast to the plethora of software that is typically run on a general-purpose desktop system, such as Web browsers, email clients, Microsoft Office, iTunes Music, Adobe Acrobat, personal task managers, Skype, and instant messaging. And, importantly, this is often a much smaller set of programs with a much more regular sequence of events in which the use of one program directly follows from another, as well, rather than the constant attention-span-driven context switching of the use of general-purpose computers. For example, on the NERSC HPC systems, in 2014, for over 5950 unique users that were active in 2014, just 13 applications comprised 50% of the cycles consumed, 25 applications comprised 66% of the cycles, and 50 applications comprised 80% of the cycles.[2] The consequences of these distinctive workflows are important, as we will discuss.

For HPC systems, we must ask what is the desired functioning of the system so that we can establish what the security policies are and better understand the mechanisms with which those policies can be enforced.

Custom operating system stacks. The third theme of the distinctiveness of security for HPC systems is that these systems often have highly exotic stacks. Current HPC environments represent a spectrum of hardware and software components, ranging from exotic and highly custom to fairly commodity.

As an example, "Cori Phase 1,"[a] the newest supercomputer at NERSC, is a Cray XC based on Intel Haswell processors, leveraging Cray Aries interconnects, a Lustre file system, and nonvolatile memory express (NVMe) in the burst buffer that is user accessible. Cori runs a full SUSE Linux distribution on the login nodes and Compute Node Linux (CNL),[44] a light-weight version of the Linux kernel and run-time environment based on the SuSE Linux Enterprise Server distribution.

Mira,[b] at the Argonne Leadership Computing Facility, is a hybrid system. The login nodes are IBM Power 7-based systems. The compute nodes are an IBM Blue Gene/Q system based on PowerPC A2 processors, IBM's 5D torus interconnect, and a similarly elaborate memory structure. The I/O nodes also use PowerPC A2 processors and are connected using Mellanox Infiniband QDR switches. The login nodes run Red Hat Linux. The compute nodes run Compute Node Kernel (CNK),[1] a Linux-like OS for compute nodes, but

a http://www.nersc.gov/users/computational-systems/cori/configuration/

b https://www.alcf.anl.gov/user-guides/machine-overview

support neither multi-tasking or virtual memory[27] (CNK has no relationship with CNL). The I/O system runs the GPFS file system client.

Aurora,[c] the system scheduled to be installed at ALCF in 2019, will be constructed by a partnership between Cray and Intel and will run third-generation Intel Xeon Phi processors with second-generation Intel Omni-Path photonic interconnects and a variety of ash memory and NVRAM components to accelerate I/O, including 3DXpoint and 3D NAND in multiple locations, all user accessible. Aurora will run Cray Linux[10]—a full Linux stack on its login nodes and I/O nodes (though the I/O nodes do not allow general user access), and mOS[46] on its compute nodes. mOS supports both a lightweight kernel (LWK) and full Linux operating system to enable users to choose between avoiding unexpected operating system overhead, and the flexibility of a full Linux stack.

Summit,[d] the system scheduled to be installed at OLCF in 2018, will be based on both IBM POWER9 CPUs and NVIDIA Volta GPUs, with NVIDIA NVLink on-node networks and dual-rail Mellanox interconnects.

In short, there is certainly some variation on exactly what operating systems are run—in all cases, login nodes run "full" operating systems. And in some cases, full operating systems are also used for compute nodes, while in other cases, lighter-weight but Linux API-compatible versions of operating systems are used, while in some cases entirely custom operating systems are used that are single-user only, and contain no virtual memory capabilities or multitasking.

At least for the full operating systems, it is reasonable to assume the operating systems contain similar or identical capabilities and bugs as standard desktop and server versions of Linux, are just as vulnerable to attack via various pieces of software (libraries, runtime, and application) that are running on the system.

Custom hardware and software components may have both positives and negatives. On one hand, they may receive less assurance than more common stacks. On the other hand, some custom stacks may be smaller, more easily verified, and less complex.

There is a reluctance by major stakeholders—the funding agencies that support HPC systems as well as the users who run computations on them—to agree to any solution that might impose overhead on the system.

Openness. Our final theme is the relative "openness" of at least some HPC systems. That is, scientists from all over the world whose identities have never been validated may use them. For example, many such systems, such as those used by NSF or DOE ASCR, have no traditional firewalls between the data transfer nodes and the Internet, let alone the ability to "air gap" the HPC system (that is, ensure no physical connection to the regular Internet is possible) as some communities are able to do.

Security Mechanisms and Solutions that Overcome the Constraints of HPC Environments

Traditional IT security solutions, including network and host-based intrusion detection, access controls, and software verification work about as well in HPC as traditional IT (often not very), or worse, due to constraints in HPC environments.

For example, traditional host-based security mechanisms, such as those leveraging system call data via audited, as well as certain types of network security mechanisms, like network firewalls and firewalls doing deep packet inspection, may be antithetical to the needs of the system being protected. For example, it has been shown that even 0.0046% packet loss (1 out of 22,000 packets) can cause a loss in throughput of network data transfers of approximately 90%.[13] Given that stateful and/or deep-packet inspecting firewalls can cause delays that might lead to such loss, a firewall, as traditionally defined, is inappropriate for use in environments with high network data throughput requirements.

Thus, alternative approaches must be applied. Some solutions exist that can help compensate for these constraints.

The Science DMZ[13] security framework defines a set of security policies, procedures, and mechanisms to address the distinct needs of scientific environments with high network throughput needs (HPC security theme #1). While the needs of high throughput networks do not eliminate options for security monitoring

c http://aurora.alcf.anl.gov
d https://www.olcf.ornl.gov/summit/

or mitigation, those requirements do change what is possible.

In particular, in the Science DMZ framework, the scientific computing systems are moved to their own enclave, away from other types of computing systems that might have their own distinctive security needs and perhaps even distinct regulations—for example, financial, human resources, and other business computing systems. In addition, it directs transfers through single network ingress and egress point that can be monitored and restricted.

However, the Science DMZ does not use "deep packet inspecting" or stateful firewalls. It does leverage packet filtering firewalls that is, firewalls that examine only attributes of packet headers and not packet payloads. And, separately, it also performs deep packet inspection and stateful intrusion detection, such as might be done with the Bro Network Security Monitor.[28] However, the two processes are not directly coupled, as, unlike a firewall, the IDS is not used in-line with the network traffic, and as a result, delays are not imposed on transmission of the traffic due to inspection, and thus congestion that might lead to packet loss and retransmission is also not created.

Thus, by moving the traffic to its own enclave that can be centrally monitored at a single point, the framework seeks to maintain a similar level of security to traditional organizations that typically have a single ingress/egress point, rather than simply removing network monitoring without replacing it with an alternative. However, the Science DMZ does so in a very specific way that accommodates the type and volume of network traffic used in scientific and high-performance computing environments. More specifically, it achieves throughput by reducing complexity, which is a theme that we will return to in this article.

The Science DMZ framework has been implemented widely in university and National Lab environments around the world as a result of funding from NSF, DOE ASCR, and other, international funding organizations, to support computing and networking infrastructure for open science. It goes without saying that both the Science DMZ framework and the Bro IDS must also continue to be adapted to more types of HPC environments, such as those requiring environments with greater data confidentiality guarantees, such as medical, defense, and intelligence environments. Steps have been made toward the medical context as well.

The Medical Science DMZ[29] applies the Science DMZ framework to computing environments requiring compliance with HIPAA Security Rule. Key architectural aspects include the notion that all traffic from outside compute/storage infrastructure passes through heavily monitored head nodes, that storage and compute nodes themselves are not connected directly to the Internet, and that traffic containing sensitive or controlled access data is encrypted. However, further work in medical environments, as well as other environments is required.

Leveraging the Distinctiveness of HPC as an Opportunity

The Science DMZ helps compensate for HPC's limitations—we need more such solutions. As indicated by the four themes enumerated in this article, we also need solutions that can leverage HPC distinctiveness as a strength.

Sommer and Paxson[41] point out the fact that anomaly-based detection typically is not used in traditional IT environments is due to the high-level fact that "finding attacks is fundamentally different from … other applications" (such as credit card fraud detection, for example). Among other key issues, they note that network traffic is often much more diverse than one might expect. They point out that semantic understanding is a vital component of overcoming this limitation to enable machine-learning approaches to security to be more effective.

On the other hand, as mentioned earlier, HPC systems tend to be used for very distinctive purposes, notably mathematical computations (theme #1). The specific application of HPC systems varies by the organization that uses them (for example, DOE National Lab, DOD lab), but each individual system typically has a very specific use. This is a key point because the result may be that both specification-based and anomaly-based intrusion detection may be more useful in HPC environments than in traditional IT environments. Specifically, given the hypothesis that patterns of behavior in HPC are likely more regular than in typical computing systems, one might expect that one can reduce the error rates when using anomaly-based intrusion detection, and possibly even making specifications possible to construct for specification-based intrusion detection. Thus, such security mechanisms might even fare better in HPC environments than in traditional IT environments (theme #4), though demonstrating the degree to which the increased regularity of HPC environments may be helpful for security analysis is an open research question.

Analyzing system behavior with machine learning. A second, and related key point about HPC systems being used primarily for mathematical computation is that if we can do better analysis of system behavior, the insight that most HPC machines are used for computation focuses our attention on what security risks to care about (for example, users running "illicit computations," as defined by the owners of the HPC system) and might give us better ability to understand what type of computation is taking place.

An example of a successful approach to addressing this question involved research that I was involved with at Berkeley Lab between 2009–2013.[14,30,47,48] In this project, we asked the questions: What are people running on HPC systems? Are they running what they usually run? Are they running what they requested cycle allocations to run, or mining Bitcoins?

Are they running something illegal (for example, classified)? In that work, we developed technique for answering these questions by fingerprinting communication on HPC systems.

Specifically, we collected Message Passing Interface (MPI) function calls via the Integrated Performance Monitoring (IPM)[43] tool, which showed patterns of communication between ores in an HPC system, as shown in Figure 2.

Using 1681 logs for 29 scientific applications from NERSC HPC systems, we applied Bayesian-based machine learning techniques for classification of scientific computations, as well as a graphtheoretic approach using "approximate" graph matching techniques (subgraph isomorphism and edit distance). A hybrid machine learning and graph theory approach identified test

HPC codes with 95%–99% accuracy.

Our work analyzing distributed memory parallel computation patterns on HPC compute nodes is by no means conclusive that anomaly detection is an unqualified success on HPC systems for intrusion detection. For one thing, the experiments were not conducted in an adversarial environment, and so the difficultly of an attacker intentionally evading detection by attempting to make one program look like another was not explored. In addition, in our "fingerprinting HPC computation" project, we had what we deemed to be a reasonable, though not exhaustive corpus of data representative of typical computations on NERSC facilities to examine. In addition, in examining the data, we focused on a specific set of activity contained within the NERSC Acceptable Use.

Policy as falling outside of "acceptable use." Other sites will have a different baseline of "typical computation," and are also likely have somewhat different policies that define what is or is not "illicit use."

However, regardless, we do believe the approach is an example of the type of techniques that could possibly have success in HPC environments and possibly even greater success than in many non-HPC environments. For example, consider the possibility of a skilled attacker attempting to evade detection something that any security mechanism relying on machine learning is vulnerable to. Not only do there appear to be more regular use patterns in HPC environments, but there also exist certain distinctive security policies in HPC environments that might help improve the usefulness of application-level use monitoring. There are at least two reasons for this.

First, given the organization responsible for security of HPC systems are likely to care more about misuse of cycles if very large numbers of cycles are used, this suggests focusing on the users that use cycles for many hours per day for days at a time. This is a very different practical scenario than network security monitoring where a decision about security might require a response in a fraction of a second in order to prevent compromise. Given the longer time scale, therefore, a human security analyst can be involved rather than requiring the application monitoring, on the level that we have done it, to be conclusive. Rather, that application monitoring might simply serve to focus an analyst's attention, and to lead to a manual source code analysis, or even an actual conversation with the user whose account was used to run the code.

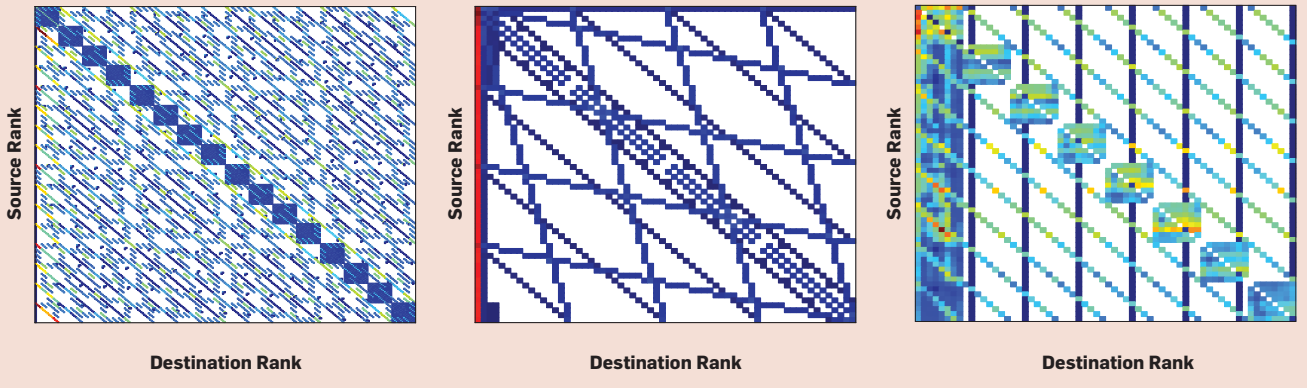
A second reason why this issue of an attacker evading detection on HPC might be harder is because, users are often given "cycle allocations" to run code. As a result, the more a program running on an HPC system is modified to mask illicit use, the more likely it is that additional cycles must be used to do additional tasks to make it look like the program is doing something different than it actually is. Thus, the faster that a stolen allocation will be used up and/or the longer it will take the HPC system to accomplish whatever illicit use the attacker is attempting.

Collecting better audit and provenance data. It is important to note the success of the work mentioned in the previous section is dependent on availability of useful security monitoring data. It is our observation that the current trend in many scientific environments on collecting provenance data for scientific reproducibility purposes, such as the Tigres workflow system,[38] and the DOE Biology Knowledgebase (KBase)[21] may help to provide better data that can be used for security monitoring, as might DARPA's "Transparent Computing" program 11, which seeks to "make currently opaque computing systems transparent by providing high-fidelity visibility into component interactions during system operation across all layers of software abstraction, while imposing minimal performance overhead."

In line with this, as noted earlier, HPC systems have a lot in common with traditional systems, but also contain a lot of highly custom OS and network-level, and application-level software. A key point here is that such exotic hardware and low-level software stacks may also provide opportunities for monitoring data going forward. An example of the performance counters used in many of today's HPC machines is an example of this.

Post-exascale systems, as well as more architectures that are still in their early phases of practical implementation, such as neuromorphic computing, quantum computing, and

Figure 2. "Adjacency matrices" for individual runs of a performance benchmark, an atmospheric dynamics simulator, and a linear equation solver SUPERLU. Number of bytes sent between ranks is linearly mapped from dark blue (lowest) to red (highest), with white indicating an absence of communication.[47,48]



photonic computing may all provide additional challenges and opportunities. For example, though neural networks were previously thought by many to be inscrutable,[16] new research suggests this may be actually possible at some point.[12,49] If successful, this might give to rise to the ability to interpret networks learned by neuromorphic chips.

Looking to the Future

In the future, it is clear that numerous aspects of HPC will change, both for the good of security and in ways that complicate it.

One key component of the National Strategic Computing Initiative is that software engineering is a key goal of the NSCI, and so perhaps automated static/runtime analysis tools might be developed and used to check HPC code for insecure behaviors.

On the other hand, science is also changing. For example, distributed, streaming sensor data collection is increasingly a source of data used in HPC. In short, science data is getting to us in new ways, and we also have more data than ever to protect.

Another change is that on HPC systems running full operating systems, we are starting to see an increasing shift toward the use of new virtualized environments for additional flexibility. In particular, as Docker containers[25] and CoreOS's Rocket[9] become more popular for virtual replication and containment in many IT environments, rather than replicating full virtual operating systems, Docker-like containers that are more appropriate to HPC environments, such as Shifter[19] or Singularity[23] are also gaining attention and use. This notion of "containerization" may well be a key benefit to security, both because of the way that containerization done properly typically limits the damage that an attacker can do, as well as because it simplifies the operation of the machine, and the reduction of complexity is also often a key benefit to system robustness, including security.

The superfacility model in which computation and visualization are more frequently tightly coupled than they currently are, seems also likely to increase. At the same time, the notion of "science gateways" essentially Web portals, providing limited interfaces to HPC, rather than full-blown UNIX command-line interfaces, may provide a reduction of complexity that superfacility would otherwise introduce. While science gateways still represent vulnerability vectors from arbitrary code, even when it is submitted via Web front-ends, since security tends to benefit from more constrained operation, the general toward science gateways may also enhance security.

In the future, it is clear that numerous aspects of HPC will change, both for the good of security and in ways that complicate it.

Finally, the prospect of new and novel security technologies, such as simulated homomorphic encryption,[34,35] differential privacy,[15] and cryptographic mechanisms for securing chains of data[3,18,40] such as blockchains,[26] may also may provide new means for interacting with data sets in a constrained fashion.

