

COMMUNICATIONS

CACM.ACM.ORG OF THE ACM 03/2021 VOL.64 NO.03

The Decline of Computers as a General Purpose Technology

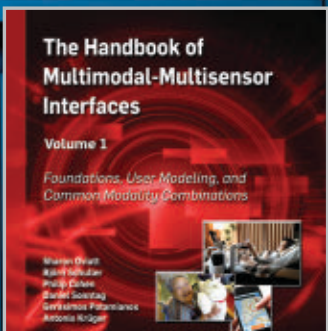
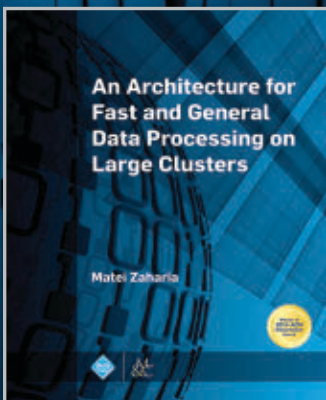
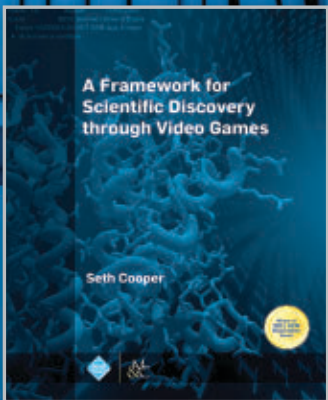
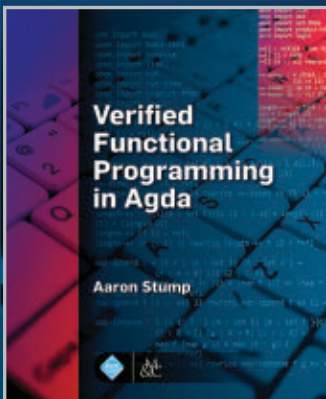
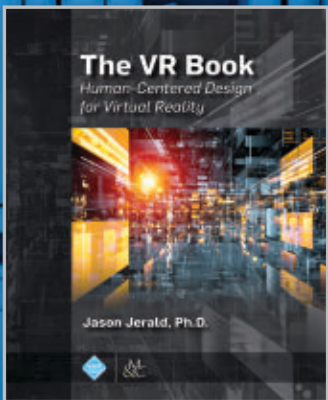
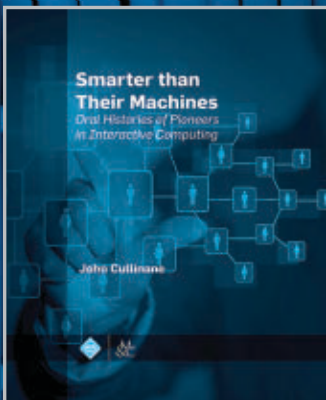
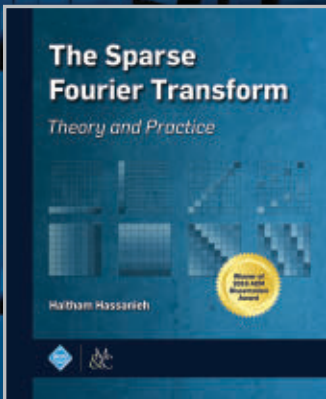
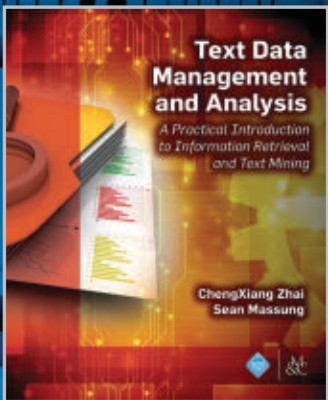
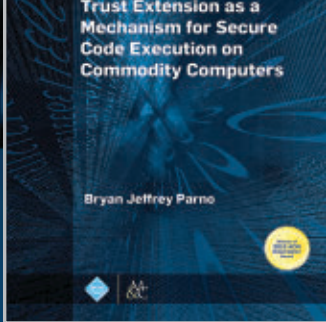
Niklaus Wirth
On Pascal at 50

Around the World (First Time)
With *Communications'*
Regional Special Sections

Cyber Reconnaissance
Techniques

Association for
Computing Machinery





In-depth. Innovative. Insightful.

Inspired by the need for high-quality computer science publishing at the graduate, faculty, and professional levels, ACM Books are affordable, current, and comprehensive in scope.

**Full Collection | Title List
Now Available**

For more information, please visit
<http://books.acm.org>



Association for Computing Machinery
1601 Broadway, 10th Floor, New York, NY 10019-7434, USA
Phone: +1-212-626-0658 Email: acmbooks-info@acm.org

A New Journal from ACM

Co-published with SAGE



Collective Intelligence, co-published by ACM and SAGE, with the collaboration of Nesta, is a global, peer-reviewed, open access journal devoted to advancing the theoretical and empirical understanding of collective performance in diverse systems, such as:

- human organizations
- hybrid AI-human teams
- computer networks
- adaptive matter
- cellular systems
- neural circuits
- animal societies
- nanobot swarms

The journal embraces a policy of creative rigor and encourages a broad-minded approach to collective performance. It welcomes perspectives that emphasize traditional views of intelligence as well as optimality, satisficing, robustness, adaptability, and wisdom.

Accepted articles will be available for free online under a Creative Commons license. Thanks to a generous sponsorship from Nesta, Article Processing Charges will be waived in the first year of publication.

For more information and to submit your work,
please visit <https://colint.acm.org>



Association for
Computing Machinery



Departments

5 **Editor's Letter**
Around the World
By *Andrew A. Chien*

9 **Vardi's Insights**
The People vs. Tech
By *Moshe Y. Vardi*

11 **Career Paths in Computing**
Enabling Renewable Energy Through Smarter Grids
By *Graham Oakes*

12 **BLOG@CACM**
Disputing Dijkstra, and Birthdays in Base 2
Mark Guzdial takes issue with Dijkstra's metaphors, while Joel C. Adams considers how birthdays might differ if based on binary numbers.

126 **Careers**

Last Byte

128 **Future Tense**
Awakening
Some technical support will never change.
By *Brian Clegg*

News

15 **The Power of Quantum Complexity**
A theorem about computations that exploit quantum mechanics challenges longstanding ideas in mathematics and physics.
By *Don Monroe*

18 **Fact-Finding Mission**
Artificial intelligence provides automatic fact-checking and fake news detection, but with limits.
By *Neil Savage*

20 **Can the Biases in Facial Recognition Be Fixed; Also, Should They?**
Many facial recognition systems used by law enforcement are shot through with biases. Can anything be done to make them fair and trustworthy?
By *Paul Marks*

23 **Edmund M. Clarke (1945–2020)**
By *Simson Garfinkel and Eugene H. Spafford*

Viewpoints

26 **Legally Speaking**
The Push for Stricter Rules for Internet Platforms
Considering the origins, interpretations, and possible changes to Communications Decency Act § 230 amid an evolving online environment.
By *Pamela Samuelson*

29 **Privacy**
Informing California Privacy Regulations with Evidence from Research
Designing and testing 'Do Not Sell My Personal Information' icons.
By *Lorrie Faith Cranor*

Viewpoints (continued)

33 **Computing Ethics**
What To Do About Deepfakes
Seeking to reap the positive uses of synthetic media while minimizing or preventing negative societal impact.
By *Deborah G. Johnson and Nicholas Diakopoulos*

36 **The Profession of IT**
Science Is Not Another Opinion
The issue is not who has the "truth," but whose claims deserve more credence.
By *Peter J. Denning and Jeffrey Johnson*

39 **Viewpoint**
50 Years of Pascal
The Pascal programming language creator Niklaus Wirth reflects on its origin, spread, and further development.
By *Niklaus Wirth*

42 **Viewpoint**
What Can the Maker Movement Teach Us About the Digitization of Creativity?
Experimenting with the creative process.
By *Sascha Friesike, Frédéric Thiesse, and George Kuk*

46 **Viewpoint**
The Transformation of Patient-Clinician Relationships with AI-based Medical Advice
A "bring your own algorithm" era in healthcare.
By *Oded Nov, Yindalon Aphinyanaphongs, Yvonne W. Lui, Devin Mann, Maurizio Porfiri, Mark Riedl, John-Ross Rizzo, and Batia Wiesenfeld*

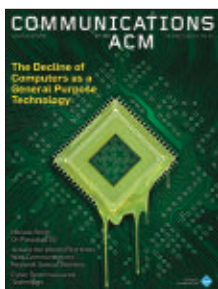
Practice



- 50 **A Second Conversation with Werner Vogels**
The Amazon CTO sits with Tom Killalea to discuss designing for evolution at scale.