For example, there may be cases where the owners of the data want to keep the raw data for themselves for an extended period of time, such as a scientific embargo. Or there may be cases where the owners of the data are unable to share the raw data due to privacy regulations, such as on medical data, system and network data that contains personally identifiable information, or sensor data containing sensitive (for example, location) information. In either case, the data owners may still wish to find a way to enable some limited type of computation on the data, or share data, but only with a certain degree of resolution. With CryptDB[34] and Mylar,[35] Popa et al. have demonstrated approaches for efficiently searching over encrypted data without requiring fully homomorphic encryption,[17] which is currently at least a million times slow to be used practically, let alone in HPC environments. Likewise, differential privacy,[15] and perhaps particularly distributed differential privacy[22] may provide new opportunities for sharing and analyzing data to be used in HPC environments as well. And in addition, blockchains and similar technologies may provide means for both monitoring the integrity of raw scientific data in HPC contexts, as well as for maintaining secure audit trails of accesses to or modifications of raw data.

Summary

Modern HPC systems do some things very similar to ordinary IT computing,

but they also have some significant differences. This article presented both challenges and opportunities.

Two key security challenges are the notions that traditional security solutions often are not effective given the paramount priority of high-performance in HPC. In addition, the need to make some HPC environments as open as possible to enable broad scientific collaboration and interactive HPC also presents a challenge.

There may also be opportunities, as described by the four themes regarding HPC security presented here. The fact that HPC systems tend to be used for very distinctive purposes, notably mathematical computations, may mean the regularity of activity within HPC systems can benefit the effectiveness of machine learning analyses on security monitoring data to detect misuse of cycles and threats to computational integrity. In addition, custom stacks provide opportunities for enhanced security monitoring, and the general trend toward containerized operation, limited interfaces, and reduced complexity in HPC is likely to help in the future much as reduced complexity has benefitted the Science DMZ model.

Acknowledgments

Appreciation to Deb Agarwal, David Brown, Jonathan Carter, Phil Colella, Dan Gunter, Inder Monga, and Kathy Yelick for their valuable feedback and to Sean Whalen and Bogdan Copos for their excellent work underlying the ideas for new approaches described here. Thanks to Glenn Lockwood for his insights on the specifications for the DOE ASCR hardware and software coming in the next few years, and both Glenn Lockwood and Scott Campbell for the time spent providing the data that supported that research.

This work used resources of the National Energy Research Scientific Computing Center and was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author and do not necessarily reflect those of the employers or sponsors of this work.

References

1. Adiga, N.R. et al. An overview of the Blue-Gene/L supercomputer. In *Proceedings of the ACM/IEEE Conference on Supercomputing*, 2002.
2. Austin, B. et al. 2014 NERSC Workload Analysis (Nov. 5., 2015); http://portal.nersc.gov/project/mpccc/baustin/NERSC_2014_Workload_Analysis_v1.1.pdf.
3. Anderson, R.J. UEPS: A second-generation electronic wallet. In *Proceedings of the 2nd European Symposium on Research in Computer Security* (Nov. 1992), 411–418.
4. Bailey, D.H. Resolving numerical anomalies in scientific computation, 2008.
5. Bailey, D.H., Borwein, J.M. and Stodden, V. Facilitating reproducibility in scientific computing: Principles and practice. *Reproducibility: Principles, Problems, Practices*. H. Atmanspacher and S. Maasen, Eds. John Wiley and Sons, New York, NY, 2015.
6. Bailey, D.H., Demmel, J., Kahan, W., Revy, G. and Sen, K. Techniques for the automatic debugging of scientific floating-point programs. In *Proceedings of the 14th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics* (Lyon, France, Sept. 2010).
7. Bishop, M. *Computer Security: Art and Science*. Addison-Wesley Professional, Boston, MA, 2003.
8. Cappello, F. Improving the trust in results of numerical simulations and scientific data analytics. 2015.
9. CoreOS, Inc. rkt - App Container runtime. https://github.com/coreos/rkt.
10. Cray, Inc. Cray Linux Environment Software Release Overview, s-2425-52xx edition (Apr 2014); http://docs.cray.com/books/S-2425-52xx.
11. DARPA. Transparent Computing; http://www.darpa.mil/Our_Work/I2O/Programs/Transparent_Computing.aspx.
12. Das, A., Agrawal, H., Zitnick, C.L., Parikh, D. and Batra, D. Human attention in visual question answering: Do humans and deep networks look at the same regions? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016.
13. Dart, E., Rotman, L., Tierney, B., Hester, M. and Zurawski, J. The science DMZ: A network design pattern for data-intensive science. In *Proceedings of the IEEE/ACM Annual SuperComputing Conference* (Denver CO, 2013).
14. DeMasi, O., Samak, T. and Bailey, D.H. Identifying HPC codes via performance logs and machine learning. In *Proceedings of the Workshop on Changing Landscapes in HPC Security* (2013).
15. Dwork, C. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, Part II. Lecture Notes in Computer Science 4052*, (July 2006), 1–12. Springer Verlag.
16. Gefter, A. Is artificial intelligence permanently inscrutable? *Nautilus 40* (Sept. 1, 2016).
17. Gentry, C. Computing arbitrary functions of encrypted data. *Commun. ACM 53*, 3 (Mar. 2010), 97–105.
18. Haber, S. and Stornetta, W.S. How to time-stamp a digital document. *J. Cryptology 3*, 2 (Jan. 1991), 99–111.
19. Jacobsen, D.M. and Canon, R.S. Contain this, unleashing docker for HPC. *Proceedings of the Cray User Group*, 2015.
20. Jiang, L. and Su, Z. Osprey: A practical type system for validating dimensional unit correctness of c programs. In *Proceedings of the 28th International Conference on Software Engineering*, (2006), 262–271 ACM, New York.
21. KBase: The Department of Energy Systems Biology Knowledgebase; http://kbase.us.
22. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S. and Smith, A. What can we learn privately? *SIAM J. Computing 40*, 3 (2011), 793–826.
23. Kurtzer, G.M. et al. Singularity; http://singularity.lbl.gov.
24. Marko, J. and Bergman, L. Internet attack is called broad and long lasting. *New York Times* (May 10, 2005).
25. Merkel, D. Docker: Lightweight Linux containers for consistent development and deployment. *Linux J.* 239 (2014).
26. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System (May 24, 2009); http://www.bitcoin.org/bitcoin.pdf.
27. Nataraj, A., Malony, A.D., Morris, A. and Shende, S. Early experiences with KTAU on the IBM BG/L. In *European Conference on Parallel Processing*, pp. 99-110. Springer, 2006.
28. Paxson, V. Bro: A system for detecting network intruders in real time. *Computer Networks 31*, 23 (1999), 2435–2463.
29. Peisert, S., et al. The Medical Science DMZ. *J. American Medical Informatics Assoc. 23*, 6 (Nov. 1, 2016).
30. Peisert S. Fingerprinting Communication and Computation on HPC Machines. TR LBNL-3483E, Lawrence Berkeley National Laboratory, June 2010.
31. Peisert, S., et al. ASCR Cybersecurity for Scientific Computing Integrity. TR LBNL-6953E, U.S. Department of Energy Office of Science, Feb. 2015.
32. Peisert, S. et al. ASCR Cybersecurity for Scientific Computing Integrity|Research Pathways and Ideas Workshop. TR LBNL-191105, U.S. Department of Energy Office of Science, Sept. 2015.
33. Pérez, F. and Granger, B.E. IPython: A System for interactive scientific computing. *Computing in Science and Engineering 9*, 3 (May 2007), 21–29.
34. Popa, R.A., Redfield, C., Zeldovich, N. and Balakrishnan, H. Cryptdb: Processing queries on an encrypted database. *Commun. ACM 55*, 9 (Sept. 2012), 103–111.
35. Popa, R.A., Stark, E., Helfer, J., Valdez, S., Zeldovich, N., Kaashoek, M.F. and Balakrishnan, H. Building Web applications on top of encrypted data using Mylar. In *Proceedings of the 11th Symposium on Networked Systems Design and Implementation* (2014), 157–172.
36. Rubio-Gonzàlez, C. Precimonious: Tuning assistant for floating-point precision. In *Proceedings of the International Conf. on High Performance Computing, Networking, Storage and Analysis.* ACM, 2013, 27.
37. Reubel, O. WarpIV: In situ visualization and analysis of ion accelerator simulations. *IEEE Computer Graphics and Applications 36*, 3 (2016), 22–35.
38. Ramakrishnan, L., Poon, S., Hendrix, V., Gunter, D., Pastorello, G.Z. and Agarwal, D. Experiences with user-centered design for the Tigres workflow API. In *Proceedings of 2014 IEEE 10th International Conference on e-Science*, vol 1. IEEE, 290–297.
39. Singer A. Tempting fate. *;login: 30*, 1 (Feb. 2005), 27–30.
40. Schneier, B. and Kelsey, J. Automatic event-stream notarization using digital signatures. In *Proceedings of the 4th International Workshop on Security Protocols*. Springer, 1996, 155–169.
41. Sommer, R. and Paxson, V. Outside the closed world: On using machine learning for network intrusion detection. In *Proceedings of the 31st IEEE Symposium on Security and Privacy*, Oakland, CA, May 2010.
42. Stoll, C. Stalking the wily hacker. *Commun. ACM 31*, 5 (May 1988), 484–497.
43. Skinner, D., Wright, N., Fürlinger, K., Yelick, K.A. and Snavely, A. Integrated Performance Monitoring; http://ipm-hpc.sourceforge.net/.
44. Wallace, D. Compute node Linux: New frontiers in compute node operating systems. *Cray User Group*, 2007.
45. Whitlock, B., Favre, J.M. and Meredith, J.S. Parallel in situ coupling of simulation with a fully featured visualization system. In *Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization*, 2011, 101–109.
46. Wisniewski, R.W., Inglett, T., Keppel, P., Murty, R. and Riesen, R. mOS: An architecture for extreme-scale operating systems. In *Proceedings of the 4th International Workshop on Runtime and Operating Systems for Supercomputers*. ACM, 2014.
47. Whalen, S., Peisert, S. and Bishop, M. Network-theoretic classification of parallel computation patterns. In *Proceedings of the First International Workshop on Characterizing Applications for Heterogeneous Exascale Systems* (Tucson, AZ, June 4, 2011).
48. Whalen, S., Peisert, S. and Bishop, M. Multiclass Classification of Distributed Memory Parallel Computations. *Pattern Recognition Letters 34*, 3 (Feb. 2013), 322–329.
49. Yosinski, J., Clune, J., Fuchs, T. and Lipson, H. Understanding neural networks through deep visualization. In *Proceedings of the Deep Learning Workshop, International Conference on Machine Learning*, 2015.
50. Yelick, K. A Superfacility for Data Intensive Science. Advanced Scientific Computing Research Advisory Committee, Washington, DC, Nov. 8, 2016; http://science.energy.gov/~/media/ascr/ascac/pdf/meetings/201609/Yelick_Superfacility-ASCAC_2016.pdf.

Sean Peisert (sppeisert@lbl.gov) is Staff Scientist at Lawrence Berkeley National Laboratory, Chief Cybersecurity Strategist at CENIC, and an associate adjunct professor at the University of California, Davis.

Watch the author discuss his work in this exclusive *Communications* video. https://cacm.acm.org/videos/security-in-high-performance-computing-environments

research highlights

DOI:10.1145/3068774

To view the accompanying paper, visit doi.acm.org/10.1145/3068776 rh

Technical Perspective
A Gloomy Look at the Integrity of Hardware

By Charles (Chuck) Thacker

SINCE THE INVENTION of the integrated circuit, the complexity of the devices and the cost of the facilities used to build them have increased dramatically. The first fabrication facility with which I was associated was built at Xerox PARC in the mid-1970s at a cost of approximately $15M ($75M today). Today, the cost of a modern fab is approximately $15B. This cost is justified by the fact that today's chips are much more complex than in earlier times. The number of layers involved has grown to over 100, and the tolerances involved are approaching atomic dimensions.

The high cost of a fab means that in order to be cost-effective, it must be fully loaded. This has led to "silicon foundries," which build chips for a variety of "fabless" semiconductor companies based on a set of physical design libraries supplied by the foundry. Carver Mead and Lynn Conway in their seminal 1980 "Introduction to VLSI Systems" initially proposed this concept, but the Taiwan Semiconductor Company (TSMC), founded in 1987, changed what had been an academic exercise into an industrial norm. Today, a few large fabs throughout the world dominate this business.

Over the last two decades, integrated circuit design has diverged into two specialties: (1) Architectural and logical design and device layout, done by a design house, with (2) mask generation and device fabrication done by a foundry. To ensure the foundry has done its job correctly, the design house relies on extensive testing to verify that devices meet their specifications.

The following paper assumes the foundry (or other parties involved in the low levels of fabrication) is malicious, and can modify the design they receive to produce a device that can later be used for malice. Their attack employs a very small Trojan circuit included in an otherwise correct design. The Trojan awaits the chip's deployment, and it may then be triggered by an external software attack. When triggered, the chip's normal function is subverted by the attacker. In the A2 implementation, the trigger is used to elevate the privilege of a user-mode program. The authors argue that the simplicity of the Trojan and its use of analog circuitry make it difficult to detect, even with enhanced levels of testing. They go to considerable lengths to verify their approach, including extensive simulation and actual fabrication of a processor in a modern silicon process. On the actual hardware, the Trojan operated as expected.

As technologists, *technical* solutions are what we do best. In the case of the attack proposed by the authors, a technical defense seems problematic.

Is this realistic? Certainly no foundry wants to compromise its business model by being identified as untrustworthy.

As I was preparing this Technical Perspective, the Dyn/Mirai DDoS attack occurred. Apparently, the attack used a large number of IoT devices (DVRs and webcams) as a botnet, which targeted a major DNS server. This is approximately what the authors of the following paper describe, although the attack was done by exploiting the lack of security in the target's software, rather than by adding hardware. The reports seem to indicate the bot devices were easily compromised, using default passwords that could not be changed, and the devices were not designed to be updated in the field. While the security provided by IoT devices will surely improve, the authors argue that the introduction of small Trojans by untrusted fabrication facilities will remain a problem for which technical solutions appear elusive.

As technologists, *technical* solutions to problems are what we do best. In the case of the attack proposed by the authors, a technical defense seems problematic. We do, however, have examples from other fields that might be promising. The A2 Trojan assumes an *untrusted* fabrication facility. While it might not be possible to do all future fabrication in trusted facilities, using a third party trusted by both the fab and its customers to monitor the behavior of the fab seems plausible. The job of the third party is to certify the proper behavior of the fab. Trusted third parties are widely used in areas ranging from financial contracts to nuclear treaty compliance. "Trust but verify" was used during the Cold War to describe this relationship.

The authors have a lot of experience with attacks on digital logic, and do a good job of explaining previous work in the area. The paper is definitely worth reading carefully, as it covers an area that will likely become much more important in an increasingly technology-dependent world. C

Charles Thacker, computing pioneer and recipient of the 2009 ACM A.M. Turing Award, passed away in June 2017, soon after this Technical Perspective was written.

DOI:10.1145/3068776

Exploiting the Analog Properties of Digital Circuits for Malicious Hardware

By Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, and Dennis Sylvester

Abstract

While the move to smaller transistors has been a boon for performance it has dramatically increased the cost to fabricate chips using those smaller transistors. This forces the vast majority of chip design companies to trust a third party—often overseas—to fabricate their design. To guard against shipping chips with errors (intentional or otherwise) chip design companies rely on post-fabrication testing. Unfortunately, this type of testing leaves the door open to malicious modifications since attackers can craft attack triggers requiring a sequence of unlikely events, which will never be encountered by even the most diligent tester. In this paper, we show how a fabrication-time attacker can leverage analog circuits to create a hardware attack that is small (i.e., requires as little as one gate) and stealthy (i.e., requires an unlikely trigger sequence before affecting a chip's functionality). In the open spaces of an already placed and routed design, we construct a circuit that uses capacitors to siphon charge from nearby wires as they transit between digital values. When the capacitors are fully charged, they deploy an attack that forces a victim flip-flop to a desired value. We weaponize this attack into a remotely controllable privilege escalation by attaching the capacitor to a controllable wire and by selecting a victim flip-flop that holds the privilege bit for our processor. We implement this attack in an OR1200 processor and fabricate a chip. Experimental results show that the purposed attack works. It eludes activation by a diverse set of benchmarks and evades known defenses.

1. INTRODUCTION

The trend toward smaller transistors in integrated circuits, while beneficial for higher performance and lower power, has made fabricating a chip expensive. For example, it costs 15% more to set up the fabrication line for each successive process node and by 2020 it is expected that setting up a fabrication line for the smallest transistor size will require a $20 billion upfront investment.[18] To amortize the cost of fabrication development, most hardware companies outsource fabrication.

Outsourcing of chip fabrication opens up hardware to attack. These hardware attacks can evade software checks because software must trust hardware to faithfully implement the instructions.[6, 12] Even worse, if there is an attack in hardware, it can contaminate all layers of a system that depend on the hardware and violates high-level security policies correctly implemented by software.

The most pernicious fabrication-time attack is the dopant-level Trojan.[2, 10] Dopant-level Trojans convert trusted circuitry into malicious circuitry by changing the dopant ratio on the input pins to victim transistors. Converting existing circuits makes dopant-level Trojans very difficult to detect since there are no added or removed gates or wires. In fact, detecting dopant-level Trojans requires a complete chip delayering and comprehensive imaging with a scanning electron microscope.[17] However, this elusiveness comes at the cost of expressiveness. Dopant-level Trojans are limited by existing circuits, making it difficult to implement sophisticated attack triggers.[10] The lack of a sophisticated trigger means that dopant-level Trojans are more detectable by post-fabrication functional testing. Thus, dopant-level Trojans represent an extreme on a trade-off space between detectability during a physical inspection and detectability during testing.

To defend against malicious hardware inserted during fabrication, researchers have proposed two fundamental defenses: (1) using side-channel information (e.g., power and temperature) to characterize acceptable behavior in an effort to detect anomalous (i.e., malicious) behavior,[1, 7, 13, 15] and (2) adding sensors to the chip that directly measure and characterize features of the chip's behavior (e.g., signal propagation delay) in order to identify dramatic changes in those features (presumably caused by activation of a malicious circuit).[3, 8, 11] Using side channels as a defense works well against large Trojans added to purely combinational circuits where it is possible to test all inputs and there exists a reference chip to compare against. While this accurately describes most existing fabrication-time attacks, we show that it is possible to implement a stealthy and powerful processor attack using only a single added gate without affecting features measured by existing on-chip sensors.

We create a new fabrication-time attack that is controllable, stealthy, and small, which borrows the idea of counter-based triggers commonly used to hide design-time malicious hardware[19, 20] and adapt it to fabrication-time. Based on analog behaviors, the attack replaces the hundreds of gates required by conventional counter-based digital triggers with analog components—a capacitor and a few transistors wrapped up in a single gate.

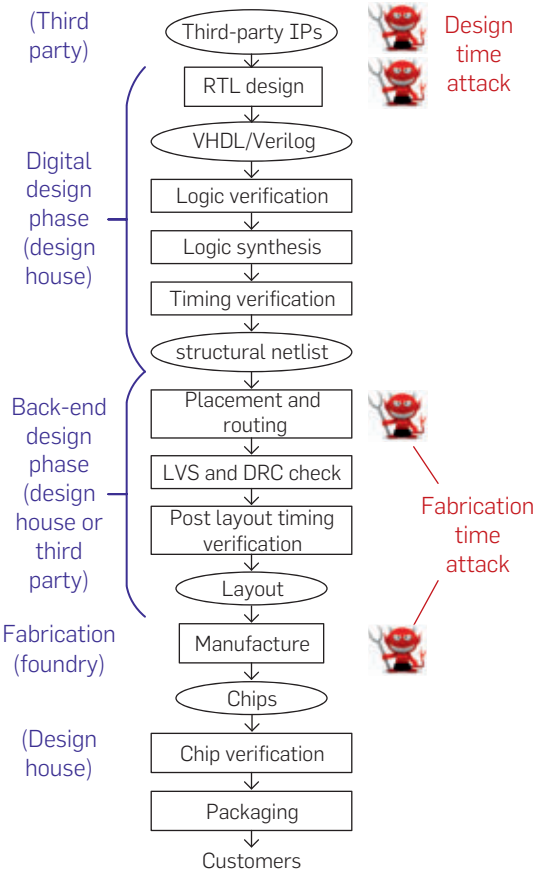
The original version of this paper is entitled "A2: Analog Malicious Hardware" and was published in 2016 IEEE International Symposium on Security and Privacy.

This paper presents three contributions. (1) We design and implement the first fabrication-time processor attack that mimics the triggered attacks often added during design time. As a part of our implementation, we are the first to show how a fabrication-time attacker can leverage the empty space common in chip layouts to implement malicious circuits, (2) We show how an analog attack can be much smaller and more stealthy than its digital counterpart. Our attack diverts charge from unlikely signal transitions to implement its trigger, so it is invisible to all known side-channel defenses. Additionally, as an analog circuit, our attack is under the digital layer and missed by functional verification performed on the hardware description language, and (3) We fabricate an openly malicious processor and then evaluate the behavior of our fabricated attacks across many chips and changes in environmental conditions. We compare these results to Simulation Program with Integrated Circuit Emphasis (SPICE) simulation models.

2. BACKGROUND AND THREAT MODEL

The typical design and fabrication process of integrated circuits is as shown in Figure 1. See Rostami[16]. This process often involves collaboration between different parties all over the world and each step is likely done by different teams even if they are in the same company. Therefore, the designs are vulnerable to malicious attacks by rogue engineers involved in any of the above steps.

Figure 1. Typical IC design process with commonly-research threat vectors highlighted in red. The blue text and brackets highlights the party in control of the stage(s).



The design house implements the specification for the chip's behavior in some Hardware Description Language (HDL). Once the specification is implemented in an HDL and that implementation has been verified, the design is passed to a back-end house, which places and routes the circuit.

Conventional digital Trojans can only be inserted in design phase and are easier to be detected by design phase verifications. Fabrication-time attacks inserted in back-end and fabrication phases can evade these defenses. Since it is strictly more challenging to implement attacks at the fabrication phase due to limited information and ability to modify the design compared to the back-end phase, we focus on that threat model for our attack.

The attacker starts with a Graphic Database System II (GDSII) file that is a polygon representation of the completely laid-out and routed circuit. Our threat model assumes that the delivered GDSII file represents a perfect implementation—at the digital level of abstraction—of the chip's specification. This is very restrictive as it means that the attacker can only modify existing circuits or—as we are the first to show in this paper—add attack circuits to open spaces in the laid-out design. The attacker can *not* increase the dimensions of the chip or move existing components around. This restrictive threat model also means that the attacker must perform some reverse engineering to select viable victim flip-flops and wires to tap. After the untrusted fabrication house completes fabrication, it sends the fabricated chips off to a trusted party for post-fabrication testing. Our threat model assumes that the attacker has no knowledge of the test cases used for post-fabrication testing. Such a model dictates the use of a sophisticated trigger to hide the attack.

3. ATTACK METHODS

A hardware attack is composed of a trigger and a payload. The trigger monitors wires and state within the design and activates the attack payload under very rare conditions such that the attack stays hidden during normal operation and testing. Previous research has identified that evading detection is a critical property for hardware Trojans designers.[5] Evading detection involves more than just avoiding attack activation during normal operation and testing, it includes hiding from visual/side-channel inspection. There is a trade-off at play between the two in that the more complex the trigger (i.e., the better that it hides at run time), the larger the impact that trigger has on the surrounding circuit (i.e., the worse that it hides from visual/side-channel inspection).

We propose A2, a fabrication-time attack that is small, stealthy, and controllable. To achieve these outcomes, we develop trigger circuits that operate in the analog domain. The circuits are based on charge accumulating on a capacitor from infrequent events inside the processor. If the charge-coupled infrequent events occur frequently enough, the capacitor will fully charge and the payload is activated to deploy a privilege escalation attack. Our analog trigger is similar to the counter-based triggers often used in digital triggers, except that using the capacitor has the advantage of a natural reset condition due to leakage. Compared

to traditional digital hardware Trojans, the analog trigger maintains a high level of stealth and controllability, while dramatically reducing the impact on area, power, and timing due to the attack. An added benefit of a fabrication-time attack compared to a design-time attack (when digital-only triggers tend to get added) is that it has to pass through fewer verification stages.

3.1. Single stage trigger circuit

Based on our threat model, the high-level design objectives of our analog trigger circuit are as follows:

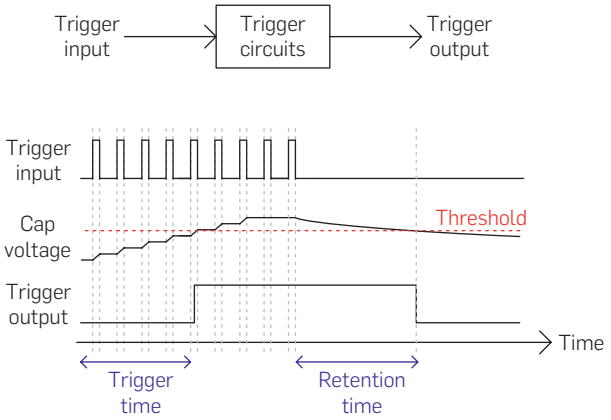
1. Functionality: The trigger circuit must be able to detect toggling events of a target victim wire similar to a digital counter and the trigger circuit should be able to reset itself if the trigger sequence is not completed in a timely manner.
2. Small area: The trigger circuit should be small enough to be inserted into the empty space of an arbitrary finished chip layout. Small area overhead also implies better chance to escape detection.
3. Low power: The trigger circuit is constantly monitoring the victim signals, therefore its power consumption must be minimized to hide within the normal fluctuations of the entire chip's power consumption.
4. Negligible timing perturbation: The added trigger circuit must not affect the timing constraints for normal operation and its timing perturbations should not be easily separable from the noise common to path delays.
5. Standard cell compatibility: Since all digital designs are based on standard cells with fixed cell height, the analog trigger circuit must fit into the height and only use the lowest metal layer for routing.[a] These requirements are important for insertion into existing chip layout and makes the trojan more difficult to detect in fabricated chips.

To achieve these design objectives, we propose an attack based on charge accumulation inside capacitors. A capacitor performs analog integration of charge from a victim wire while at the same time being able to reset itself through leakage current. A behavior model of capacitor based trigger circuits comprises charge accumulation and leakage as shown in Figure 2.

Every time the victim wire that feeds the trigger circuit's capacitor toggles, the capacitor increases in voltage by some ΔV. After a number of toggles, the capacitor's voltage exceeds a predefined threshold voltage and enables the trigger's output—deploying the attack payload. The time it takes to activate the trigger is defined as *trigger time* (Figure 2).

On the other hand, leakage current exists all the time and it dumps charge from the trigger circuit's capacitor. The attacker can design the capacitor's leakage to be weaker than its accumulation when the trigger input is active. On the other hand, when the trigger input is inactive, leakage gradually reduces the capacitor's voltage, eventually disabling an already activated trigger. This mechanism ensures that the attack is not expressed when no intentional attack happens. The time it takes to reset trigger output after trigger input stops is defined as *retention time*.

Figure 2. Behavior model of proposed analog trigger circuit.



Because of leakage, a minimum toggling frequency must be reached to successfully trigger the attack. At the minimum frequency, charge added in each cycle equals charge leaked away. *Trigger time* and *retention time* are the two main design metrics in the analog trigger circuits that we can make use of to create flexible trigger conditions and more complicated trigger patterns as discussed in Section 3.2. A stricter triggering condition (i.e., faster toggling rate and more toggling cycles) reduces the probability of a false trigger during normal operation or testing, but non-idealities in circuits and process, temperature and voltage variations can cause the attack to fail—impossible to trigger or trivial to accidentally trigger—for some chips. As a result, a trade-off should be made between a reliable attack that can be expressed in every chip and a more stealthy attack that can only be triggered for certain chips under certain conditions.

The conventional current-based charge pump is not suitable for the attack due to area and power constraints. A new charge pump circuit based on charge sharing is specifically designed for the attack purpose as shown in Figure 3. During the negative phase of *Clk*, *Cunit* is charged to *VDD*. Then during positive phase of *Clk*, the two capacitors are shortened together, causing the two capacitors to share charges. After charge sharing, final voltage of the two capacitors is the same and ΔV on *Cmain* is as,

$$\Delta V = \frac{Cunit \times (VDD - V_0)}{Cunit + Cmain}$$

where V_0 is initial voltage on *Cmain* before the transition happens. We can achieve different *trigger time* by sizing the two capacitors. The capacitor keeps leaking over time and finally ΔV equals the voltage drop due to leakage, which sets the maximum capacitor voltage.

A transistor-level schematic of the proposed analog trigger is as shown in Figure 4. *Cunit* and *Cmain* are implemented

[a] Several layers of metal wires are used in modern CMOS technologies to connect cells together, lower level metal wires are closer to transistors at bottom for short interconnections, while higher metal layers are used for global routing.

with Metal Oxide Semiconductor (MOS) caps. *M*0 and *M*1 are the two switches as shown in Figure 3. A detector is used to compare cap voltage with a threshold voltage and can be implemented by inverters or Schmitt triggers. An inverter has a switching voltage depending on its sizing and when the capacitor voltage is higher than the switching voltage, the output is 0; otherwise, the output is 1. A Schmitt trigger is an inverter with hysteresis. It has a large threshold when input goes from low to high and a small threshold when input goes from high to low. The hysteresis is beneficial for our attack because it extends both *trigger time* and *retention time*. To balance the leakage current through *M*0 and *M*1, an additional leakage path to ground (NMOS *M*2 as shown in Figure 4) is added to the design.

A SPICE simulation waveform is as shown in Figure 5 to illustrate the operation of our analog trigger circuit after optimization. The operation is same as the behavioral model that we proposed as shown in Figure 2, allowing us to use the behavior model for system-level attack design.

Figure 3. Design concepts of analog trigger circuit based on capacitor charge sharing.

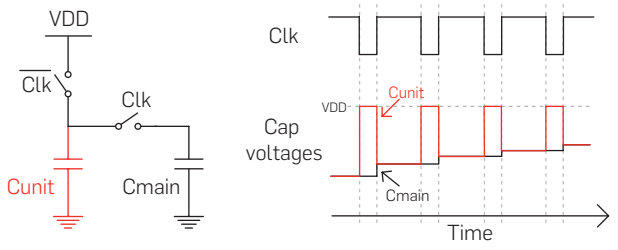


Figure 4. Transistor-level schematic of analog trigger circuit.

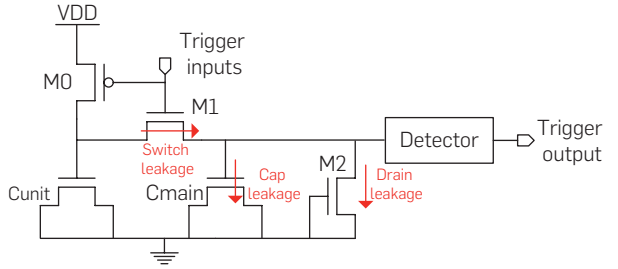
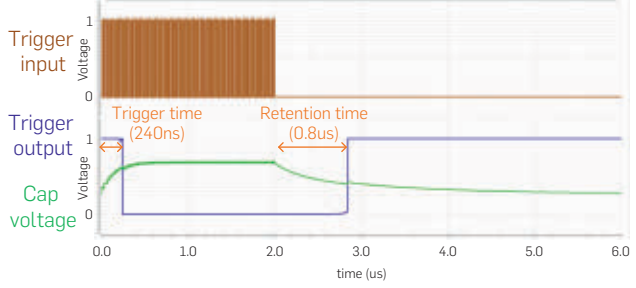


Figure 5. SPICE simulation waveform of analog trigger circuit.



3.2. Multi-stage trigger circuit

The one-stage trigger circuit described in the previous section takes only one victim wire as an input. Using only one trigger input limits the attacker in two ways: (1) Because fast toggling of one signal for tens of cycles triggers the single stage attack, there is still a chance that normal operations or certain benchmarks can expose the attack, and (2) Certain instructions are required to create fast toggling of a single trigger input and there is not much room for a flexible and stealthy attack program.

We note that an attacker can make a logical combination of two or more single-stage trigger outputs to create a variety of more flexible multi-stage analog triggers. Basic operations to combine two triggers include *AND* and *OR*. When analyzing the behavior of logic operations on single stage trigger output, it should be noted that the single-stage trigger outputs 0 when triggered. Thus, for *AND* operation, the final trigger is activated when either *A* or *B* triggers fire. For *OR* operation, the final trigger is activated when both *A* and *B* triggers fire. It is possible for an attacker to combine these simple *AND* and *OR*-connected triggers into an arbitrarily complex multi-level multi-stage trigger.

3.3. Triggering the attack

For A2, the payload design is independent of the trigger mechanism, so our proposed analog trigger is suitable for various payloads to achieve different attacks. Since the goal of this work is to achieve a Trojan that is nearly invisible while providing a powerful foothold for a software-level attacker, we couple our analog triggers to a privilege escalation attack,[9] which provides maximum capabilities to an attacker. We propose a simple design to overwrite security critical registers directly by adding one *AND/OR* gate to asynchronous set or reset pins of the registers. These reset/set pins are specified in original designs for processor reset. These reset signals are asynchronous with no timing constraints so that adding one gate into the reset signal of one register does not affect functionality or timing constraints of the design. Because there are no timing constraints on asynchronous inputs, the payload circuit can be inserted manually after final placement and routing in a manner consistent with our threat model.

3.4. Selecting victims

It is important that the attacker validate their choice of victim signal. This requires verifying that the victim wire has low baseline activity and its activity level is controllable given the expected level of access of the attacker. To validate that the victim wire used in A2 has a low background activity, we use benchmarks from the MiBench embedded systems benchmark suite. For cases where the attacker does not have access to such software or the attacked processor will see a wide range of use, the attacker can follow A2's example and use a multi-stage trigger with wires that toggle in a mutually-exclusive fashion and require inputs that are unlikely to be produced using off-the-shelf tools (e.g., GNU Compiler Collection (GCC)).

Validating that the victim wire is controllable requires that the attacker reason about their expected level of access to the end user system for the attacked processor. In A2, we

assume that the attacker can load and execute any unprivileged instruction. This allows us to create hand-crafted assembly sequences that activate the attack. This model works for attackers that have an account on the system, attackers in a virtual machine, or even attackers that can convince users to load code.

4. IMPLEMENTATION

To experimentally verify A2, we implement and fabricate an open source processor with the proposed analog Trojans inserted in 65nm General Purpose Complementary Metal-Oxide-Semiconductor (CMOS) technology. Multiple attacks are implemented in the chip. One set of attacks are Trojans aimed at exposing A2's end-to-end operation, while the other set of attacks are implemented outside the processor, directly connected to Input/Output (IO) pins so that we can investigate trigger behavior directly.