- 58 **Out-of-This-World Additive Manufacturing**
From thingamabobs to rockets, 3D printing takes many forms.
By *Jessie Frazelle*

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



About the Cover:
This month's cover story traces the technological and economic forces that are now pushing computing away from being general purpose and toward specialization. Cover illustration by The Image Foundation.

Contributed Articles



- 64 **The Decline of Computers as a General Purpose Technology**
Technological and economic forces are now pushing computing away from being general purpose and toward specialization.
By *Neil C. Thompson and Svenja Spanuth*



Watch the authors discuss this work in the exclusive *Communications* video.
<https://cacm.acm.org/videos/the-decline-of-computers>

- 73 **Educational Interventions and Female Enrollment in IT Degrees**
A study of female students enrolled in IT degrees in Australia traces how programs influenced decision making.
By *Andreea Molnar, Therese Keane, and Rosemary Stockdale*
- 78 **Gender Trends in Computer Science Authorship**
Under optimistic projection models, gender parity is forecast to be reached after 2100.
By *Lucy Lu Wang, Gabriel Stanovsky, Luca Weihs, and Oren Etzioni*

Review Articles

- 86 **Cyber Reconnaissance Techniques**
The evolution of and countermeasures for ...
By *Wojciech Mazurczyk and Luca Caviglione*
- 96 **Knowledge Graphs**
Tracking the historical events that lead to the interweaving of data and knowledge.
By *Claudio Gutierrez and Juan F. Sequeda*

Research Highlights

- 106 **Technical Perspective**
Why Don't Today's Deep Nets Overfit to Their Training Data?
By *Sanjeev Arora*
- 107 **Understanding Deep Learning (Still) Requires Rethinking Generalization**
By *Rajyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals*



Watch the authors discuss this work in the exclusive *Communications* video.
<https://cacm.acm.org/videos/understanding-deep-learning>

- 116 **Technical Perspective**
Localizing Insects Outdoors
By *Prabal Dutta*
- 117 **3D Localization for Subcentimeter-Sized Devices**
By *Rajalakshmi Nandakumar, Vikram Iyer, and Shyamnath Gollakota*



COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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

Executive Director and CEO
Vicki L. Hanson
Deputy Executive Director and COO
Patricia Ryan
Director, Office of Information Systems
Wayne Graves
Director, Office of Financial Services
Darren Ramdin
Director, Office of SIG Services
Donna Cappel
Director, Office of Publications
Scott E. Delman

ACM COUNCIL
President
Gabriele Kotsis
Vice-President
Joan Feigenbaum
Secretary/Treasurer
Elisa Bertino
Past President
Cherri M. Pancake
Chair, SGB Board
Jeff Jortner
Co-Chairs, Publications Board
Jack Davidson and Joseph Konstan
Members-at-Large
Nancy M. Amato; Tom Crick;
Susan Dumais; Mehran Sahami;
Alejandro Saucedo
SGB Council Representatives
Sarita Adve and Jeanna Neefe Matthews

BOARD CHAIRS
Education Board
Mehran Sahami and Jane Chu Prey
Practitioners Board
Terry Coatta

REGIONAL COUNCIL CHAIRS
ACM Europe Council
Chris Hankin
ACM India Council
Abhiram Ranade
ACM China Council
Wenguang Chen

PUBLICATIONS BOARD
Co-Chairs
Jack Davidson and Joseph Konstan
Board Members
Jonathan Aldrich; Phoebe Ayers;
Chris Hankin; Mike Heroux; James Larus;
Tulika Mitra; Marc Najork;
Michael L. Nelson; Theo Schlossnagle;
Eugene H. Spafford; Divesh Srivastava;
Bhavani Thuraisin; Robert Walker;
Julie R. Williamson

ACM U.S. Technology Policy Office
Adam Eisgrau
Director of Global Policy and Public Affairs
1701 Pennsylvania Ave NW, Suite 200,
Washington, DC 20006 USA
T (202) 580-6555; acmpo@acm.org

Computer Science Teachers Association
Jake Baskin
Executive Director

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

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Andrew Rosenbloom
Senior Editor/News
Lawrence M. Fisher
Web Editor
David Roman
Editorial Assistant
Danbi Yu

Art Director
Andrij Borys
Associate Art Director
Margaret Gray
Assistant Art Director
Mia Angelica Balaquiot
Production Manager
Bernadette Shade
Intellectual Property Rights Coordinator
Barbara Ryan
Advertising Sales Account Manager
Ilija Rodriguez

Columnists
David Anderson; Michael Cusumano;
Peter J. Denning; Mark Guzdial;
Thomas Haigh; Leah Hoffmann; Mari Sako;
Pamela Samuelson; Marshall Van Alstyne

CONTACT POINTS
Copyright permission
permissions@hq.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmhelp@acm.org
Letters to the Editor
letters@cacm.acm.org

WEBSITE
<http://cacm.acm.org>

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

AUTHOR GUIDELINES
<http://cacm.acm.org/about-communications/author-center>

ACM ADVERTISING DEPARTMENT
1601 Broadway, 10th Floor
New York, NY 10019-7434 USA
T (212) 626-0686
F (212) 869-0481

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

Media Kit acmmédiasales@acm.org

Association for Computing Machinery (ACM)
1601 Broadway, 10th Floor
New York, NY 10019-7434 USA
T (212) 869-7440; F (212) 869-0481

EDITORIAL BOARD
EDITOR-IN-CHIEF
Andrew A. Chien
aic@cacm.acm.org
Deputy to the Editor-in-Chief
Morgan Denlow
cacm.deputy.to.eic@gmail.com
SENIOR EDITOR
Moshe Y. Vardi

NEWS
Co-Chairs
Marc Snir and Alain Chesnais
Board Members
Tom Conte; Monica Divitini; Mei Kobayashi;
Rajeev Rastogi; François Sillion

VIEWPOINTS
Co-Chairs
Tim Finin; Susanne E. Hambrusch;
John Leslie King
Board Members
Virgilio Almeida; Terry Benzel; Michael L. Best;
Judith Bishop; Lorrie Cranor; Boi Falting;
James Grimmermann; Mark Guzdial;
Haym B. Hirsch; Anupam Joshi; Richard Ladner;
Carl Landwehr; Beng Chin Ooi; Francesca Rossi;
Len Shustek; Loren Terveen; Marshall Van
Alstyne; Jeannette Wing; Susan J. Winter

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

CONTRIBUTED ARTICLES
Co-Chairs
James Larus and Gail Murphy
Board Members
Robert Austin; Kim Bruce; Alan Bundy;
Peter Buneman; Premkumar T. Devanbu;
Jane Cleland-Huang; Yannis Ioannidis;
Trent Jaeger; Somesh Jha; Gal A. Kaminka;
Ben C. Lee; Igor Markov; m.c.schraefel;
Hannes Werthner; Reinhard Wilhelm;
Rich Wolski

RESEARCH HIGHLIGHTS
Co-Chairs
Shriram Krishnamurthi
and Orna Kupferman
Board Members
Martin Abadi; Amr El Abbadi;
Animashree Anandkumar; Sanjeev Arora;
Michael Backes; Maria-Florina Balcan;
Azer Bestavros; David Brooks; Stuart K. Card;
Jon Crowcroft; Lieven Eeckhout;
Alexei Efron; Bryan Ford; Alon Halevy;
Gernot Heiser; Takeo Igarashi;
Srinivasan Keshav; Sven Koenig;
Ran Libeskind-Hadas; Karen Liu;
Tim Roughgarden; Guy Steele, Jr.;
Robert Williamson; Margaret H. Wright;
Nicolai Zeldovich; Andreas Zeller

SPECIAL SECTIONS
Co-Chairs
Sriram Rajamani, Haibo Chen,
and P. J. Narayanan
Board Members
Sue Moon; Tao Xie; Kenjiro Taura; David Padua

ACM Copyright Notice
Copyright © 2021 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from permissions@hq.acm.org or fax (212) 869-0481.