4.1. Attacking a real processor

We implemented an open source OR1200 processor[14] to verify our A2 attack including software triggers, analog triggers and payload. The OR1200 Central Processing Unit (CPU) is an implementation of the 32-bit OR1K instruction set with a five stage pipeline. The implemented system in silicon consists of a OR1200 core with 128B instruction cache and an embedded 128KB main program memory connected through a Wishbone bus. The OR1K instruction set specifies the existence of a privileged register called the Supervision Register (SR). The SR contains bits that control how the processor operates (e.g., Memory Management Units (MMU) and caches enabled) and flags (e.g., carry flag). One particular bit is interesting for security purposes; SR[0] controls the privilege mode of user, with 0 denoting user mode and 1 denoting supervisor mode. By overwriting the value of this register, an attacker can escalate a user mode process to supervisor mode as a backdoor to deploy various high-level attacks.[5, 9] Therefore, we make the payload of our attack setting this bit in the SR to 1 to give a user mode process full control over the processor.

Our analog trigger circuits require trigger inputs that can have a high switching activity under certain (attacker) programs but are almost inactive during testing or common case operation so that the Trojan is not exposed. To search for suitable victim wires as trigger inputs, we run a series of programs from MiBench (see Section 5) on the target processor in an HDL simulator, capturing the toggling rates of all wires. The result shows that approximately 3% of total wires have nearly zero activity rate, which provides a wide range of options for an attacker. The target signals must also be easy to control by attack programs. In our attack, we select divide by zero flag signal as the trigger for the one-stage attack, because it is unlikely for normal programs to continuously perform division-by-zero while it is simple for an attacker to deliberately perform such operations in a tight loop. For the two-stage trigger, we select wires that report whether the division was signed or unsigned as trigger inputs. The attack program alternatively switches the two wires by performing signed, then unsigned division, until both analog trigger circuits are activated, deploying the attack payload.

Triggering the attack in usermode-only code is only the first part of a successful attack. For the second part, the attacker must be able to verify that the triggering software works—without risk of alerting the operating system. To check whether the attack is successful, we take advantage of a special feature of some registers on the OR1200: some privileged registers are able to be read by user mode code, but the value reported has some bits redacted. We use this behavior to let the attacker's code know whether it gets privileged access to the processor or not.

4.2. Analog activity trigger

We implement both the one-stage and two-stage trigger circuits in 65nm GP CMOS technology based on SPICE simulations. Both trigger circuits are inserted into the processor to demonstrate the attack.

Implementation in 65nm GP technology. For prototype purposes, we optimize the trigger circuit towards a reliable version and building a reliable circuit under process, temperature, and voltage (PVT) variations is always more challenging than only optimizing for a certain PVT range—that is, we construct our attacks so that they work in all fabricated processors at all corner-case environments. 65nm CMOS technology is not a favorable technology for our attack because the gate oxide is thinner than older technologies due to dimension scaling and also thinner than latest technologies because high-κmetal gate techniques now being employed to reduce gate leakage. However, through careful sizing, it's still possible to design a circuit robust across PVT variations, but this requires trading-off *trigger time* and *retention time*.

To reduce gate leakage, another solution is to use thick oxide transistors commonly used in IO cells as the MOS cap for *Cmain*, which shows negligible gate leakage. This option provides larger space for the configuration of *trigger time* and *retention time* but requires larger area due to design rules. Trigger circuit using IO device is implemented for the two-stage attack and the one without IO device is used for the one-stage attack in the system.

Inserting A2 into existing chip layouts. Since A2's analog trigger circuit is designed to follow sizing and routing constraints of standard cells and has the area of a single standard cell, inserting the trigger circuit to the layout at fabrication time is not complicated. In typical placement and routing cases, around 60% to 70% of total area is used for standard cells, otherwise routing can not complete due to routing congestions (our chip is more challenging to attack as it has 80% area utilization). Therefore, in any layout of digital designs, empty space exists. This empty space presents an opportunity for attackers as they can occupy the free space with their own malicious circuit. In our case, we require as little space as one cell. There are four steps to insert a trigger into the layout of a design:

The first step is to locate the signals chosen as trigger inputs and the target registers to attack. The insertion of A2 attack can be done at both back-end and fabrication stage. Our threat model focuses on the fabrication stage because it is significantly more challenging and implies a more stealthy attack over compared to attack at back-end stage attacks. The back-end stage attacker has access to the netlist

of the design, so locating the desired signal is trivial. But an attack inserted at back-end stage can still be discovered by SPICE simulation and layout checks, though the chance is extremely low if no knowledge about the attack exists. In contrast, fabrication time attacks can only be discovered by post-silicon testing, which is believed to be very expensive and difficult to find small Trojans. To insert an attack during chip fabrication, some insights about the design are needed, which can be extracted from layout through physical verification tools and digital simulations or from a co-conspirator involved in the design phase.

The next step is to find empty space around the victim wire and insert the analog trigger circuit. Unused space is usually automatically filled with filler cells or capacitor cells by placement and routing tools. Removing these cells will not affect the functionality or timing.

To insert the attack payload circuit, the reset wire needs to be cut as discussed in Section 3.3. It has been shown that timing of reset signal is flexible, so the *AND* or *OR* gate only need to be placed somewhere close to the reset signal. Because the added gates can be a minimum strength cell, their area is small and finding space for them is trivial.

The last step is to manually do the routing from trigger input wires to analog trigger circuit and then to the payload circuits. There is no timing requirement on this path so that the routing can go around existing wires at same metal layer (jogging) or jump over existing wires by going to another metal layer (jumping). If long and high metal wires become a concern of the attacker due to potentially easier detection, repeaters (buffers) can be added to break long wire into small sections. Furthermore, it is possible that the attacker can choose different trigger input wires and/or payload according to the existing layout of the target design.

In our OR1200 implementation, inserting the attack following the steps above is trivial, even with the design's 80% area utilization. Routing techniques including jogging and jumping are used, but such routing approach is very common for automatic routing tools so the information leaked by such wires is limited.

Side-channel information. For the attack to be stealthy and defeat existing protections, the area, power and timing overhead of the analog trigger circuit should be minimized. High accuracy SPICE simulation is used to characterize power and timing overhead of implemented trigger circuits. Comparisons with several variants of *NAND*2 and *DFlip–Flop* standard cells from commercial libraries are summarized in Table 1. The area of the trigger circuit not using IO device is similar to a X4 strength *DFlip–Flop*. Using an IO device increases trigger circuit size significantly, but area is still similar to the area of two standard cells, which ensures it can be inserted into empty space in final design layout. AC power is the total energy consumed by the circuits when input changes, the power numbers are simulated with SPICE on a netlist including extracted parasitics. Standby power is the power consumption of the circuits when inputs are static, which comes from leakage currents of CMOS devices.

After inserting A2, post-layout simulation with extracted parasitics shows that the extra delay of victim wires is 1.2ps on average, which is only 0.33% of 4ns clock period and well below the process variation and noise range. In practice, such delay difference is nearly impossible to measure, unless a high-resolution time to digital converter is included on chip, which is impractical due to its large area and power overhead.

Comparison to digital-only attacks. If we look at a previously proposed, digital only and smallest implementation of a privilege escalation attack,[5] it requires 25 gates and 80μm^2 while our analog attack requires as little as one gate for the same effect. Our attack is also much more stealthy as it requires dozens of consecutive rare events, where the other attack only requires two. We also implement a digital only, counter-based attack that aims to mimic A2. The digital version of A2 requires 91 cells and 382μm^2, almost two orders-of-magnitude more than the analog counterpart. These results demonstrate how analog attacks can provide attackers the same power and control as existing digital attacks, but much more difficult to catch.

5. EVALUATION

We perform all experiments with our fabricated 2.1mm^2 malicious OR1200 processor as shown in Figure 6. Figure 6 also marks the locations of A2 attacks, with two levels of zoom to aide in understanding the challenges of identifying A2 in a sea of non-malicious logic. In fact, A2 occupies less than 0.08% of the chip's area. Our fabricated chip contains two sets of attacks: the first set of attacks are one and two-stage triggers baked-in to the processor that we use to assess the end-to-end impact of A2. The second set of attacks exist

Table 1. Comparison of area and power between our implemented analog trigger circuits and commercial standard cells in 65nm GP CMOS technology.

Function	Drive strength	Width†	AC power†	Standby power†
$NAND_2$	X1	1	1	1
$NAND_2$	X4	3	3.7	4.1
$NAND_2$	X8	5.75	7.6	8.1
DFF with Async reset	X1	6	12.7	2.6
DFF with Async reset	X4	7.75	21.8	7.2
DFF with Async set and reset	X1	7.5	14.5	3.3
DFF with Async set and reset	X4	8.75	23.6	8.1
Trigger w/o IO device	–	8	7.7	2.2
Trigger w/ IO device	–	13.5	0.08	0.08

* DFF stands for D Flip Flop. † Normalized values.

outside of the processor and are used to fully characterize A2's operation.

We use the testing setup as shown in Figure 7 to evaluate our attacks' response to changing environmental conditions and a variety of software benchmarks. The chip is packaged and mounted on a custom testing board to interface with a PC. Through a custom scan chain, we can load programs into the processor's memory and also check the values of the processor's registers. The system's clock is provided by an on-chip 240MHz clock generator at the nominal condition (1V supply voltage and 25°C).

5.1. Does the attack work?

To prove the effectiveness of A2, we evaluate it from two perspectives. One is a system evaluation that explores the end-to-end behavior of our attack by loading attack-triggering programs on the processor, executing them in user mode, and verifying that after executing the trigger sequence, they have escalated privilege on the processor. The other perspective seeks to explore the behavior of our attacks by directly measuring the performance of the analog trigger circuit, the most important component in our attack, but also the most difficult aspect of our attack to verify using simulation.

System attack. Malicious programs described in Section 4.1. are loaded to the processor and then we check the target register values. In the program, we initialize the target registers *SR*[0] (the mode bit) to user mode (i.e., 0) and *SR*[1] (a free register bit that we can use to test the two-stage trigger) to 1. When the respective trigger deploys the attack, the single-stage attack will cause *SR*[0] to suddenly have a 1 value, while the two-stage trigger will cause *SR*[1] to have a 0 value—the opposite of their initial values. Because our attack relies on analog circuits, environmental aspects dictate the performance of our attack. Therefore, we test the chip at six temperatures from –25°C to 100°C to evaluate the robustness of our attack. Measurement results confirm that both the one-stage and two-stage attacks in all ten tested chips successfully overwrite the target registers at all temperatures.

Analog trigger circuit measurement results. Figure 8 shows the measured distribution of *retention time* and trigger cycles at three different trigger toggling frequencies across ten chips. The results show that our trigger circuits have a regular behavior in the presence of real-world manufacturing variances, confirming SPICE simulation results. *retention time* at the nominal condition (1V supply voltage and 25°C) is around 1μs for the trigger with only core devices and 5μs for attacks constructed using IO devices. It is verified that the number of cycles to trigger attack for both trigger circuits (i.e., with and without IO devices) are very close in chip measurements and SPICE simulations. The results indicate that SPICE is capable of providing results of sufficient accuracy for these unusual attack circuits.

To verify the implemented trigger circuits are robust across voltage and temperature variations (as SPICE simulation suggests), we characterize each trigger circuit under different supply voltage and temperature conditions. We

Figure 6. Die micrograph of analog malicious hardware test chip with a zoom-in layout of inserted A2 trigger.

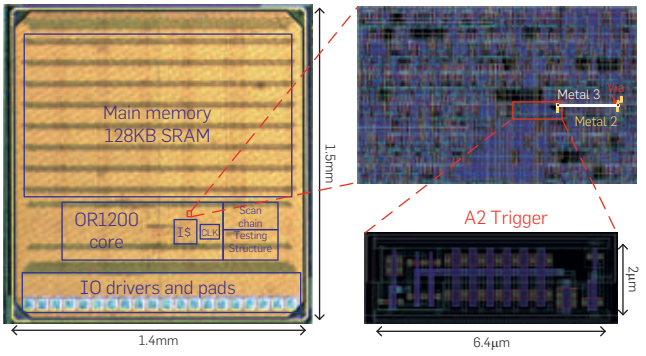


Figure 7. Testing setup for test chip measurement.

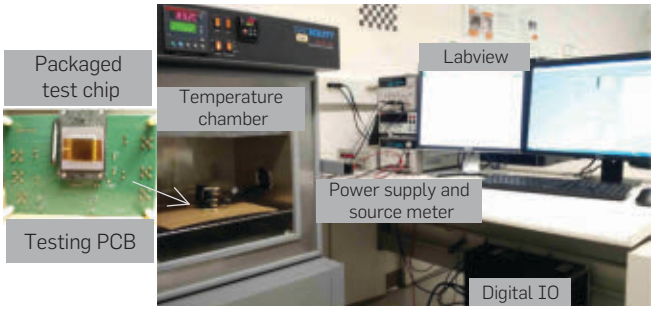
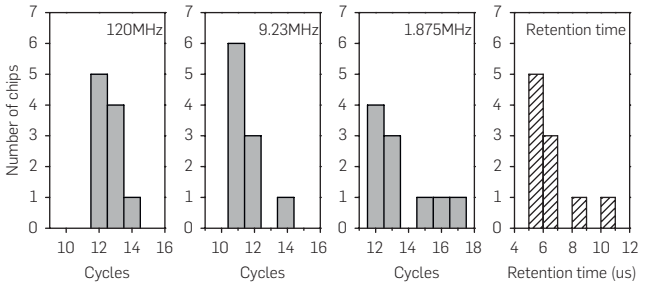
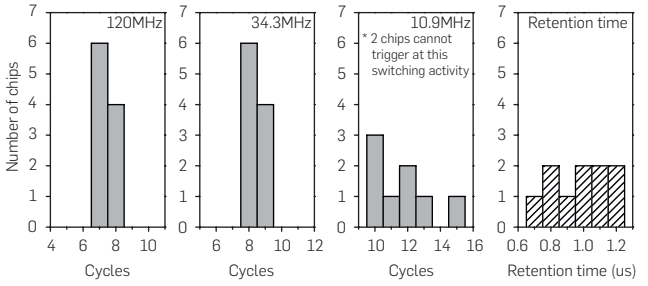


Figure 8. Measured distribution of retention time and trigger cycles under different trigger input divider ratios across 10 chips at nominal 1V supply voltage and 25°C.



(a) Distribution of analog trigger circuit using IO device



(b) Distribution of analog trigger circuit using only core device

confirmed that the trigger circuit can be activated when the victim wire toggles between 0.46MHZ and 120MHz, the supply voltage varies between 0.8V and 1.2V, and the ambient temperature varies between –25°C and 100°C.

As expected, different conditions yield different minimum toggling rates to activate the trigger. Temperature has a stronger impact than voltage on the trigger condition because of leakage current's exponential dependence on temperature. At higher temperature, more cycles are required to trigger and higher switching activity is required because leakage from capacitor is larger.

5.2. Is the attack triggered by non-malicious benchmarks?

Another important property for any hardware Trojan is not exposing itself under normal operations. Because A2's trigger circuit is connected only to the trigger input signal, digital simulation of the design is enough to acquire the activity of the signals. However, since we make use of analog characteristics to attack, analog effects should also be considered as potential effects to accidentally trigger the attack. We use MiBench[4] as test bench because it targets the class of processor that best fits the OR1200 and it consists of a set of well-understood applications that are popular benchmarks in both academia and in industry. To validate that A2's trigger avoids spurious activations from a wide variety of software, we select five benchmark applications from MiBench, each from a different class. This ensures that we thoroughly test all subsystems of the processor—exposing likely activity rates for the wires in the processor. Again, in all programs, the victim registers are initialized to opposite states that A2 puts them in when its attack is deployed. The processor runs all five programs at six different temperatures from –25°C to 100°C. Results prove that neither the one-stage nor the two-stage trigger circuit is exposed when running these benchmarks across such wide temperature range.

5.3. Existing protections

Existing protections against fabrication-time attacks are mostly based on side-channel information, for example, power, temperature, and delay. In A2, we only add one gate in the trigger, thus minimizing power and temperature perturbations caused by the attack.

Table 2 summarizes the average power consumption measured when the processor runs our five benchmark programs, at the nominal condition (1V supply voltage and 25°C). Direct measurement of trigger circuit power is infeasible in our setup, so simulation is used as an estimation. Simulated trigger power consumption in Table 1 translates to 5.3nW and 0.5μW for trigger circuits constructed with and without IO devices. These numbers are based on the assumption that trigger inputs keep toggling at 1/4 of the clock frequency of 240MHz, which is the maximum switching activity that our attack program can achieve. In the common case of non-attacking software, the switching activity is much lower—approaching zero—and only lasts a few cycles so that the extra power due to our trigger circuit is even smaller. In our experiments, the power of the attack circuit is orders-of-magnitude less than the normal power fluctuations that occur in a processor while it executes different instructions. Further discussions about possible defenses such as split manufacturing and runtime verifications are presented in our original A2 paper.[21]

Table 2. Power consumption of our test chip running a variety of benchmark programs.

Program	Power (mW)
Standby	6.210
Basic math	23.703
Dijkstra	16.550
FFT	18.120
SHA	18.032
Search	21.960
Single-stage attack	19.505
Two-stage attack	22.575
Unsigned division	23.206

6. CONCLUSION

Experimental results with our fabricated malicious processor show that a new style of fabrication-time attack is possible, which applies to a wide range of hardware, spans the digital and analog domains, and affords control to a remote attacker. Experimental results also show that A2 is effective at reducing the security of existing software, enabling unprivileged software full control over the processor. Finally, the experimental results demonstrate the elusive nature of A2: (1) A2 is as small as a single gate—two orders of magnitude smaller than a digital-only equivalent; (2) attackers can add A2 to an existing circuit layout without perturbing the rest of the circuit; (3) a diverse set of benchmarks fail to activate A2 and (4) A2 has little impact on circuit power, frequency, or delay.

Our results expose two weaknesses in current malicious hardware defenses. First, existing defenses analyze the digital behavior of a circuit using functional simulation or the analog behavior of a circuit using circuit simulation. Functional simulation is unable to capture the analog properties of an attack, while it is impractical to simulate an entire processor for thousands of clock cycles in a circuit simulator—this is why we had to fabricate A2 to verify that it worked. Second, the minimal impact on the run-time properties of a circuit (e.g., power, temperature, and delay) due to A2 suggests that it is an extremely challenging task for side-channel analysis techniques to detect this new class of attacks. We believe that our results motivate a different type of defense, where trusted circuits monitor the execution of untrusted circuits, looking for out-of-specification behavior in the digital domain.

Acknowledgments

This work was supported in part by C-FAR, one of the six SRC STARnet Centers, sponsored by MARCO and DARPA. This work was also partially funded by the National Science Foundation. Any opinions, findings, conclusions, and recommendations expressed in this paper are solely those of the authors.

References

1. Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., Sunar, B. Trojan detection using IC fingerprinting. In *Symposium on Security and Privacy* (S&P, Washington, DC, 2007). IEEE Computer Society, 296–310.
2. Becker, G.T., Regazzoni, F., Paar, C., Burleson, W.P. Stealthy dopant-level hardware Trojans. In *International Conference on Cryptographic Hardware and Embedded Systems* (CHES, Berlin, Heidelberg, 2013). Springer-Verlag, 197–214.
3. Forte, D., Bao, C., Srivastava, A. Temperature tracking: An innovative run-time approach for hardware Trojan detection. In *International Conference on Computer-Aided Design* (ICCAD, 2013). IEEE, 532–539.
4. Guthaus, M.R., Ringenberg, J.S., Ernst, D., Austin, T.M., Mudge, T., Brown, R.B. MiBench: A free, commercially representative embedded benchmark suite. In *Workshop on Workload Characterization* (Washington D.C., 2001). IEEE Computer Society, 3–14.
5. Hicks, M., Finnicum, M., King, S.T., Martin, M.M.K., Smith, J.M. Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically. *USENIX;login 35*, 6 (Dec. 2010), 31–41.
6. Hicks, M., Sturton, C., King, S.T., Smith, J.M. Specs: A lightweight runtime mechanism for protecting software from security-critical processor bugs. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems* (ASPLOS, Istanbul, Turkey, 2015). ACM, 517–529.
7. Jin, Y., Makris, Y. Hardware Trojan detection using path delay fingerprint. In *Hardware-Oriented Security and Trust* (HOST, Washington, DC, 2008). IEEE Computer Society, 51–57.
8. Kelly, S.,Zhang, X., Tehranipoor, M., Ferraiuolo, A. Detecting hardware Trojans using on-chip sensors in an ASIC design. *Journal of Electronic Testing 31*, 1 (Feb. 2015), 11–26.
9. King, S.T., Tucek, J., Cozzie, A., Grier, C., Jiang, W.n., Zhou, Y. Designing and implementing malicious hardware. In *Workshop on Large-Scale Exploits and Emergent Threats*, volume 1 of *LEET* (USENIX Association, Apr. 2008).
10. Kumar, R., Jovanovic, P., Burleson, W., Polian, I. Parametric Trojans for fault-injection attacks on cryptographic hardware. In *Workshop on Fault Diagnosis and Tolerance in Cryptography* (IEEE, FDT, 2014), 18–28.
11. Li, J., Lach, J. At-speed delay characterization for IC authentication and Trojan horse detection. In *Hardware-Oriented Security and Trust* (HOST, Washington, DC, 2008). IEEE Computer Society, 8–14.
12. Li, M.-L., Ramachandran, P., Sahoo, S.K., Adve, S.V., Adve, V.S., Zhou, Y. Understanding the propagation of hard errors to software and implications for resilient system design. In *International Conference on Architectural Support for Programming Languages and Operating Systems* (ASPLOS, Seattle, WA, Mar. 2008). ACM, 265–276.
13. Narasimhan, S., Wang, X., Du, D., Chakraborty, R.S., Bhunia, S. TeSR: A robust temporal self-referencing approach for hardware Trojan detection. In *Hardware-Oriented Security and Trust* (HOST, San Diego, CA, June 2011). IEEE Computer Society, 71–74.
14. OpenCores.org. OpenRISC OR1200 processor.
15. Potkonjak, M., Nahapetian, A., Nelson, M., Massey, T. Hardware Trojan horse detection using gate-level characterization. In *Design Automation Conference*, volume 46 of *DAC* (2009), 688–693.
16. Rostami, M., Koushanfar, F., Rajendran, J., Karri, R. Hardware security: Threat models and metrics. In *Proceedings of the International Conference on Computer-Aided Design* (ICCAD, San Jose, CA, 2013). IEEE Press, 819–823.
17. Sugawara, T., Suzuki, D., Fujii, R., Tawa, S., Hori, R., Shiozaki, M., Fujino, T. Reversing stealthy dopant-level circuits. In *International Conference on Cryptographic Hardware and Embedded Systems* (CHES, New York, NY, 2014). Springer-Verlag, 112–126.
18. S.S. Technology. Why node shrinks are no longer offsetting equipment costs, (online webpage, Oct. 2012).
19. Waksman A., Sethumadhavan, S. Silencing hardware backdoors. In *IEEE Security and Privacy* (S&P, Oakland, CA, May 2011). IEEE Computer Society.
20. Wang, X., Narasimhan, S., Krishna, A., Mal-Sarkar, T., Bhunia, S. Sequential hardware trojan: Side-channel aware design and placement. In *Computer Design (ICCD), 2011 IEEE 29th International Conference on* (IEEE, Oct 2011), 297–300.
21. Yang, K., Hicks, M., Dong, Q., Austin, T., Sylvester, D. A2: Analog malicious hardware. In *2016 IEEE Symposium on Security and Privacy (SP)* (May 2016). IEEE Computer Society, 18–37.

* **Kaiyuan Yang** (kyang@rice.edu), Dept. of ECE, Rice University, Houston, TX.

* **Matthew Hicks** (mdhicks@gmail.com), Dept. of CS, Virginia Tech, Blacksburg, VA.

Qing Dong, Todd Austin, and Dennis Sylvester ({kaiyuan, mdhicks, qingdong, austin, dmcs}@umich.edu), Department of EECS, University of Michigan, Ann Arbor, MI.

* This work was done at the University of Michigan, Ann Arbor.

DOI:10.1145/3068614

To view the accompanying paper, visit doi.acm.org/10.1145/3068663 rh

Technical Perspective
Humans and Computers Working Together on Hard Tasks

By Ed H. Chi

THE FIELD OF crowdsourcing and human computation has evolved considerably from its early days. At first, crowdsourcing was mainly conceived as a way to obtain ground truth labels for datasets, particularly image datasets, in the mid-2000s. Soon after, researchers began to utilize crowdsourcing for performing large-scale user studies of systems.[a,b] As our understanding of crowdsourcing continued to evolve, researchers realized the workers can be reserved ahead of time to perform real-time tasks.[c] Utilizing this idea, the system described in the following paper demonstrates how a crowd of workers can caption speech nearly as well as a professional captionist. Importantly, this paper was one of the first in a recent set of crowdsourcing papers that demonstrated how human workers can collaborate in concert with computing systems to accomplish a real-time task that is difficult for either one to do by itself. This is notable for many reasons, but let me first summarize the significance of this work.

First, the system demonstrated that significant innovation is needed to get human workers to productively perform the captioning task. For example, the Scribe system slows down the continuous speech for a brief period of time with the right volume changes to emphasize what passage to transcribe for the worker. The volume variations help with audio saliency. This technique is interesting to human-computer interaction (HCI) researchers, since it utilizes our intuition about how we can direct human attention, helping to transform individual untrained workers into better captionists.

Second, the system uses a MapReduce programming paradigm to divide and conquer the various pieces of the captioning tasks and coordinates the workers and their tasks through this organization paradigm. First introduced by Kittur et al.,[d] this is a clever application of the MapReduce paradigm, but instead of applying to computing tasks, the system applies the concept to organizing human tasks.

Third, impressively, to combine the partial contributions from individual workers, the system utilizes a sequence alignment algorithm to combine the streams of input from various workers. This is novel because most crowdsourcing systems use a simple majority voting approach to combine the worker inputs. The use of a sophisticated algorithm here is necessary to fit the captioning problem, and it points to the possibility of other combiner functions in other problems in future research. A natural extension of the alignment algorithm here would be to utilize a task-specific language model trained using deep learning.

From a historical perspective, augmenting humans has been at the very center of much personal computing and HCI research. There has been much talk about the degree in which machine learning (ML) will replace human labor (HL) in the future, but I think that is misguided. Instead, what we see in this research is a good example in which humans and machines work in concert on a very hard task that is currently still too difficult to do by either alone. Interestingly, this aligns well with a historical recounting of the code-breaking work by Turing and colleagues at Bletchley Park in a recent issue of *Communications*: "Another myth is that code-breaking machines eliminated human labor and code-breaking skill ... Technology transcended, rather than supplemented, human labor and bureaucracy."[e] The article points out the real challenge of the whole effort was a combination of the management of a (mostly female!) human operator force along with the Enigma machines. From my perspective, intelligent augmentation of our abilities is the real research frontier.

While we continue to explore the boundary of what is possible for machine intelligence, we should also be exploring the boundary of how humans will interact with machine intelligence. For example, how can we have an intelligent conversation with computing systems? Can I talk to a restaurant recommendation system while I drive home to get ready for a dinner date? How should my television respond if I say I wanted an exciting action film tonight that takes into account the tastes of other family members? If it doesn't have enough information on everyone in the room, will it (he/she?) ask intelligent questions while naturally conversing with my guests? Can I give feedback both via hand gestures as well as voice dialog?

Since an important application of machine intelligence is to augment humans in their desires, goals, and tasks, what we should do is to ask important research questions about human interactions with ML systems. In other words, we should have much better research of ML+HL, ML+HCI, and ML+Human Interaction, and this research is a shining example that points the way. C

a Kittur, A., Chi, E.H., Suh B.. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the ACM Conference on Human-Factors in Computing Systems*, ACM Press (Florence, Italy, 2008), 453–456.

b Egelman, S., Chi, E.H., Dow, S. Crowdsourcing in HCI research. *Ways of Knowing in HCI.* J.S. Olson and W.A. Kellogg, Eds. Springer, NY, 2014, 267–289.

c Bernstein, M., Brandt, J., Miller, R., and Karger, D. Crowds in two seconds: Enabling real-time crowd-powered interfaces. *UIST* 2011.

d Kittur. K, Smus. B., Khamkar. S., and Kraut. R.E. CrowdForge: Crowdsourcing complex work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (2011), 43–52; http://dx.doi.org/10.1145/2047196.2047202

e Haigh, T. Colossal genius: Tutte, flowers, and a bad imitation of Turing. *Commun. ACM 60*, 1 (Jan. 2017), 29–35; https://doi.org/10.1145/3018994

Ed H. Chi is Research Lead Manager and Sr. Staff Research Scientist at Google Inc., Mountain View, CA.

DOI:10.1145/3068663

Scribe: Deep Integration of Human and Machine Intelligence to Caption Speech in Real Time

By Walter S. Lasecki, Christopher D. Miller, Iftekhar Naim, Raja Kushalnagar, Adam Sadilek, Daniel Gildea, and Jeffrey P. Bigham

Abstract

Quickly converting speech to text allows deaf and hard of hearing people to interactively follow along with live speech. Doing so reliably requires a combination of perception, understanding, and speed that neither humans nor machines possess alone. In this article, we discuss how our Scribe system combines human labor and machine intelligence in real time to reliably convert speech to text with less than 4s latency. To achieve this speed while maintaining high accuracy, Scribe integrates automated assistance in two ways. First, its user interface directs workers to different portions of the audio stream, slows down the portion they are asked to type, and adaptively determines segment length based on typing speed. Second, it automatically merges the partial input of multiple workers into a single transcript using a custom version of multiple-sequence alignment. Scribe illustrates the broad potential for deeply interleaving human labor and machine intelligence to provide intelligent interactive services that neither can currently achieve alone.

1. INTRODUCTION AND BACKGROUND

Real-time captioning converts speech to text in under 5s to provide access to live speech content for deaf and hard of hearing (DHH) people in classrooms, meetings, casual conversation, and other events. Current options are severely limited because they either require highly-skilled professional captionists whose services are expensive and not available on demand, or use automatic speech recognition (ASR) which produces unacceptable error rates in many real-world situations.[10] We present an approach that leverages groups of non-expert captionists (people who can hear and type, but are not specially trained stenographers) to collectively caption speech in real-time, and explore this new approach via *Scribe*, our end-to-end system allowing on-demand real-time captioning for live events.[19] Scribe integrates human and machine intelligence in real time to reliably caption speech at natural speaking rates.

The Word Health Organization (WHO) estimates that around 5% of the world population, that is, 360 million people, have disabling hearing loss.[32] They struggle to understand speech and benefit from visual input. Some combine lip-reading with listening, while others primarily watch visual translations of aural information, such as sign language interpreters or real-time typists. While visual access to spoken material can be achieved through sign language interpreters, many DHH people do not know sign language. This is particularly true of the large (and increasing) number of DHH people who lost their hearing later in life, which includes one third of people over 65.[12] Captioning may also be preferred by some to sign language interpreting for technical domains because it does not involve translating from the spoken language to the sign language, but rather transliterating an aural representation to a written one. Finally, like captionists, sign language interpreters are also expensive and difficult to schedule.

People learn to listen and speak at a natural rate of 120–180 words per minute (WPM).[17] They acquire this skill effortlessly without direct instruction while growing up or being immersed in daily linguistic interaction, unlike text generation, which is a trained skill that averages 60–80 WPM for both handwriting[29] and typing.[14] Professional captionists (stenographers) can keep up with most speakers and provide captions that are accurate (95%+) and real-time (within a few seconds). But they are not on-demand (need to be prebooked for at least an hour), and are expensive ($120–$200 per hour).[30] As a result, professionals usually cannot provide access for last minute lectures or other events, or for unpredictable and ephemeral learning opportunities, such as conversations with peers after class.

Automatic speech recognition (ASR) is inexpensive and available on-demand, but its low accuracy in many real settings makes it unusable. For example, ASR accuracy drops below 50% when it is not trained on the speaker, captioning multiple speakers, and/or when not using a high-quality microphone located close to the speaker.[3, 6] Both ASR and the software used to assist real-time captionists often make errors that can change the meaning of the original speech. As DHH people use context to compensate for errors, they often have trouble following the speaker.[6]

Our approach is to combine the efforts of multiple non-expert captionists. Because these non-expert captionists can be drawn from more diverse labor pools than professional captionists, they are more affordable and more easily available on demand. Recent work has shown, for instance, that

Sign languages, such as American Sign Language (ASL) are not simply codes for an aural language, but rather entirely different languages with their own vocabulary, grammar, and syntax.

The original version of this paper is entitled "Real-Time Captioning by Groups of Non-Experts" and was published in UIST, 10/2012, ACM.

workers on Mechanical Turk can be recruited within a few seconds,[1, 2, 11] and engaged in continuous tasks.[21, 24, 25, 28] Recruiting from a broader pool allows workers to be selectively chosen for their expertise not in captioning but in the technical areas covered in a lecture. While professional stenographers are able to type faster and more accurately than most crowd workers, they are not necessarily experts in the field they are captioning, which can lead to mistakes that distort the meaning of transcripts of technical talks.[30] Scribe allows student workers to serve as non-expert captionists for $8–$12 per hour (a typical work-study pay rate). Therefore, we could hire several students for much less than the cost of one professional captionist.

Scribe makes it possible for non-experts to collaboratively caption speech in real time by providing automated assistance in two ways. First, it assists captionists by making the task easier for each individual. It directs each worker to type only part of the stream audio, it slows down the portion they are asked to type so they can more easily keep up, and it adaptively determines the segment length based on each individual's typing speed. Second, it solves the coordination problem for workers by automatically merging the partial input of multiple workers into a single transcript using a custom version of multiple-sequence alignment.

Because captions are dynamic, readers spend far more mental effort reading real-time captions compared to static text. Also, regardless of method, captions require users to absorb information that is otherwise consumed via two senses (vision and hearing) via only one (vision). In classroom settings, this can be particularly common, with content appearing on the board and being referenced in speech. The effort required to track both the captions and the material they pertain to simultaneously is one possible reason why deaf students often lag behind their hearing peers, even with the best accomodations.[26] To address these issues, we also explore how captions can be best presented to users,[16] and show that controlling bookmarks in caption playback can even increase comprehension.[22]

This paper outlines the following contributions:

- Scribe, an end-to-end system that has advantages over current state-of-the-art solutions in terms of availability, cost, and accuracy.
- Evidence that non-experts can collectively cover speech at rates similar to or above that of a professional.
- Methods for quickly merging multiple partial captions to create a single, accurate stream of final results.
- Evidence that Scribe can produce transcripts that both cover more of the input signal and are more accurate than either ASR or any single constituent worker.
- The idea of automatically combining the real-time efforts of dynamic groups of workers to outperform individuals on human performance tasks.