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

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

ACM Media Advertising Policy
Communications of the ACM and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current advertising rates can be found by visiting <http://www.acm-media.org> or by contacting ACM Media Sales at (212) 626-0686.

Single Copies
Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

COMMUNICATIONS OF THE ACM (ISSN 0001-0782) is published monthly by ACM Media, 1601 Broadway, 10th Floor New York, NY 10019-7434 USA. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER
Please send address changes to *Communications of the ACM* 1601 Broadway, 10th Floor New York, NY 10019-7434 USA

Printed in the USA.



Association for Computing Machinery





DOI:10.1145/3448648

Andrew A. Chien

Around the World

(the first time) with *Communications'* Regional Special Sections.



The team has navigated complex politics, regional rivalries, and a wealth of logistics challenges (including COVID-19!) to reveal remarkable creativity, technology, excellence, and unique computing culture around the globe.



IN 2017, WE made the strategic decision to launch *Communications'* Regional Special Sections (RSS) and declared “Here comes Everybody ... to *Communications*.” Next month (April 2021), we will publish the special section for Arabia, completing our circumnavigation of the world in three years. We are already hearing clamor for “we want another chance to highlight the best in our region” with a collection of Hot Topics and Big Trends.

It was not easy. The team has navigated complex politics, regional rivalries and tensions, and a wealth of logistics challenges (including COVID-19!) to reveal the remarkable creativity, technology, excellence, and unique computing culture around the globe. And while our reach has been broad and inclusive, we have by no means touched it all. There is much more to be done!

Communications' RSS global initiative's goal is to “give deeper insight, focused coverage, and elevate distinc-

tive and compelling highlights of computing drawn from regions around the world” to enhance the inclusiveness of *Communications* and the ACM. We have circled the globe, putting together six special sections focused on China, Europe, East Asia and Oceania, India, Latin America, and Arabia. We have fulfilled our promise to have each special section led by and comprised of authors from the region. We believe this approach essential to building a strong ACM community throughout the world, spanning academic research, industry, government, and beyond.

The Regional Special Sections each opened a window into a different part of the world for the readership of *Communications*, providing insights and perspective on technical, social, and cultural issues in computing. Specifically, we set out to represent the best of computing leadership and distinctive development for each region with a sharp focus on:

- ▶ Leading technical and

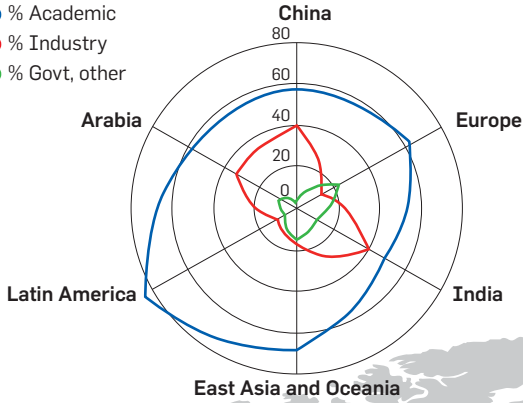


China Region Special Section

Europe Region Special Section

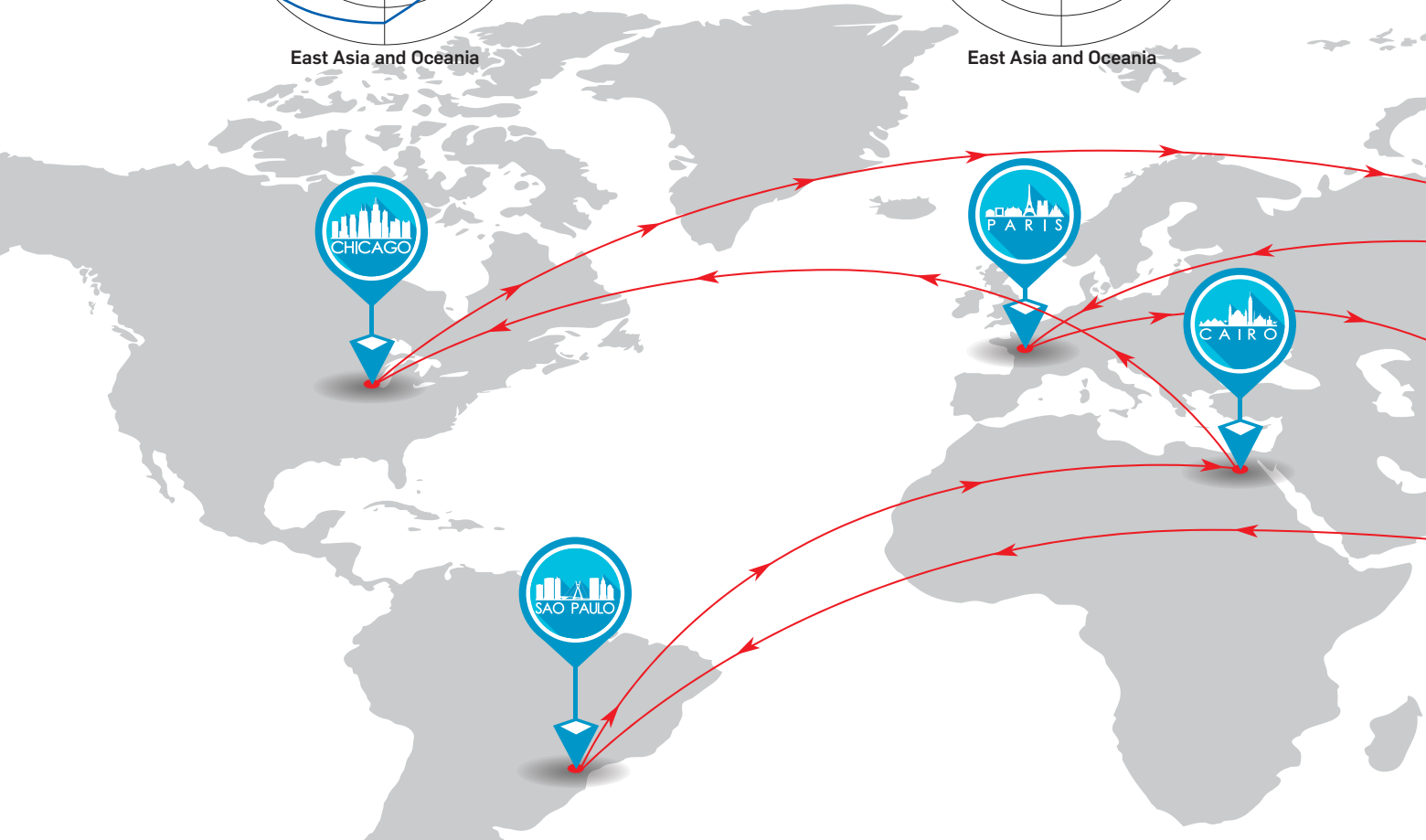
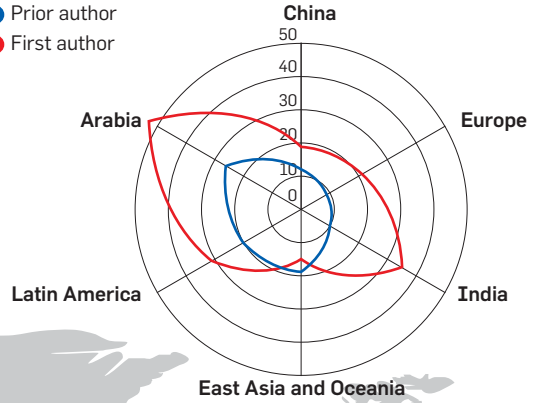
Background: academic, industry, government

- % Academic
- % Industry
- % Govt, other



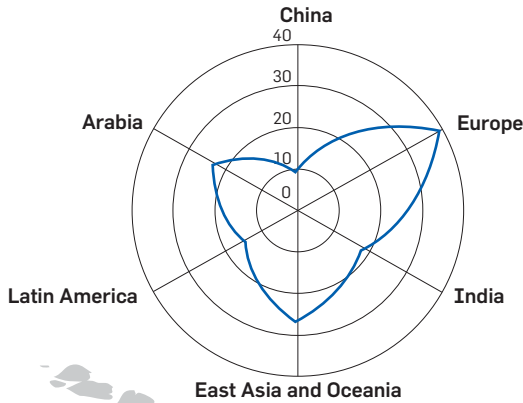
Prior author in ACM or not

- Prior author
- First author



A collage of journal covers. On the left, the 'India Region Special Section' cover features a 3D cube of images on an orange background. On the right, the 'East Asia & Oceania Region Special Section' cover features a similar 3D cube of images on a blue background. In the background, other journal covers are visible, including 'Welcome to the India Region Special Section' and 'Welcome' for the East Asia & Oceania region.

% female authorship



China's Computing Ambitions • Can China Lead the Development of Data Trading and Sharing Markets? • The European Perspective on Responsible Computing

- Women Are Needed in STEM: European Policies and Incentives
- India Stack—Digital Infrastructure as Public Good • The Positive and Negative Effects of Social Media in India
- Capturing Cultural Heritage in East Asia and Oceania • Detecting Fake News in Social Media: An Asia-Pacific Perspective
- Understanding Salsa: How Computing Is Defining Latin Music
- A Panorama of Computing in Central America and the Caribbean • Autonomous Driving in the Face of Unconventional Odds • Building a Research University in the Arab Region: The Case of KAUST

- research advances and activities;
- ▶ Leading industry and research players;
- ▶ Innovation and the shape of computing in the region; and
- ▶ Unique challenges and opportunities.

The special sections have emerged as a “multicultural splendor,” and many readers have shared their praise and compliments for what they learned in reading the special sections—in some cases even one for their own region!

The RSS's unusual diversity of topics is by design. It has a creative format, combining short articles of two formats—Hot Topics and Big Trends. But the insights and colorful perspectives were provided by the extraordinary collection of 269 authors.

As you can see, we have done well in regional diversity, and in reaching beyond the academic research community, but definitely have room for improvement in expanding the community and increasing gender diversity!

I am particularly proud to report that each of the RSSs were led by extraordinary co-leads and authors drawn exclusively from the regions. The Regional Special Section is led overall by Editorial Board co-chairs Jakob Rehof, Haibo Chen, and P J Narayanan (thanks to Sri-ram Rajamani who has recently stepped down), and the Editorial board members (David Padua, Kenjiro Taura, Sue Moon, and Tao Xie). Thanks to this team that works tirelessly to achieve excellence, inclusion, coverage, and interesting insights. Thanks to the co-leads and authors from each region whose efforts are instrumental in bringing an insightful and comprehensive perspective.

Finally, the RSSs were driven by the extraordinary efforts of Morgan Denlow and Lihan Chen; each Deputies to the EiC.

A heartfelt thanks from all of us who have benefited! Now, around again!

Andrew A. Chien, EDITOR-IN-CHIEF

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

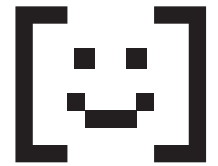
Copyright held by author/owner.

**Coming Next Month:
Arab World Special Section**



MENSCH UND COMPUTER 2021

acm In-Cooperation



With about 800 participants in the past years, “Mensch und Computer” is one of the largest conferences in the field of human-computer interaction (HCI) in Europe.

INGOLSTADT GERMANY

5th-8th September

muc2021.mensch-und-computer.de

Submission deadlines

Workshop proposals: **March 14, 2021**

Long papers/full papers: **April 11, 2021**

Workshop/tutorials (practitioner track): **April 11, 2021**

Rise into
a new future




Technische Hochschule
Ingolstadt


HUMAN-COMPUTER
INTERACTION GROUP



 GERMAN UPA
German Institute for Usability
and User Experience Professionals



Moshe Y. Vardi

DOI:10.1145/3446769

The People vs. Tech

SIZE MATTERS. Today, the top seven companies in the S&P 500 index are all tech companies. Large companies wield power, and that power often leads to a clash between these companies and “The People,” that is, with governments. This clash has been imminent. In January 2019, I wrote in this column: “If society finds the surveillance business model offensive, then the remedy is public policy, in the form of laws and regulations, rather than an ethics outrage.” In November 2019, I wrote: “What may have been a radical position less than a year ago has become a conventional wisdom now. There have been initiatives to regulate big tech and the question is how rather than if.” I also quoted legal scholar Tim Wu’s 2018 book, *The Curse of Bigness: Antitrust in the New Gilded Age*, where it is argued that the government must enforce anti-trust laws. Now we have a flurry of lawsuits by governments against tech companies, described by the media as “a stunning reversal of fortunes for Silicon Valley.”^a

Attorneys general in more than 30 U.S. states launched a lawsuit against Google in December 2020. They accused Google of an illegal monopoly in its search business. This is the third government lawsuit against Google. It follows two suits filed by the U.S. Federal Trade Commission and 48 states against Facebook for abusing its power in social networking. Action is not limited to the U.S.; in December, Chinese regulatory agencies announced scrutiny of Chinese tech giants Ali Baba and Ant, following European Union anti-trust charges against Amazon.

The effort of the people to control large corporations is over a century old.

The U.S. Sherman Antitrust Act of 1890^b aims at ensuring competition in commerce. According to the U.S. Supreme Court, the act is to protect people from market failure: “The law directs itself ... against conduct which unfairly tends to destroy competition itself.” Most major applications of the Sherman Act have often been aimed at “big tech” of the time. In the late 19th century this was railroad tech, and the Sherman Act was aimed at busting railroad cartels. In the early 20th century, it was oil tech, as when President Theodore Roosevelt used the Act to break up the monopolistic oil giant Standard Oil.

Anti-trust actions aimed at communication and computing companies—against AT&T, IBM, and Microsoft—played a crucial role over the past 50 years in shaping today’s tech industry. In the 1970s, the U.S. argued that AT&T was using monopoly profits from its Western Electric subsidiary to subsidize the costs of its network, which was contrary to U.S. antitrust law. The case was settled in 1982, which led to the 1984 breakup of the old AT&T into new, seven regional Bell operating companies and the much smaller new AT&T (which has since been acquired by Southwestern Bell). Without this breakup, the Internet of today would likely have been run solely by what was known as “The Phone Company.”

Throughout the 20th century the U.S. government repeatedly clashed with IBM. In 1936, IBM was forced to no longer require only IBM-made punch cards, and to assist alternative suppliers of cards with competing production facilities. In 1956, IBM was forced again to allow more competition in the data-processing industry. Following long-running (1969–1982) U.S. anti-

trust action, IBM softened its anti-competitive conduct in ways that probably stopped it from buying Intel and Microsoft in the 1980s—two critical suppliers of the IBM PC—who ultimately controlled the PC platform.

In 1998, the U.S. used anti-trust law to accuse Microsoft of maintaining a monopoly position in the PC market. The U.S. prevailed in the trial, but Microsoft won on appeal. The final 2001 settlement required Microsoft to share application programming interfaces with third-party companies, as well as other softening of Microsoft’s anti-competitive conduct, which, arguably, enabled Google and Facebook to grow and become “tech giants.”

The issue has always been “large,” not “tech,” but the connection between large size and tech stands out.^c In 1901, President Roosevelt asked the U.S. Congress to curb the power of trusts because of their size: “Great corporations exist only because they are created and safeguarded by our institutions,” he said, adding that it is “our right and our duty to see that they work in harmony with these institutions.” Anti-trust law enforcement has served us well over the past 130 years. With market capitalization of the top five Big Tech corporations now at over USD7T,^d the people, working through governments, are carrying on this anti-trust law legacy. It should be welcomed!