2. CURRENT APPROACHES

We first overview current approaches for real-time captioning, introduce our data set, and define the evaluation metrics used in this paper. Methods for producing real-time captioning services come in three main varieties:

Computer-Aided Real-time Transcription (CART): CART is the most reliable real-time captioning service, but is also the most expensive. Trained stenographers type in shorthand on a "steno" keyboard that maps multiple key presses to phonemes that are expanded to verbatim text. Stenography requires 2–3 years of training to consistently keep up with natural speaking rates that average 141 WPM and can reach 231 WPM.[13]

Non-Verbatim Captioning: In response to the cost of CART, computer-based macro expansion services like C-Print were introduced.[30] C-Print captionists need less training, and generally charge around $60 an hour. However, they normally cannot type as fast as the average speaker's pace, and cannot produce a verbatim transcript. Scribe employs captionists with no training and compensates for slower typing speeds and lower accuracy by combining the efforts of multiple parallel captionists.

Automated Speech Recognition: ASR works well in ideal situations with high-quality audio equipment, but degrades quickly in real-world settings. ASR is has difficulty recognizing domain-specific jargon, and adapts poorly to changes, such as when the speaker has a cold.[6] ASR systems can require substantial computing power and special audio equipment to work well, which lowers availability. In our experiments, we used Dragon Naturally Speaking 11.5 for Windows.

Re-speaking: In settings where trained typists are not common (such as in the U.K.), alternatives have arisen. In re-speaking, a person listens to the speech and enunciates clearly into a high-quality microphone, often in a special environment, so that ASR can produce captions with high accuracy. This approach is generally accurate, but cannot produce punctuation, and has considerable delay. Additionally, re-speaking still requires extensive training, since simultaneous speaking and listening is challenging.

3. LEGION: SCRIBE

Scribe gives users on-demand access to real-time captioning from groups of non-experts via their laptop or mobile devices (Figure 1). When a user starts Scribe, it immediately begins recruiting workers to the task from Mechanical Turk, or a pool of volunteer workers, using LegionTools.[11, 20] When users want to begin captioning audio, they press the start button, which forwards audio to Flash Media Server (FMS) and signals the Scribe server to begin captioning.

Workers are presented with a text input interface designed to encourage real-time answers and increase global coverage (Figure 2). A display shows workers their rewards for contributing in the form of both money and points. In our experiments, we paid workers $0.005 for every word the system thought was correct. As workers type, their input is forwarded to an *input combiner* on the Scribe server. The input combiner is modular to accommodate different implementations without needing to modify Scribe. The combiner and interface are discussed in more detail later in this article.

Figure 1. Scribe allows users to caption audio on their mobile device. The audio is sent to multiple amateur captionists who use Scribe's Web-based interface to caption as much of the audio as they can in real time. These partial captions are sent to our server to be merged into a final output stream, which is then forwarded back to the user's mobile device. Crowd workers are optionally recruited to edit the captions after they have been merged.

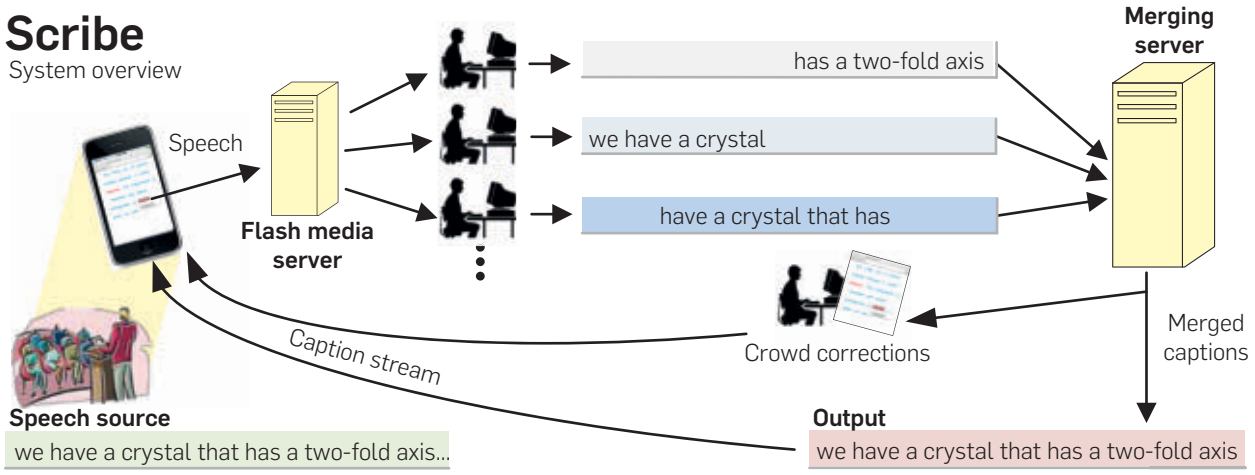


Figure 2. The original worker interface encourages captionists to type quickly by locking in words soon after they are typed. To encourage coverage of specific segments, visual and audio cues are presented, the volume is reduced during off periods, and rewards are increased during these periods.

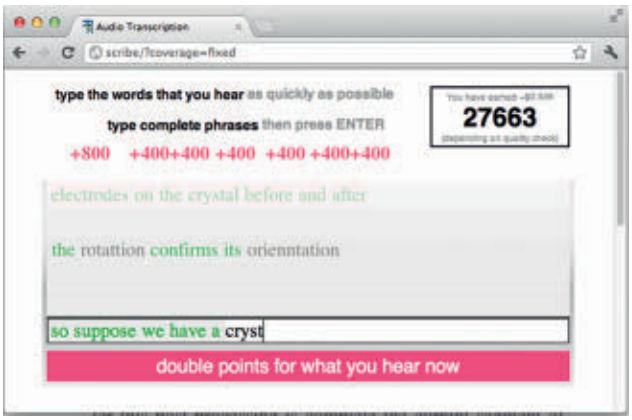
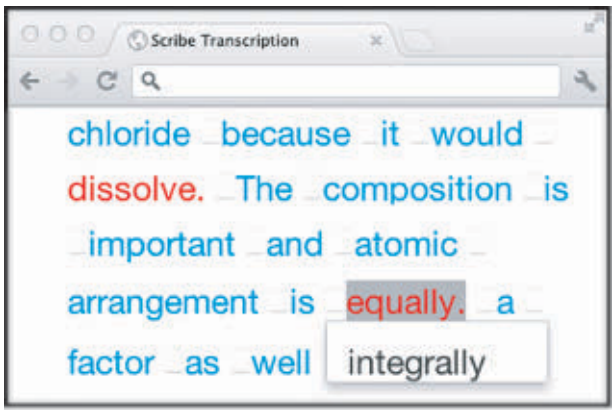


Figure 3. The Web-based interface that shows users the live caption stream returned by Scribe.



The user interface for Scribe presents streaming text within a collaborative editing framework (see Figure 3). Scribe's interface masks the staggered and delayed format of real-time captions with a more natural flow that mimics writing. In doing this, the interface presents the merged inputs from the crowd workers via a dynamically updating Web page, and allows users to focus on reading, instead of tracking changes. We have also developed methods for letting users have more control over their own caption playback, which can improve comprehension.[22] When users are done, pressing stop will end the audio stream, but lets workers complete their current transcription task. Workers are asked to continue working on other audio for a time to keep them active so that response time is reduced if users need to resume captioning.

Though this article focuses on captioning speech from a single person, Scribe can handle dialogues using automated speaker segmentation techniques. We use a standard convolution-based kernel method to first identify distinct segments in a waveform. We then use a one-class support vector machine (SVM) to classify each segment and assign a speaker ID.[15] Prior work has shown such segmentation techniques to be accurate even in the presence of severe noise, such as when talking on a cellphone while driving.[12] The segmentation allows us to *decompose* a dialogue in real-time, then caption each part individually, without burdening workers with the need to determine and annotate which person is currently speaking.

Our solution to the transcription problem is two-fold. First, we designed an interface that facilitates real-time captioning by non-experts and encourages covering the entire audio signal. Second, we developed algorithms for merging partial captions to form one final output stream. The interface and algorithm have been developed to address these problems jointly. For instance, because determining where each word in a partial caption fits into the final transcript is difficult, we designed the interface to encourage workers to type continuous segments during specified periods.

In the following sections, we detail the co-evolution of the worker interface and algorithm for merging partial captions in order to form a final transcript.

4. COORDINATING CAPTIONISTS

Scribe's non-expert captioning interface allows contributors to hear an audio stream of the speaker(s), and provide captions with a simple user interface (UI) (Figure 2). Captionists are instructed to type as much as they can, but are under no pressure to type everything they hear. If they are able, workers are asked to separate contiguous sequences of words by pressing enter. Knowing which word sequences are likely to be contiguous can help later when recombining the partial captions from multiple captionists.

To encourage real-time entry of captions, the interface "locks in" words a short time after they are typed (500ms). New words are identified when the captionist types a space after the word, and are sent to the server. The delay is added to allow workers to correct their input while adding as little additional latency as possible to it. When the captionist presses enter (or following a 2s timeout during which they have not typed anything), the line is confirmed and animates upward. During the 10–15s trip to the top of the display (depending on settings), words that Scribe determines were entered correctly (based on either spell-checking or overlap with another worker) are colored green. When the line reaches the top, a point score is calculated for each word based on its length and whether it has been determined to be correct.

To recover the true speech, non-expert captions must *cover* all of the words spoken. A primary reason why the partial transcriptions may not fully cover the true signal relates to *saliency*, which is defined in a linguistic context as "that quality which determines how semantic material is distributed within a sentence or discourse, in terms of the relative emphasis which is placed on its various parts".[7] Numerous factors influence what is salient, and so it is likely to be difficult to detect automatically. Instead, we inject artificial saliency adjustments by systematically varying the volume of the audio signal that captionists hear. Scribe's captionist interface is able to vary the volume over a given a period with an assigned offset. It also displays visual reminders of the period to further reinforce this notion.

Initially, we tried dividing the audio signal into segments that we gave to individual workers. We found several problems with this approach. First, workers tended to take longer to provide their transcriptions as it took them some time to get into the flow of the audio. A continuous stream avoids this problem. Second, the interface seemed to encourage workers to favor quality over speed, whereas streaming content reminds workers of the real-time nature of the task. The continuous interface was designed in an iterative process involving tests with 57 remote and local users with a range of backgrounds and typing abilities. These tests showed that workers tended to provide chains of words rather than disjoint words, and needed to be informed of the motivations behind aspects of the interface to use them properly.

A non-obvious question is what the period of the volume changes should be. In our experiments, we chose to play the audio at regular volume for 4s and then at a lower volume for 6s. This seems to work well in practice, but it is likely that it is not ideal for everyone (discussed below). Our experience suggests that keeping the *in* period short is preferable even when a particular worker was able to type more than the period because the latency of a worker's input tended to go up as they typed more consecutive words.

5. IMPROVING HUMAN PERFORMANCE

Even when workers are directed to small, specific portions of the audio, the resulting partial captions are not perfect. This is due to several factors, including bursts of increased speaking rates being common, and workers mis-hearing some content due to a particular accent or audio disruption. To make the task easier for workers, we created *TimeWarp,*[23] which allows each worker to type what they hear in clips with a lower playback rate, while still keeping up with real time *and* maintaining context from content they are not responsible for.

5.1. Warping time

TimeWarp manages this by balancing the play speed during *in* periods, where workers are expected to caption the audio and the playback speed is reduced, and *out* periods, where workers listen to the audio and the playback speed is increased. A *cycle* is one *in* period followed by an *out* period. At the beginning of each cycle, the worker's position in the audio is aligned with the real-time stream. To do this, we first need to select the number of different sets of workers N that will be used in order to partition the stream. We call the length of the *in* period P_i, the length of the *out* period P_o and the play speed reduction factor r. Therefore, the playback rate during *in* periods is $\frac{1}{r}$. The amount of the real-time stream that gets buffered while playing at the reduced speed is compensated for by an increased playback speed of $\frac{N-1}{N-r}$ during *out* periods. The result is that the cycle time of the modified stream equals the cycle time of the unmodified stream.

To set the length of P_i for our experiments, we conducted preliminary studies with 17 workers drawn from Mechanical Turk. We found that their mean typing speed was 42.8 WPM on a similar real-time captioning task. We also found that a worker could type *at most* 8 words in a row on average before the per-word latency exceeded 8s (our upper bound on acceptable latency). Since the mean speaking rate is around 150 WPM,[13] workers will hear 8 words in roughly 3.2s, with an entry time of roughly 8s from the last word spoken. We used this to set $P_i = 3.25s, P_o = 9.75s$, and $N = 4$. We chose $r = 2$ in our tests so that the playback speed would be $\frac{1}{2} = 0.5$ times for *in* periods, and the play speed for *out* periods is $\frac{N-1}{N-r} = \frac{3}{2} = 1.5$ times.

To speed up and slow down the play speed of content being provided to workers without changing the pitch (which would make the content more difficult to understand for the worker), we use the Waveform Similarity Based Overlap and Add (WSOLA) algorithm.[4] WSOLA works by dividing the signal into small segments, then either skipping (to increase play speed) or adding (to decrease play speed) content, and finally stitching these segments back together. To reduce the number of sound artifacts, WSOLA finds overlap points with similar wave forms, then gradually transitions between sequences during these overlap periods.

5.2. Integrating ASR into crowd captioning

Combining ASR into human captioning workflows can also help improve captioning performance. By using the suggestions from an ASR system to provide an initial "baseline" answer that crowd workers can correct, we can reduce latency. However, above an error rate of $\geq \sim 30\%$ error, the ASR input actually increases latency because of the cost of finding and repairing mistakes.[9] The opposite integration is also possible: by using sparse human input to provide corrections to the word lattice of an ASR system, it is possible to reduce the error rate.[8]

6. AGGREGATING PARTIAL CAPTIONS

The problem of aligning and aggregating multiple partial transcripts can be mapped to the well-studied Multiple Sequence Alignment (MSA) problem. The basic formulation of the problem involves some number of ordered sequences that include at least some similar elements (coming from the same "dictionary" of possible terms plus a "gap" term). Finding the alignment that minimizes total distance between all pairs of sequences is a non-trivial problem because, in the worst case, all possible alignments of the content of each sequence—including all possible spaces containing a gap term—may need to be explored. This optimization problem has been shown to be NP-complete,[31] and exact algorithms have time complexity that is exponential in the number of sequences. As a result, it is often necessary to apply heuristic approximations to perform MSA with in a reasonable amount of time.

In practice, MSA is a well-studied problem in the bioinformatics literature that has long been used in aligning genome sequences, but also has applications in approximate text matching for information retrieval, and in many other domains. Tools like MUSCLE Edgar[5] provide extremely powerful solvers for MSA problems. Accordingly, our approach is to formulate our text-matching problem as MSA.

6.1. Progressive alignment algorithms

Most MSA algorithms for biological sequences follow a progressive alignment strategy that first performs pairwise alignment among the sequences, and then merges sequences progressively according to a decreasing order of pairwise similarity. Due to the sequential merging strategy, progressive alignment algorithms cannot recover from the errors made in the earlier iterations, and typically do not work well for the caption alignment task.

6.2. Graph-based alignment

We first explored a graph-based incremental algorithm to combine partial captions on the fly.[19] The aggregation algorithm incrementally builds a chain graph, where each node represents a set of equivalent words entered by the workers, and the links between nodes are adjusted according to the order of the input words. A greedy search is performed to identify the path with the highest confidence, based on worker input and an n-gram language model. The algorithm is designed to be used online, and hence has high speed and low latency. However, due to the incremental nature of the algorithm and the lack of a principled objective function, it is not guaranteed to find the globally optimal alignment for the captions.

6.3. Weighted A* search algorithm

We next developed a weighted A* search based MSA algorithm to efficiently align the partial captions.[27] To do this, we formulate MSA as graph-traversal over a specialized lattice. Our search algorithm then takes each node as a state, allowing us to estimates the cost function $g(n)$ and the heuristic function $h(n)$ for each state.

At each step of the A* search algorithm, the node with the smallest evaluation function is extracted from the priority queue Q and expanded by one edge. This is repeated until a full alignment is produced (the goal state). While weighted A* significantly speeds the search for the best alignment, it is still too slow for very long sequences. To counteract this, we use fixed-size time windows to scope the exploration to the most-likely paths.

7. EXPERIMENTAL RESULTS

We have tested our system with non-expert captionists drawn from both local and remote crowds. As a data set, we used lectures freely available from MIT OpenCourseWare. These lectures were chosen because one of the main goals of Scribe is to provide captions for classroom activities, and because the recording of the lectures roughly matches our target as well—there is a microphone in the room that often captures multiple speakers, for example, students asking questions. We chose four 5 min segments that contained speech from courses in electrical engineering and chemistry, and had them professionally transcribed at a cost of $1.75 per minute. Despite the high cost, we found a number of errors and omissions. We corrected these to obtain a completely accurate baseline.

7.1. Core system study results

Our study used 20 local participants. Each participant captioned 23 min of aural speech over a period of approximately 30 min. Participants first took a standard typing test and averaged a typing rate of 77.0 WPM (*SD*=15.8) with 2.05% average error (*SD*=2.31%). We then introduced participants to the real-time captioning interface, and had them caption a 3 min clip using it. Participants were then asked to caption the four 5 min clips, two of which were selected to contain saliency adjustments. We measure coverage (recall within a 10s per-word time bound), precision, and WER.