Follow me on Facebook and Twitter.

^c <https://cacm.acm.org/magazines/2019/11/240377-the-winner-takes-all-tech-corporation/fulltext>

^d <https://finance.yahoo.com/?guccounter=1>

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

^b <https://www.americanhistoryusa.com/topic/sherman-antitrust-act/>

Copyright held by author.

^a <http://bit.ly/3oeXZ2Y>

The Essentials of Modern Software Engineering

Free the Practices from the Method Prisons!

This text/reference is an in-depth introduction to the systematic, universal software engineering kernel known as “Essence.” This kernel was envisioned and originally created by Ivar Jacobson and his colleagues, developed by Software Engineering Method and Theory (SEMAT) and approved by The Object Management Group (OMG) as a standard in 2014. Essence is a practice-independent framework for thinking and reasoning about the practices we have and the practices we need. **It establishes a shared and standard understanding of what is at the heart of software development. Essence is agnostic to any particular methods, lifecycle independent, programming language independent, concise, scalable, extensible, and formally specified.** Essence frees the practices from their method prisons.

HIGH PRAISE FOR THE ESSENTIALS OF MODERN SOFTWARE ENGINEERING

“Essence is an important breakthrough in understanding the meaning of software engineering. It is a key contribution to the development of our discipline and I’m confident that this book will demonstrate the value of Essence to a wider audience. It too is an idea whose time has come.” – Ian Somerville, St. Andrews University, Scotland (author of *Software Engineering, 10th Edition*, Pearson)

“What you hold in your hands (or on your computer or tablet if you are so inclined) represents the deep thinking and broad experience of the authors, information you’ll find approachable, understandable, and, most importantly, actionable.”
– Grady Booch, IBM Fellow, ACM Fellow, IEEE Fellow, BCS Ada Lovelace Award, and IEEE Computer Pioneer

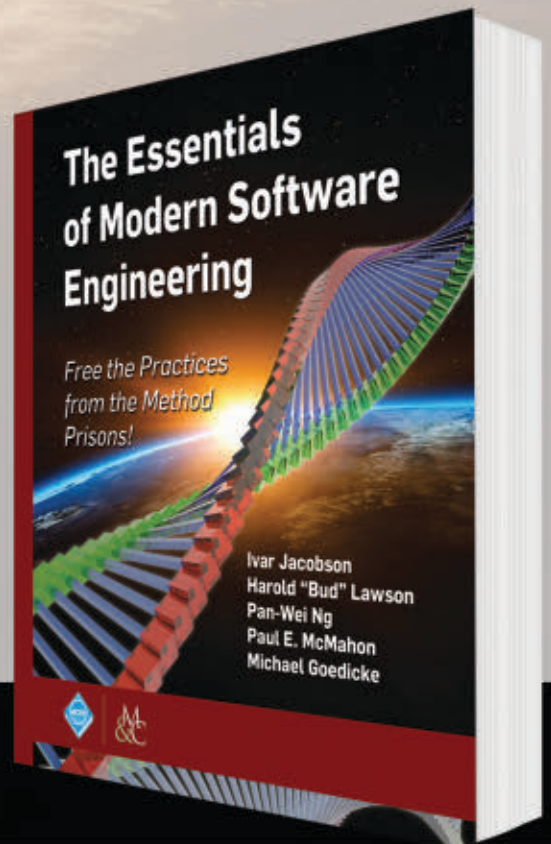
**Ivar Jacobson, Harold “Bud” Lawson,
Pan-Wei Ng, Paul E. McMahon,
Michael Goedicke**

ISBN: 978-1-947487-24-6

DOI: 10.1145/3277669

<http://books.acm.org>

<http://store.morganclaypool.com/acm>



ACM BOOKS
Collection I



CAREER PATHS IN COMPUTING

DOI:10.1145/3441558

Computing enabled me to . . .

Enabling Renewable Energy Through Smarter Grids



NAME

Graham Oakes

BACKGROUND

**Born and raised in Australia.
Moved to UK to do his PhD
and got stuck there.**

CURRENT JOB TITLE/EMPLOYER

**Independent Consultant
supporting local and municipal
energy projects in U.K. and EU.**

EDUCATION

**B.Sc.App, University of
Queensland, Australia
Ph.D. Geophysics and
Remote Sensing, Imperial
College, London, U.K.**

EVERY DAY IN the United Kingdom hundreds of power stations, thousands of substations, and millions of kilometers of cable come together to power our homes, offices, transport, and more. Yet we only take notice of this when something goes wrong. Now that's happening at a wider scale—global warming, forest fires, extreme weather events, and so on.

We must stop burning stuff in large power plants and make more use of renewables. Yet, the sun won't shine more brightly, nor the wind blow more fiercely

just because we have hit a light switch—the underlying control assumption for the entire system is different.

I came into the energy sector in 2013, when Nesta, an innovation foundation in the U.K., ran a challenge prize to engage domestic consumers with “demand side response,” that is, adjusting their energy consumption to match the generation available. This is critical if we are to integrate high proportions of solar and wind energy onto the grid. I suggested using the cloud to coordinate thousands of smart, IoT-connected devices. (We subsequently patented aspects of this idea, about how we scaled it to handle huge numbers of devices in real time.)

During the course of the challenge, I joined up with three computer science graduates from Lancaster University to build a prototype. We also submitted grant applications to the U.K. government. After winning a couple of those, we figured there must be something to our idea and we formed a company: Upside Energy. Things snowballed from there. Over the next four years, Upside grew to employ 35 software engineers and data scientists and raised £10m of grant and equity finance. Together we built a cloud service capable of coordinating hundreds of thousands of domestic appliances, home batteries, electric vehicles, among others, to make the best use of renewable energy on the grid.

I stayed with Upside through this growth and financing, helping it recruit a management team after the Series A venture capital funding in late 2017. Once that team was in place, I handed over the company and got back to my passion—helping people use technology to create a better world. I'm now supporting a

number of local and municipal energy projects in the U.K. and EU, mentoring a couple of energy-tech startups, and working on regulatory initiatives that are helping to define our path to a net-zero energy system.

All of this was possible because of my 30 years' experience as a systems engineer. I started in image processing, then data acquisition and control for large, experimental scientific systems, and then on to commercial command-and-control systems. Eventually, I shifted into games. (Some people think this is a big shift, but graphics is mostly just image processing in reverse.) From there I went into consulting for organizations like Greenpeace, Oxfam, Cisco, Intel, Skype, and Vodafone (on developing software-intensive products). All of which gave me the building blocks to found a startup.

Why do I do this? What gets me out of bed in the morning?

Three things: First, it's the opportunity to build something beautiful. Sometimes the beauty is hidden from all but the specialist's view, but it's there.

Second, I want to make the world a better place. We have enormous power to change the world through the systems we create. A world with clean air and water in which every person is respected seems like something we can all aspire to. Let's use our power to build something we can be proud of as we pass it on to coming generations.

Third, it's about welcoming more and more computer scientists into the industry. The depth of talent and passion that is coming through in our teenagers and 20-year-olds is exciting and fills me with hope—even in these times of climate crisis and Covid-19. **Q**

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:10.1145/3446806

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

Disputing Dijkstra, and Birthdays in Base 2

Mark Guzdial takes issue with Dijkstra's metaphors, while Joel C. Adams considers how birthdays might differ if based on binary numbers.



**Mark Guzdial
Dijkstra Was Wrong
About 'Radical
Novelty': Metaphors
in CS Education**

<http://bit.ly/35dg21S>

November 30, 2020

Edsger Dijkstra's 1988 paper "On the Cruelty of Really Teaching Computer Science" (in plain text form at <https://bit.ly/3b6bFto>) is one of the most well-cited papers on computer science (CS) education. It is also wrong. A growing body of recent research explores the very topic that Dijkstra tried to warn us away from—how we learn and teach computer science with metaphor.

According to Google Scholar, Dijkstra's paper has been cited 571 times. In contrast, the most-cited paper in all of the ACM Digital Library papers related to SIGCSE has 412 citations (see data at <https://bit.ly/3bae0Ub>). Dijkstra's paper has been cited more than any peer-reviewed CS education research. Many of these citations might be citing the "cruelty" paper as a foil, like Owen Astrachan's "On the Cruelty of Really Teaching Computer Science redux" (<https://bit.ly/3pSXSKI>).

Dijkstra's argument is that computers represent "radical novelty." There's nothing like them in human experience, and we cannot use our past experience to understand them. In particular, we shouldn't use metaphors.

"It is the most common way of trying to cope with novelty: by means of metaphors and analogies we try to link the new to the old, the novel to the familiar. Under sufficiently slow and gradual change, it works reasonably well; in the case of a sharp discontinuity, however, the method breaks down: though we may glorify it with the name 'common sense,' our past experience is no longer relevant, the analogies become too shallow, and the metaphors become more misleading than illuminating. This is the situation that is characteristic for the "radical" novelty.

"Coping with radical novelty requires an orthogonal method. One must consider one's own past, the experiences collected, and the habits formed in it as an unfortunate accident of history, and one has to approach

the radical novelty with a blank mind, consciously refusing to try to link it with what is already familiar, because the familiar is hopelessly inadequate."

We now know this is likely impossible. The learning sciences tell us all learning is based on connecting new experiences to previous, through a process called *constructivism* developed by Jean Piaget (see a nice explanation at <http://bit.ly/3oiCZZ8>). Trying to learn something *without* connection to prior experience inhibits learning. It leads to a phenomenon called *inert knowledge* (<http://bit.ly/3oiCZZ8>) where you have memorized stuff to pass the test, but you don't really understand and can't really use the knowledge.

I never really thought much about the metaphors we use to learn and teach computer science until the SIGCSE 2014 paper "Metaphors we teach by" (<https://bit.ly/3pQ9bn1>). CS teachers and students have been ignoring Dijkstra's admonitions all along. They teach with a variety of metaphors, and though all of them have limitations (Dijkstra was right about that), this paper explored how teachers dealt with the breaking point.

The 2019 paper "Identifying embodied metaphors for computing education" (<https://bit.ly/3od9uI9>) goes a step further to focus on the metaphors that are based on physicality. From a "radical novelty" perspective, this may seem ridiculous—nothing could be less physical than ideas like "arrays" and "control flow." But from a "constructivism" perspective, nothing could be more natural. The basis for all our

experiences are being physical beings in a physical world. When we're dealing with new ideas, we will likely relate them to physical processes.

I am working with Ph.D. student Amber Solomon, who has been studying how teachers teach recursion and how students learn it. She had a paper last summer at the 2020 International Conference of the Learning Sciences about the embodied metaphors that teachers use when teaching recursion (see summary at <http://bit.ly/3ogO9xq>). Teachers gesture and point, but it's not clear to what. They talk about being "here" and "going." They use language that suggests metaphors like the program "says" something.

Solomon is co-advised by Betsy DiSalvo and myself. The three of us have been spending time coding her videos of CS students understanding and modifying programs that use recursion. These are absolutely fascinating, and once you start looking for metaphors and uses of embodiment, you see it everywhere. I particularly like how students shift metaphors, such as talking about the recursive function "going" and then being "stopped" by the base case, then talking about "going down" the stack and execution being different "on the way back up." We know that there is no "down," "back," or "up" in a computer process—these are examples of using concepts from our everyday physical world to understand computational processes.

In 1988 when Dijkstra wrote this piece, cognitive science journals were only about a decade old, and learning sciences was not established until the 1990s. It is understandable that Dijkstra might not have known about constructivism. Today, we know constructivism as the most widely-accepted theory of how humans learn. Using a constructivist lens on learning about computing, we can better understand how to help students use their everyday knowledge as metaphors to learn computer science.



Joel C. Adams Birthday Bit Boundaries

<http://bit.ly/38gYp3p>

December 1, 2020

My family and I recently celebrated my 63rd birthday. As we were eating dinner that

night, one of my sons asked if I had anything special planned for this upcoming year. I hadn't given next year much thought, but since 63_{10} is 111111_2 , it occurred to me that this was my last birthday for which my age can be represented in six bits, as it will take seven bits (1000000_2) to represent my age when I turn 64. When I mentioned this, it triggered a surprisingly long and whimsical discussion. (My sons have both graduated with CS degrees and my wife teaches statistics, so...) Some of the points raised during that discussion included:

- We might define a *birthday bit boundary* as a birthday that requires an additional bit to represent one's new age. On my next birthday, I will cross a birthday bit boundary when my age changes from 111111_2 (63) to 1000000_2 (64).

- After birthday #64, my next possible birthday bit boundary would be #128. According to the *Guinness Book of World Records*, the most long-lived person on record was Jean Calment of France, who was 122 when she died in 1997. With no intention of being morbid, barring a medical longevity breakthrough, #64 will almost certainly be the last time I cross a birthday bit boundary.

- Our culture places special emphasis on some birthdays. Often these are multiples of 10 (like 30, 40, 50, 60, ...), presumably because our culture primarily uses decimal numbers. What birthdays would be deemed special if we used a different number system, such as base 12?

- A few other birthdays also receive special attention, such as #12 in some cultures, or "Sweet Sixteen" in popular U.S. culture.

- My previous birthday bit boundary—#32—is quite close to 30, which is commonly regarded as the threshold-age separating youth from non-youth (as in, "never trust anyone over 30"). Why not use 32 instead of 30 as that threshold?

- Each birthday bit boundary—#2, #4, #8, #16, #32, #64—is reasonably close to a key threshold in one's life stages. If our culture were based on binary numbers instead of decimal numbers, might we celebrate these birthdays as having special significance?

If we were to celebrate birthday bit boundaries as the entry points to new life stages, the table here shows the result.

Decimal Age	Binary Age	Life Stage
0	0	Infant
1	1	
2	10	Toddler
3	11	
4	100	Child
7	111	
8	1000	Adolescent
15	1111	
16	10000	Adult
31	11111	
32	100000	Middle Age
63	111111	
64	1000000	Senior Citizen
127	1111111	

In this table, the bit-boundary ages map surprisingly well to the start of significant life-stage transitions. For example, the start of adolescence is often associated with the onset of puberty, which can occur anytime in the age-range 8–14. In many U.S. states, teenagers can get their driver licenses at 16, marking their transition to adulthood.

Likewise, in the U.S., 60–65 is commonly thought of as the age at which one becomes a senior citizen, and 65 has long been thought of as the typical "retirement" age. However, 65 seems fairly arbitrary; 64 is obviously close by and might be used instead.

As a result of our family discussion, I've decided to: (i) declare my next birthday (#64) to be one of extra-special significance, and (ii) hold a special party to celebrate my crossing of this final birthday bit boundary. Assuming, of course, that I am still around.

If you have read this far, you may well be thinking that this seems like an especially geeky idea. You may even think this seems like evidence of encroaching elderly eccentricity. This would be difficult to dispute.

However, before you render a final judgment, it is worth noting there is a well-known Beatles song about reaching old age, and the title of that song is not "When I'm Sixty Five," but rather "When I'm Sixty Four"!

Mark Guzdial is professor of electrical engineering and computer science in the College of Engineering, and professor of information in the School of Information, of the University of Michigan. Joel C. Adams is a professor of computer science at Calvin University.

Publish Your Work Open Access With ACM!

ACM offers a variety of Open Access publishing options to ensure that your work is disseminated to the widest possible readership of computer scientists around the world.