We found that saliency adjustment made a significant difference on coverage ranges. For the electrical engineering clip, the difference was 54.7% (*SD*=9.4%) for words in the selected periods as compared to only 23.3% (*SD*=6.8%) for words outside of those periods. For the chemistry clips, the difference was 50.4% (*SD*=9.2%) of words appearing inside the highlighted period as compared to 15.4% (*SD*=4.3%) of words outside of the period.

To see if workers on Mechanical Turk could complete this task effectively—which would open up a large new set of workers who are available on-demand—we recruited a crowd to caption the four clips (20 min of speech). Our tasks paid $0.05 and workers could make an additional $0.002 bonus per word. We provided workers with a 40s instructional video to beign

http://ocw.mit.edu/courses/.

with. In total, 18 workers participated, collectively achieving 78.0% coverage. The average coverage over just three workers was 59.7% (*SD*=10.9%), suggesting we could be conservative in recruiting workers and cover much of the input signal.

In our tests, workers achieved an average of 29.0% coverage, ASR achieved 32.3% coverage, CART achieved 88.5% coverage and Scribe reached 74% out of a possible 93.2% coverage using 10 workers (Figure 4). Collectively, workers had an average latency of 2.89 significantly improving on CART's latency of 4.38s. For this example, we tuned our combiner to balance coverage and precision (Figure 5), getting an average of 66% and 80.3% respectively. As expected, CART outperforms the other approaches. However, our combiner presents a clear improvement over both ASR and a single worker.

7.2. Improved combiner results

We further improved alignment accuracy by applying a novel weighted-A* MSA algorithm.[27] To test this, we used the same four 5 min long audio clips as before. We tested three configurations of our algorithm: (1) no agreement needed with a 15s sliding window, (2) two-person agreement needed with a 10s window, and (3) two-person agreement needed with a 15s window. We compare the results from these three configurations to our original graph-based method, and to the MUSCLE package (Figure 6).

The with agreement and a 15s window (the best performing setting), our algorithm achieves 57.4% average (1-WER) accuracy, providing 29.6% improvement with respect to the graph-based system (average accuracy 42.6%), and 35.4% improvement with respect to the MUSCLE-based MSA system (average accuracy 41.9%). On the same set of audio clips, we obtained 36.6% accuracy using ASR (Dragon Naturally Speaking, version 11.5 for Windows), which is worse than all the crowd-powered approaches. We intentionally did not optimize the ASR for the speaker or acoustics, since DHH students would also not be able to do this in realistic settings.

7.3. TimeWarp results

To evaluate *TimeWarp*, we ran two studies that asked participants to caption a 2.5 min (12 captioning cycles) lecture clip. Again, we ran our experiments with both local participants and workers recruited from Mechanical Turk. Tests were divided into two conditions: time warping on or off, and were randomized across four possible time offsets: 0s, 3.25s, 6.5s, 9.75s.

Local participants were again generally proficient (but non-expert) typists and had time to acquaint themselves with the system, which may better approximate student employees captioning a classroom lecture. We recruited 24 volunteers (mostly students) and had them practice with our baseline interface before using the time warp interface. Each worker was asked to complete two trials, one with *TimeWarp* and one without, in a random order.

We also recruited 139 Mechanical Turk workers, who were allowed to complete at most two tasks and were randomly routed to each condition (providing 257 total responses). Since Mechanical Turk often contains low quality (or even malicious workers),[18] we first removed inputs which got less than 10% coverage or precision or were outliers more than 2σ from the mean. A total of 206 tasks were approved by this quick check. Task payment amounts were the same as for our studies described above.

Our student captionists were able to caption a majority of the content well even without *TimeWarp*. The mean coverage from all 48 trials was 70.23% and the mean precision was 70.71%, compared to the 50.83% coverage and 62.23% precision for workers drawn from Mechanical Turk. For student captionists, total coverage went up 2.02%, from 69.54% to 70.95%, and precision went up by 2.56% from 69.84% to 71.63%, but neither of these differences were detectably significant. However, there was a significant improvement in mean latency per word, which improved 22.46% from 4.34s to 3.36s ($t(df) = 2.78$, $p <$

Figure 4. Optimal coverage reaches nearly 80% when combining the input of four workers, and nearly 95% with all 10 workers, showing captioning audio in real time with non-experts is feasible.

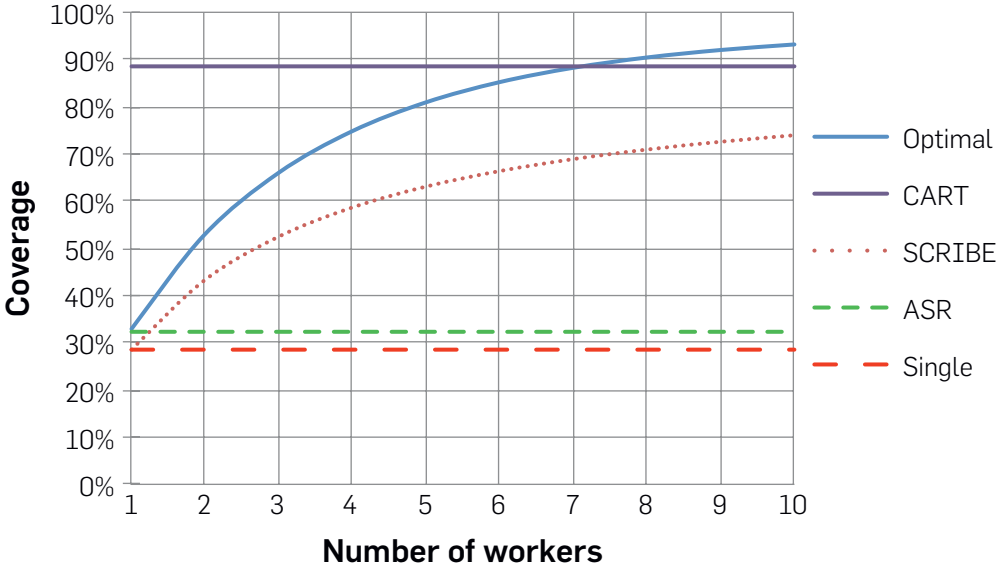


Figure 5. Precision-coverage curves for the electrical engineering (EE) and chemistry (Chem) lectures using different combiner parameters with 10 workers. In general, increasing coverage reduces accuracy.

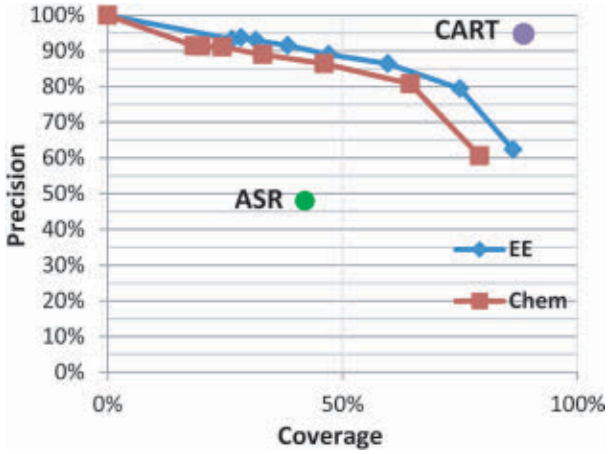


Figure 6. Evaluation of different systems on using (1-WER) as an accuracy measure (higher is better).

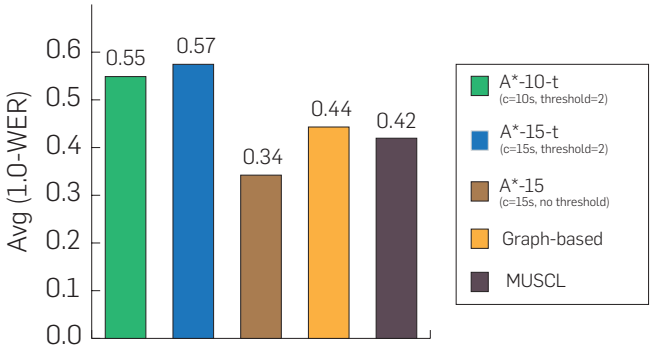
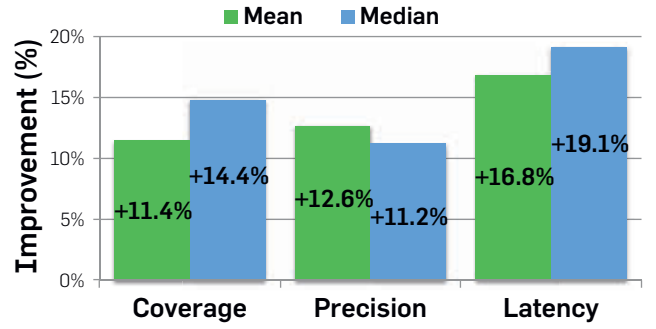


Figure 7. Relative improvement from no warp to warp conditions in terms of mean and median values of coverage, precision, and latency. We expected coverage and precision to improve. Shorter latency was unexpected, but resulted from workers being able to consistently type along with the audio instead of having to remember and go back as the speech outpaced their typing.



.01). Mechanical Turk workers' mean coverage (Figure 7) increased 11.39% ($t(df) = 2.19, p < .05$), precision increased 12.61% ($t(df) = 3.90, p < .001$), and latency was reduced by 16.77% ($t(df) = 5.41, p < .001$).

8. CONCLUSION AND FUTURE WORK

Scribe is the first system capable of making reliable, affordable captions available on-demand to deaf and hard of hearing users. Scribe has allowed us to explore further issues related to how real-time captions can be made more useful to end users. For example, when captions are used, we have shown that students' comprehension of instructional material significantly improves when they have the ability to control when the captions play, and track their position so that they are not overwhelmed by using one sensory channel to absorb content that is designed to be split between both vision and hearing. To help address this problem, we built a tool that lets students highlight or pause at the last position they read before looking away from the captions to view other visual content.[22]

While we have discussed how automation can be used to effectively mediate human caption generation, advances in ASR technologies can aid Scribe as well. By including ASR systems as workers, we can take advantage of the affordable, highly-scalable nature of ASR in settings where it works, while using human workers to ensure that DHH users always have access to accurate captions. ASR can eventually use Scribe as an in situ training tool, resulting in systems that are able to provide reliable captions right out of the box using human intelligence, and scale to fully automated solutions quicker than would otherwise be possible.

More generally, Scribe is an example of an interactive system that deeply integrates human and machine intelligence in order to provide a service that is still beyond what computers can do alone. We believe it may serve as a model for interactive systems that solve other problems of this type.

Acknowledgments

This work was supported by the National Science Foundation under awards #IIS-1149709 and #IIS-1218209, the University of Michigan, Google, an Alfred P. Sloan Foundation Fellowship, and a Microsoft Research Ph.D. Fellowship.

References

1. Bernstein, M.S., Brandt, J.R., Miller, R.C., Karger, D.R. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11 (New York, NY, USA, 2011). ACM, 33–42.
2. Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tatarowicz, A., White, B., White, S., Yeh, T. Vizwiz: Nearly real-time answers to visual questions. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, (New York, NY, USA, 2010). ACM, 333–342.
3. Cooke, M., Green, P., Josifovski, L., Vizinho, A. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech commun. 34*, 3 (2001), 267–285.
4. Driedger, J. Time-scale modification algorithms for music audio signals. Master's thesis, Saarland University, 2011.
5. Edgar, R. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research 32*, 5 (2004), 1792–1797.
6. Elliot, L.B., Stinson, M.S., Easton, D., Bourgeois, J. College students learning with C-print's education software and automatic speech recognition. In *American Educational Research Association Annual Meeting* (New York, NY, 2008), AERA.
7. Flowerdew, J.L. Salience in the performance of one speech act:the case of definitions. *Discourse Processes 15*, 2 (Apr–June 1992), 165–181.
8. Metze, F., Gaur, Y., Bigham, J. P. Manipulating word lattices to incorporate human corrections. In *Proceedings of INTERSPEECH*, (2016).
9. Gaur, Y., Lasecki, W.S., Metze, F., Bigham, J.P. The effects of automatic speech recognition quality on human transcription latency. In *Proceedings of the 13th Web for All Conference* (2016) ACM.
10. Glass, J.R., Hazen, T.J., Cyphers, D.S., Malioutov, I., Huynh, D., Barzilay, R. Recent progress in the MIT spoken lecture processing project. In *Interspeech* (2007), 2553–2556.
11. Gordon, M., Bigham, J.P., Lasecki, W.S. Legiontools: A toolkit+ UI for recruiting and routing crowds to synchronous real-time tasks. In *Adjunct Proceedings of the 28th*

Annual ACM Symposium on User Interface Software & Technology (2015) ACM, 81–82.
12. Gordon-Salant, S. Aging, hearing loss, and speech recognition: stop shouting, i can't understand you. In *Perspectives on Auditory Research*, volume 50 of *Springer Handbook of Auditory Research*. A.N. Popper and R.R. Fay, eds. Springer New York, 2014, 211–228.
13. Jensema, C., McCann, R., Ramsey, S. Closed-captioned television presentation speed and vocabulary. In *Am Ann Deaf 140*, 4 (October 1996), 284–292.
14. John, B.E. Newell, A. Cumulating the science of HCI: from s-R compatibility to transcription typing. *ACM SIGCHI Bulletin 20*, SI (Mar. 1989), 109–114.
15. Kadri, H., Davy, M., Rabaoui, A., Lachiri, Z., Ellouze, N., et al. Robust audio speaker segmentation using one class SVMs. In *IEEE European Signal Processing Conference* (Lausanne, Switzerland, 2008) ISSN: 2219-5491.
16. Kushalnagar, R.S., Lasecki, W.S., Bigham, J.P. Captions versus transcripts for online video content. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, W4A '13, (New York, NY, 2013), ACM, 32:1–32:4.
17. Kushalnagar, R.S., Lasecki, W.S., Bigham, J.P. Accessibility evaluation of classroom captions. *ACM Trans Access Comput. 5*, 3 (Jan. 2014), 1–24.
18. Lasecki, W. Bigham, J. Online quality control for real-time crowd captioning. In *International ACM SIGACCESS Conference on Computers & Accessibility*, ASSETS 2012, 2012.
19. Lasecki, W., Miller, C., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R., Bigham, J. Real-time captioning by groups of non-experts. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, (2012), 23–34.
20. Lasecki, W.S., Gordon, M., Koutra, D., Jung, M.F., Dow, S.P., Bigham, J.P. Glance: rapidly coding behavioral video with the crowd. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, (New York, NY, 2014). ACM, 1.
21. Lasecki, W.S., Homan, C., Bigham, J.P. Architecting real-time crowd-powered systems. *Human Computation 1*, 1 (2014).
22. Lasecki, W.S., Kushalnagar, R., Bigham, J.P. Helping students keep up with real-time captions by pausing and highlighting. In *Proceedings of the 11th Web for All Conference*, W4A '14 (New York, NY, 2014). ACM, 39:1–39:8.
23. Lasecki, W.S., Miller, C.D., Bigham, J.P. Warping time for more effective real-time crowdsourcing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13 (New York, NY, 2013). ACM, 2033–2036.
24. Lasecki, W.S., Murray, K., White, S., Miller, R.C., Bigham, J.P. Real-time crowd control of existing interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, (New York, NY, 2011). ACM, 23–32.
25. Lasecki, W.S., Song, Y.C., Kautz, H., Bigham, J.P. Real-time crowd labeling for deployable activity recognition. In *Proceedings of the 2013 conference on Computer supported cooperative work* (2013) ACM, 1203–1212.
26. Marschark, M., Sapere, P., Convertino, C., Seewagen, R. Access to postsecondary education through sign language interpreting. *J Deaf Stud Deaf Educ. 10*, 1 (Jan. 2005), 38–50.
27. Naim, I., Gildea, D., Lasecki, W.S., Bigham, J.P. Text alignment for real-time crowd captioning. In *Proceedings North American Chapter of the Association for Computational Linguistics (NAACL)* (2013), 201–210.
28. Salisbury, E., Stein, S., Ramchurn, S. Real-time opinion aggregation methods for crowd robotics. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems (2015), 841–849.
29. Turner, O.G. The comparative legibility and speed of manuscript and cursive handwriting. *The Elementary School Journal* (1930), 780–786.
30. Wald, M. Creating accessible educational multimedia through editing automatic speech recognition captioning in real time. *Interactive Technology and Smart Education 3*, 2 (2006), 131–141.
31. Wang, L., Jiang, T. On the complexity of multiple sequence alignment. *J Comput Biol. 1*, 4 (1994), 337–348.
32. World Health Organization. Deafness and hearing loss, fact sheet N300. http://www.who.int/mediacentre/factsheets/fs300/en/, February 2014.

Walter S. Lasecki (wlasecki@umich.edu), Computer Science & Engineering, University of Michigan.

Christopher D. Miller, Iftekhar Naim, Adam Sadilek, and Daniel Gildea (c.miller@rochester.edu) ([inaim,sadilek,gildea]@cs.rochester.edu), Computer Science Department, University of Rochester.

Raja Kushalnagar (raja.kushalnagar@gallaudet.edu), Information Technology Program, Gallaudet University.

Jeffrey P. Bigham (jbigham@cmu.edu), HCI and LT Institutes, Carnegie Mellon University.

Brigham Young University
Faculty Position

The Department of Electrical and Computer Engineering at Brigham Young University announces an opening for a professorial continuing-faculty-status (tenure) track position. While our preference is in the area of Computer Engineering, applicants in all areas of Electrical and Computer Engineering will be considered.