Please visit ACM's website to learn more about ACM's innovative approach to Open Access at:
<https://www.acm.org/openaccess>



Association for
Computing Machinery

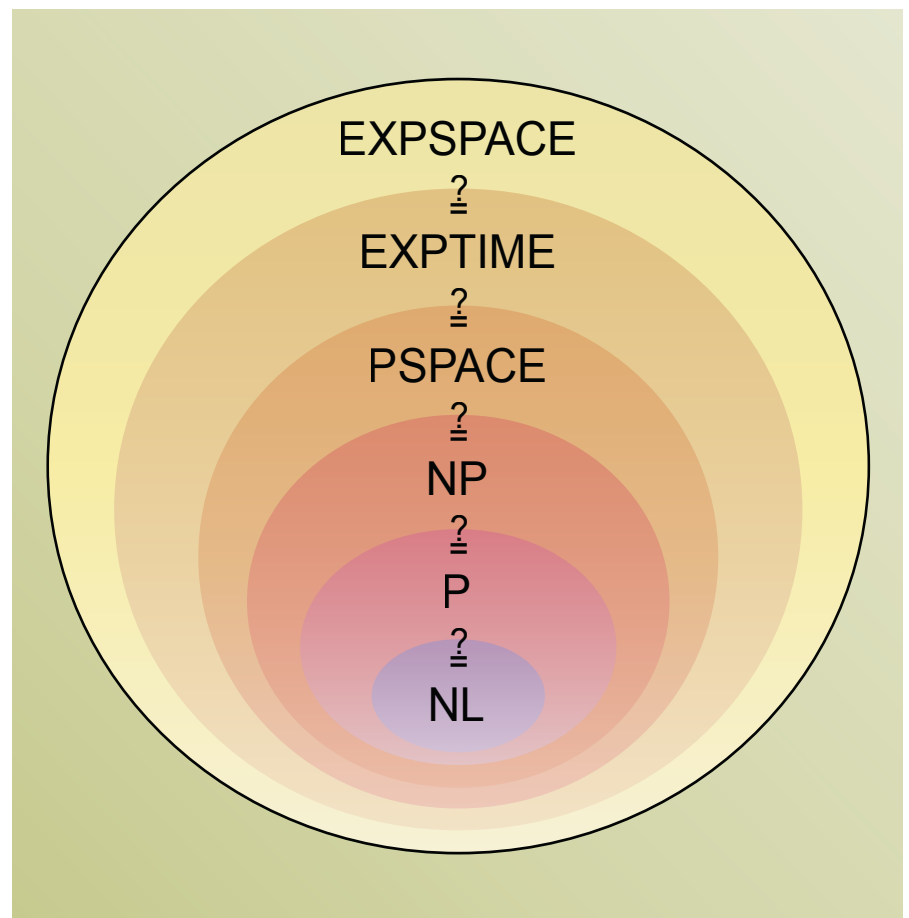
The Power of Quantum Complexity

A theorem about computations that exploit quantum mechanics challenges longstanding ideas in mathematics and physics.

FOR DECADES, COMPUTER scientists have compared the fundamental difficulty of solving various tasks, such as factoring a number or finding the most efficient route for a traveling salesperson. Along this way, they have described an alphabet soup of computational complexity classes and formal techniques for showing how various classes relate to each other.

The advent of quantum computers has introduced new flavors into such classification. It also has given urgency to understanding the potential of these still-limited machines, including the role of mysterious correlations of distant particles, known as entanglement. A recent manuscript concludes that incorporating entanglement into a well-known framework could allow verification of a staggering range of proofs, no matter how long they are.

The new result was first posted on a preprint server in January 2020, and immediately stimulated chatter in the vibrant computational-complexity blogosphere, but it has not yet been peer reviewed. Indeed, the authors already have identified a flaw in an earlier paper they built upon, although



A representation of the relationship among complexity classes, which are subsets of each other.

they later devised an alternate argument that left their conclusions intact.

If it stands up, the proof also will disprove a longstanding mathematical conjecture, with profound implications for both pure mathematics and physics. As a result, “Quite a few people take this to be true and are trying to follow up on it without really fully understanding it,” said Vern Paulsen, a mathematician at the University of Waterloo, Ontario, who was not involved in the work. “It just shows how mainstream to the world of science computational complexity is.”

“At some point, people who are working far away from computer science will find another proof of this result in their own language,” said Henry Yuen of the University of Toronto. For now, however, “There’s a lot of computer science concepts that lend themselves very naturally to putting the pieces together.” Yuen co-authored the new paper with fellow computer scientists Zhengfeng Ji of the University of Technology, Sydney; Anand Natarajan and Thomas Vidick of the California Institute of Technology, and John Wright of the University of Texas, Austin.

Enlisting Omnipotence

Computational-complexity theory classifies problems in terms of the resources, such as time, circuit size, or memory, needed to solve them. The complexity class P , for example, can be solved in a time that is only a polynomial function (some power) of the size of the problem.

In contrast, such efficient solutions are not known for problems in the class NP , such as the “traveling salesman” problem. However, for these problems, if a solution were somehow provided, it would require only polynomial time to verify it. A major open question is whether having such efficient verifiability implies some as-yet-undiscovered efficient way to find solutions, which would mean $P=NP$, although this is considered unlikely.

For non-specialists, the seemingly magical appearance of a solution may seem an odd ingredient for a mathematical proof. For decades, however, complexity theorists have considered infinitely powerful “provers,” not to re-

veal solutions but to interact with a “verifier” that has more limited computational power. The prover’s answers—aimed to convince the verifier of the proof—are not necessarily trusted by the verifier, which can cross-check the responses to randomly selected questions. Granting the imaginary prover infinite power gives the system the credibility to prove a negative, Yuen said. “If even an infinitely powerful person couldn’t convince you of a statement, then that statement must have been false.”

These techniques are amazingly powerful. Work in the 1980s showed that, even when requiring only polynomial-time verification, “Interactive proofs are equal to a complexity class called $PSPACE$,” said Lance Fortnow, dean of the College of Science of the Illinois Institute of Technology. That complexity class includes “everything you can do with a small [polynomial] amount of memory,” even in exponential time.

In 1991, Fortnow and two colleagues examined a further extension proposed a few years earlier: multiple provers, that are isolated to prevent them from coordinating their answers. Fortnow likens this to questioning a couple claiming they are married, for immigration. “You can put them in separate rooms, and ask them questions like ‘What side of the bed do you sleep on?’” By interacting with such multiple provers, a verifier can do in polynomial time what would otherwise require an exponentially longer time.

The new result adds another feature to this scheme: although the provers cannot share information about

“You have to deduce whether the entire proof is a valid proof or not. The PCP theorem says that this is possible, which is really astounding.”

what they are asked, they share access to an infinite supply of entangled quantum bits, or qubits. “I would have guessed they could possibly use it to cheat, and that it would actually make the model weaker,” said Fortnow. “Surprisingly, it’s much stronger.” As a result, this MIP^* class (multiple independent provers, with the asterisk indicating access to entanglement) can verify a proof of “basically any size,” a huge complexity class denoted RE , for “recursively enumerable.”

Checking the Checkers

As it turns out, this conclusion contradicts a widely influential conjecture in mathematics. Unfortunately, the proof itself is long, currently 206 pages, and relies heavily on techniques from computational complexity theory that are unfamiliar to mathematicians. “Mathematicians have a very strong sense of what a proof should look like, and what’s a deep proof,” said Vidick, adding that they do not know what to make of this one. When people ask him what is deep in this result, he says, “There’s the PCP theorem, and that’s it.”

PCP stands for “probabilistically checkable proof,” and the theorem, which built on the study of multiple provers, is an established “crown jewel” of computational complexity, Yuen said. A verifier is asked to check a long proof by looking at only a handful of spots, which can be chosen at random. “You have to deduce whether the entire proof is a valid proof or not. The PCP theorem says that this is possible, which is really astounding.”

To exploit entanglement, the researchers first must prevent the provers from using it to coordinate their answers. “If you devise your game in a clever-enough way, you can actually detect any time your provers are trying to use entanglement to trick you,” Wright said. Then, “We use this entanglement to generate big random strings for the two provers, and these random strings are then viewed as questions for a PCP protocol. These questions index different parts of their computation, and you can use them to check whether they’re carrying out a giant computation correctly for you.”

“The verifiers become like a puppet master for these wildly powerful, infi-

nately adversarial provers,” Yuen said. Then “the verifier checks that these two provers themselves are interrogating two more complex provers, that are interrogating two more complex provers, and so forth,” Vidick said.