Areas of interest include but are not limited to: Computer Systems (including architecture, IoT and embedded/real-time systems, networking, security, software, compilers, O/S, parallel systems, etc.), Robotics and Autonomous Systems, Computer Vision, Machine Learning, Data Science, Distributed Systems, and Digital Systems Design (FPGA and/or VLSI).

The department has state-of-the-art facilities in computing and supercomputing, autonomous vehicles and computer vision, control systems, optics, and microelectronic fabrication. Excellent research programs exist in the department in the areas of FPGA-based computing, high-performance embedded systems, autonomous vehicles and control, robotics and computer vision, high-speed low-power electronics, digital communications systems, signal processing, biomedical imaging, optics, and microfluidics. Successful candidates will be expected to strengthen undergraduate and graduate education and to develop an outstanding research program to complement existing research or develop new research areas.

The ACT score for the average BYU entering freshman is above the 90th percentile nationally. BYU is also fifth on the NSF's list of U.S. baccalaureate-origin institutions for engineering doctorate recipients. We expect our faculty to challenge these outstanding students to reach their potential.

Successful candidates will be hired at the assistant, associate, or full professor level depending on experience. Requirements include a doctorate in computer engineering, computer science, electrical engineering, or closely related field and a willingness to fully support and participate in the ideals and mission of BYU.

An on-line application for this position can be found at: https://yjobs.byu.edu, job posting #64783.

Questions regarding the position can be directed to:

Dr. Aaron Hawkins, Faculty Committee Chair
Dept of ECE, Brigham Young University
459 CB
Provo UT 84602
ahawkins@byu.edu

*The position will remain open until filled.

** Brigham Young University is an equal opportunity employer. All faculty are required to abide by the university's Honor Code and Dress & Grooming Standards. Strong preference will be given to qualified candidates who are members in good standing of the affiliated Church, The Church of Jesus Christ of Latter-day Saints. Successful candidates are expected to support and contribute to the academic and religious missions of the university within the context of the principles and doctrine of the affiliated Church.

Equal Opportunity Employer: m/f/Vets/Disability

Macalester College
Two Tenure-Track Assistant Professors of Computer Science

Macalester invites applications for two tenure-track positions at the assistant professor level to begin Fall 2018. Candidates must have or be completing a PhD in Computer Science and have a strong commitment to both teaching and research in an undergraduate liberal arts environment. We are especially interested in candidates who are enthusiastic to teach a broad range of undergraduate courses. This person will contribute to the teaching of our introductory, core and advanced courses, and mentor undergraduate research.

Macalester offers majors in Computer Science, Mathematics, and Applied Mathematics and Statistics, and minors in Computer Science, Mathematics, and Statistics, as well as a new minor in Data Science. Typical class sizes range from 15 to 32 students. We encourage innovative pedagogy and curriculum and emphasize computer science's interdisciplinary connections. We have close relationships with several disciplines both within and beyond the sciences, and we are interested in candidates whose work spans disciplinary boundaries. Areas of highest priority include computer and data security and privacy, mobile and ubiquitous computing, computer networks and systems. For more information about our programs, see: http://macalester.edu/mscs

About Macalester

Macalester College is a highly selective, private liberal arts college in the vibrant Minneapolis-Saint Paul metropolitan area. The Twin Cities have a population of approximately three million, a rich arts community, strong local industries, an award-winning parks system, and are home to many colleges and universities, including the University of Minnesota. Macalester's diverse student body comprises over 2000 undergraduates from 40 states and the District of Columbia and over 90 nations. The College maintains a longstanding commitment to academic excellence with a special emphasis on internationalism, multiculturalism, and service to society. We are especially interested in applicants dedicated to excellence in teaching and research/creative activity within a liberal arts college community. As an Equal Opportunity employer supportive of affirmative efforts to achieve diversity among its faculty, Macalester College strongly encourages applications from women and members of underrepresented minority groups.

Applying

To apply via Academic Jobs Online submit (1) curriculum vitae, (2) graduate transcripts, (3) three letters of recommendation (at least one of which discusses your potential as a teacher), (4) a cover letter that addresses why you are interested in Macalester, (5) a statement of teaching philosophy, and (6) a research statement. Please contact Shilad Sen at ssen@macalester.edu with any questions about the position. Evaluation of applications will begin October 15, 2017 and continue until the position is filled.

Apply now: https://www.macalester.edu/academics/mscs/compscitenure-trackjob.html

National University of Singapore
Senior and Junior Tenure-Track Faculty Positions in Artificial Intelligence

The Department of Computer Science at the National University of Singapore (NUS) invites applications for one Distinguished Professorship and several tenure-track faculty positions in artificial intelligence, machine learning, computational neuroscience and related areas of robotics. The Department enjoys ample research funding, moderate teaching loads, excellent facilities, and extensive international collaborations. We have a full range of faculty covering all major research areas in computer science and a thriving PhD program that attracts the brightest students from the region and beyond. More information is available at www.comp.nus.edu.sg/careers.

NUS offers highly competitive salaries and is situated in Singapore, an English-speaking cosmopolitan city that is a melting pot of many cultures, both the east and the west. Singapore offers a safe and family-friend environment with high quality education and healthcare at all levels, as well as very low tax rates. Singapore has also recently launched a S$150 million national initiative, AI.SG, to expand research, development, and adoption of AI technologies. AI.SG will be hosted at NUS.

Candidates for the Distinguished Professor position should have an established record of outstanding research achievements, thought leadership, and international stature in artificial intelligence.

Candidates for Assistant Professor positions should demonstrate excellent research potential in AI, and a strong commitment to teaching. Truly outstanding Assistant Professor applicants will be considered for the endowed Sung Kah Kay Assistant Professorship.

Application Details:

Submit the following documents (in a single PDF) online via: https://faces.comp.nus.edu.sg

1. A cover letter that indicates the position applied for and the main research interests
2. Curriculum Vitae
3. A teaching statement
4. A research statement

► Provide the contact information of 3 referees when submitting your online application, or, arrange for at least 3 references to be sent directly to csrec@comp.nus.edu.sg.
► Application reviews will commence immediately and continue until positions are filled.
► Please submit your application by 1 December 2017.

If you have further enquiries, please contact the Search Committee Chair, Weng-Fai Wong, at csrec@comp.nus.edu.sg

University of Central Florida

Assistant or Associate Professor in Faculty Cluster for Cyber Security and Privacy

The University of Central Florida (UCF) is recruiting a tenure-track assistant or associate professor for its cyber security and privacy cluster. This position has a start date of August 8, 2018.

This will be an interdisciplinary position that will be expected to strengthen both the cluster and a chosen tenure home department, as well as a possible combination of joint appointments. The candidate can choose a combination of units from the cluster for their appointment (see http://www.ucf.edu/faculty/cluster/cyber-security-and-privacy/).

The ideal junior candidates will have a strong background in cyber security and privacy, and be on an upward leadership trajectory in these areas. They will have research impact, as reflected in high-quality publications and the ability to build a well-funded research program. All relevant technical areas will be considered. We are looking for a team player who can help bring together current campus efforts in cyber security or privacy. In particular, we are looking for someone who will work at the intersection of several areas, such as: (a) hardware and IoT security, (b) explaining and predicting human behavior, creating policies, studying ethics, and ensuring privacy, (c) cryptography and theory of security or privacy, or (d) tools, methods, training, and evaluation of human behavior.

Minimum qualifications include a Ph.D., terminal degree, or foreign degree equivalent from an accredited institution in an area appropriate to the cluster, and a record of high impact research related to cyber security and privacy, demonstrated by a strong scholarly and/or funding record. A history of working with teams, especially teams that span multiple disciplines, is a strongly preferred qualification. The position will carry a rank commensurate with the candidate's prior experience and record.

Candidates must apply online at https://www.jobswithucf.com/postings/50404 and attach the following materials: a cover letter, curriculum vitae, teaching statement, research statement, and contact information for three professional references. In the cover letter candidates must address their background in cyber security and privacy, and identify the department or departments for their potential tenure home and the joint appointments they would desire. When applying, have all documents ready so they can be attached at that time, as the system does not allow resubmittal to update applications.

As an equal opportunity/affirmative action employer, UCF encourages all qualified applicants to apply, including women, veterans, individuals with disabilities, and members of traditionally underrepresented populations.

For questions, please contact the Cluster's Search Committee Chair, Gary T. Leavens, at Leavens@ucf.edu.

University of Central Florida

Cluster Lead, Cyber Security and Privacy Cluster

The University of Central Florida (UCF) is recruiting a lead for its cluster on cyber security and privacy. This position has a start date of August 8, 2018. The position will carry a rank of associate or full professor, commensurate with the candidate's prior experience and record. The lead is expected to have credentials and qualifications like those expected of a tenured associate or full professor. To obtain tenure, the selected candidate must have a demonstrated record of teaching, research and service commensurate with rank.

This will be an interdisciplinary position that will be expected to strengthen both the cluster and a chosen tenure home department, as well as a possible combination of joint appointments. The candidate can choose a combination of units from the cluster for their appointment. (See http://www.ucf.edu/faculty/cluster/cyber-security-and-privacy/.) Both individual and interdisciplinary infrastructure and startup support will be provided.

The ideal candidate will have a strong background in cyber security and privacy and outstanding research credentials and research impact, as reflected in a sustained record of high quality publications and external funding. All relevant technical areas will be considered including: network security, cryptography, blockchains, hardware security, trusted computing bases, cloud computing, human factors, anomaly detection, forensics, privacy, and software security, as well as applications of security and privacy to areas such as IoT, cyber-physical systems, finance, and insider threats. A history of working with teams, especially teams that span multiple disciplines, is a strongly preferred qualification. A record of demonstrated leadership is highly desired, as we are looking for a leader to bring together all the current campus efforts in cyber security and privacy. This includes three cluster members already hired, as well as a pending hire for the 2017-18 academic year.

Minimum qualifications include a Ph.D. from an accredited institution in an appropriate area, and a record of high impact research related to cyber security and privacy demonstrated by a strong scholarly publication record and a significant amount of sustained funding.

Candidates must apply online at http://www.jobswithucf.com/postings/50044 and upload the following materials: cover letter, CV, teaching and research statements, and contact information for 3 professional references. In the cover letter, candidates should address their background, and identify the department for their potential tenure home and any desired joint appointments.

An equal opportunity/affirmative action employer, UCF encourages all qualified applicants to apply, including women, veterans, individuals with disabilities, and members of traditionally underrepresented populations.

Questions can be directed to the search committee chair, Gary T. Leavens, at Leavens@ucf.edu.

[CONTINUED FROM P. 104] ing.

Another example is the Shannon trick of synthesizing text. Imagine if you start typing an SMS on your phone but you keep using the predictive function. The algorithm is very basic—it's just "look for the last time something like this occurred and steal the next most probable letter." But you get really interesting results, because you have a lot of data.

Thanks to the Internet, you've got access to a massive corpus of data. Didn't one of your early papers examine two million images from Flickr?

Exactly. Initially, we said, "We'll just download 20,000 images." The results weren't great. But my then-grad student, James Hays, was like, "Why don't we just keep downloading?" If you look at the big neural networks right now, it is really impressive what they can do. But I think people are forgetting that one of the reasons they're so powerful is that they are able to gobble up orders of magnitude more data than we could do with earlier methods. This is not very glamorous, because it suggests that humans are not so smart. It's really the data.

That reminds me of the old philosophical debate about experiential vs. a priori knowledge.

People like to rationalize. They like to get a nice beautiful theory of the world. But reality is often really noisy and complicated, and in a way, data allows you to use this complexity, to not have to throw it away. It's not the minimalist beauty, the clean lines. It's the beauty of a jumbled mess.

Your analyses of photographic data sets like faces and building facades have also revealed lots of visual trends that might not otherwise have been easy to notice.

That is a big beautiful promise and we're only scratching the surface. People are good at finding certain kinds of patterns. We can hold a small number of things in our minds and compare them. We are not able to find a tiny, tiny little pattern over thousands or millions of data points, or very subtle changes over a long range of time. Using computer vision and techniques, I'm hoping we can make new discoveries in ways people have not been able to do before. I would love to discover something that people haven't noticed yet.

"We don't see things as they are; we see them tinted by language and culture and all the baggage."

What about your recent discovery, in an analysis of 150,000 American yearbook photos, that people's smiles broadened during each decade since 1900?

For the portraits, we were very happy to see the increase in smiling over time. We thought, wow, this is a really cool discovery. Of course, then we found some psychological literature that indicates people have already noticed this.

Your work has found applications in areas from entertainment to security. What other pie-in-the-sky applications or discoveries do you hope to see?

Frankly, my goal has always been to understand and model biological vision. Human vision is too hard, because it connects with everything else. We don't see things as they are; we see them tinted by language and culture and all the baggage. But if I'm able to build a model of a rabbit's vision or a rat's vision by the time I retire, I think that would be absolutely fantastic. Imagine having a model of this remarkable apparatus that almost all living creatures possess.

Now, because this is such a hard problem, you don't get wins very often. A lot of the time, it's a depressing slog. But once in a while, as a kind of by-product, some really neat things come up that you can use to create pretty pictures. And I think the world needs more pretty pictures.

Leah Hoffmann is a technology writer based in Piermont, NY.

DOI:10.1145/3121444

Leah Hoffmann

Q&A
All The Pretty Pictures

Alexei Efros, recipient of the 2016 ACM Prize in Computing, works to harness the power of visual complexity.

DESPITE the fact that he does not see very well, Alexei Efros, recipient of the 2016 ACM Prize in Computing and a professor at the University of California at Berkeley, has spent most of his career trying to understand, model, and recreate the visual world. Drawing on the massive collection of images on the Internet, he has used machine learning algorithms to manipulate objects in photographs, translate black-and-white images into color, and identify architecturally revealing details about cities. Here, he talks about harnessing the power of visual complexity.

You were born in St. Petersburg (Russia), and were 14 when you came to the U.S. What drew you to computer science?

I was interested in computers from an early age. I remember reading a book about PDP-11 assembly language programming when I was 12 and dreaming about how one day, I might actually have a computer of my own to try this out in practice. Then, in high school, I did some research with a professor at the University of Utah. It sounds kind of brazen, but I went to the CS department and was like, "Bring me to your chairman." Tom Henderson was the chair at that time and, you know, he actually saw me. I told him that I wanted to do computer science and asked him for a problem. And he basically said, "Ok, weird Russian kid. I have a robot running around; do you want to help with that?" It was wonderful.

You did your undergraduate work at the University of Utah, as well.

Interestingly enough, I was actually considering whether I should go into computer science (CS) or theater. In fact, I applied to Carnegie Mellon University because it's one of the top departments in CS, but also one of the top universities for theater. Then I showed my father the tuition, and, well, we were immigrants. So I went to the University of Utah, where CS was much stronger than theater, and I think I got a very good education. But I'm still practicing my stagecraft twice a week in my classes.

I've seen your talks. You're a very engaging speaker.

There is this whole dichotomy between the geeks and the artsy people—either you are good with numbers, or with arts and humanities. I think it's misplaced. CS is hot right now. A lot of smart kids go into CS, and many look down at all of these humanities people with disdain. In my classes, I try to remind them that computer scientists are hot now, but physicists were hot in the Sixties, and chemists were hot in the Thirties, and they're not superhot now. Shakespeare is going to be around much longer than Python.

How did you get involved with computer vision, graphics, and machine learning?

Even in high school, my goal was to solve AI. But then I reasoned it out: AI is too hard, and you don't know when you're succeeding. With language, you kind of know when you're succeeding, but that's also very high-level. Meanwhile, almost all animals have vision. Vision seems like the most basic thing, so it's got to be easy, right?

Of course.

Basically, I think I've just had one idea throughout my whole career, and I've been milking it since undergrad, and the idea is not even that profound. It's that we fetishize intellectual contributions—algorithms, data structures, and so on. And we often forget that a lot of the complexity in the world is actually due to the data. My favorite example is in computer graphics. We know how light behaves, and we can simulate everything we want. But the reason current animated movies don't look like the real thing is the data. There is a lot of entropy in the world and it's just too hard to capture. The algorithms are fine. It's the data that is miss- [CONTINUED ON P. 103]

PHOTO BY NOAH BERGER, COURTESY OF UC BERKELEY

Science Park, Amsterdam
The Netherlands

http://mmsys2018.org

June 12-15, 2018

MMSys '18

ACM Multimedia Systems Conference

Welcome submissions on

- Complete multimedia systems that provide a new kind of multimedia experience or systems whose overall performance improves the state-of-the-art through new research results in more than one component, or
- Enhancements to one or more system components that provide a documented improvement over the state-of-the-art for handling continuous media or time-dependent services.

Submission Deadline
November 30, 2017
https://submissions.mmsys2018.org

Relevant themes

Adaptive streaming, games, virtual reality, augmented reality, mixed reality, 3D video, Ultra-HD, HDR, immersive systems, plenoptics, 360° video, multimedia IoT, multi- and many-core, GPGPUs, mobile multimedia and 5G, wearable multimedia, P2P, cloud-based multimedia, cyber-physical systems, multi-sensory experiences, smart cities, QoE

Sponsors

co-Sponsors

SIGGRAPH ASIA 2017 BANGKOK

10th Edition

CONFERENCE 27 – 30 November 2017
EXHIBITION 28 – 30 November 2017
BITEC, Bangkok, Thailand

THE CELEBRATION OF LIFE & TECHNOLOGY

The 10th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia

Register online by 15 October 2017, & enjoy early bird discounts of up to 20%

SA2017.SIGGRAPH.ORG/REGISTRATION

Sponsored by

Organized by koelnmesse
we energize your business | since 1924