This recursive compression scheme can address a wide array of problems, including whether a program will ever terminate. Alan Turing proved that this “halting problem” is undecidable in general, but, amazingly, MIP^* can verify a solution.

Beyond Complexity

The significance of the new result extends far beyond computational complexity, to other areas of mathematics and even to the understanding of quantum mechanics. This is because it is inconsistent with a longstanding conjecture made by Alain Connes, who received the prestigious Fields Medal in 1982 for his extension of John Von Neumann’s classification of “operator algebras.” As part of that program, Connes suggested the set of infinite matrices, or factors, that define some algebras could all be approximated by a consistent scheme of finite matrices, in what has become known as his embedding conjecture.

“What’s amazing is the relatively innocent conjecture Connes made back in the Seventies turned out to have so many implications, if it was true,” Paulsen said, even in seemingly unrelated fields like group theory, or models of entropy.

In recent years, though, it had begun to seem “a little too good to be true,” he noted. “Now all that’s wiped out. We’re back to ground zero,” Paulsen said. Still, the proof only shows that exceptions to the conjecture can exist. It does not provide much guidance for finding them, so there is much work to be done.

Disproving Connes’ embedding conjecture also has implications for the mathematical representation of quantum entanglement, distinct from its role in MIP^* . When two particles are entangled, for example because they emerge from a single quantum process, their properties remain correlated even if they become widely separated. Moreover, as shown by John Bell in 1964, the outcomes from some combinations of measure-

When two particles are entangled because they emerge from a single quantum process, their properties remain correlated even if they are separated.

ments on the two entangled particles are more correlated than would be possible if the particles “knew” the outcome to every possible measurement beforehand.

Nonetheless, physicists tend to describe the pair of measurements as a “tensor product” of two measurements, rather than requiring a comprehensive description of the entire system. About a decade ago, it was shown that Connes’ embedding conjecture is equivalent to the assertion, formalized by mathematician Boris Tsirelson, that the representations match up, even for complex combinations of measurements. (Paulsen and his fellow mathematicians carefully distinguish a third description.) The new proof shows that this assumption is false. As a result, “we don’t know which of these mathematical models is really the one that represents physical reality,” said Paulsen. **Q**

Further Reading

Ji, Z., Natarajan, A., Vidick, T., Wright, J., and Yuen, H.
 $MIP^*=RE$, <https://arxiv.org/abs/2001.04383>

M. Junge et al.
Connes’ embedding problem and Tsirelson’s problem, *J. Math. Phys.* 52, 012102 (2011)

Hartnett, K.
Landmark Computer Science Proof Cascades Through Physics and Math, *Quanta*, March 4, 2020.

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

ACM Member News

COMPUTATIONAL ANALYSIS OF MICROBIAL COMMUNITIES



Mihai Pop is a professor in the Department of Computer Science at the University of Maryland in

College Park, MD, where he also serves as director of the University of Maryland Institute for Advanced Computer Studies.

Pop earned his undergraduate degree in computer science from Romania’s Politehnica University of Bucharest, and his master’s and Ph.D. degrees (both in computer science) from Johns Hopkins University in Baltimore, MD.

On completing his Ph.D. in 2000, Pop went to work as a bioinformatics scientist at the Institute for Genomic Research in Rockville, MD. He joined the faculty of the University of Maryland in 2005, and has remained there ever since.

Much of Pop’s recent research has focused on computational analysis of microbial communities, a scientific field called metagenomics, with a particular focus on the microbes that inhabit the human body.

His current research centers on understanding strain variations, looking at organisms that are closely related but have key differences.

In the future, Pop is interested in working on how to integrate different types of information about microbial communities, to gain a better understanding of what’s going on. “The next step is, how do you integrate those signatures with other types of data to understand what it really means, in terms of what those microbial communities actually do,” he explains.

Pop also has an eye toward building diversity in computer science, to promote a more inclusive environment. “There is a great need and opportunity to broaden the field, and I am very passionate about that,” he says.

—John Delaney

Fact-Finding Mission

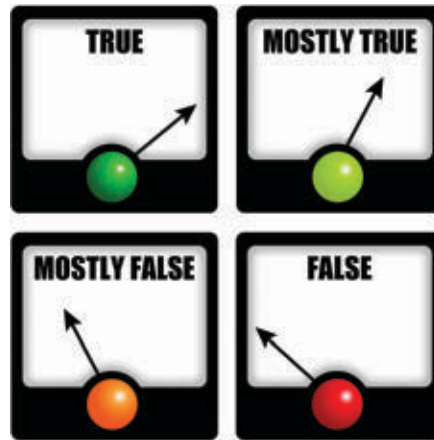
Artificial intelligence provides automatic fact-checking and fake news detection, but with limits.

SEEKING TO CALL into question the mental acuity of his opponent, Donald Trump looked across the presidential debate stage at Joseph Biden and said, “So you said you went to Delaware State, but you forgot the name of your college. You didn’t go to Delaware State.”

Biden chuckled, but viewers may have been left wondering: did the former vice president misstate where he went to school? Those who viewed the debate live on an app from the London-based company Logically were quickly served an answer: the president’s assertion was false. A brief write-up posted on the company’s website the next morning provided links to other fact-checks from National Public Radio and the *Delaware News Journal* on the same claim, which explain that Biden actually said his first Senate campaign received a boost from students at the school.

Logically is one of a number of efforts, both commercial and academic, to apply techniques of artificial intelligence (AI), including machine learning and natural language processing (NLP), to identify false or misleading information. Some focus their efforts on automating fact-checking to verify the claims in news stories or political speeches, while others try to root out fake news deployed on social media and websites to deliberately mislead people.

While 2020, with its U.S. presidential election and a global pandemic, provided plenty of fodder for fake news, the problem is not new. A 2018 study from the Massachusetts Institute of Technology’s Media Lab found false news stories on Twitter were 70% more likely to be retweeted than true ones, and that true stories take about six times as long to reach 1,500 people as false ones. In April 2020, Facebook—combining AI with the work of more than 60 fact-checking organizations in more than 50



languages—placed warning labels on 50 million pieces of content related to COVID-19.

Logically relies heavily on a team of human fact-checkers, who examine perhaps 300 claims each day, says Anil Bandhakavi, head of data science at the company. Those people seek out sources that allow them to label an assertion as true, false, or partially true, and add those assessments to a database. The software examines text or speech to automatically extract claims, and groups similar claims into clusters. Once the humans have ruled on the veracity of one of those claims, that ruling is propagated to the rest of the claims in the cluster, thus quickly expanding the universe of examined claims. “In that way, we are constantly growing our database of facts by this semi-automated process,” Bandhakavi says. Humans carry about 60% of the Logically workload, but Bandhakavi hopes that will shift more to computers over time.

The company also uses fairly common techniques to determine where the content comes from and how it propagates through the network, tracing it to its source domain and determining which other domains that source links to and which link to it. If a domain is the source of a lot of stories that have been deemed to be untrustworthy, or it passes a lot of content among other less-credible sites, then

new content from that same source will be considered questionable, too. Content from a respected news source will score better for credibility.

At the same time, Logically’s software also is learning on its own to tell truth from fiction, using NLP to develop statistical descriptions of factual and non-factual statements and how they differ from one another. Bandhavi says the software can examine the style of language used in conveying falsehoods, and distinguish it from language used for conveying facts.

Such style-based examinations also can help an AI algorithm distinguish between content written by a human and that produced by a machine. Computer scientists worry about so-called ‘neural fake news’, which uses language models developed by neural networks to produce convincing stories, mimicking the style of particular news outlets and adding bylines that make those outlets look like the source of the stories.

Learning by Doing

Researchers at the University of Washington’s Paul G. Allen School of Computer Science and Engineering developed an algorithm called Grover to both generate and detect neural fake news. Grover uses a generative adversarial network. One part, the adversary, which is trained on a collection of real news stories, learns to generate fake stories from a prompt, such as the headline “Research Shows Vaccines Cause Autism.” A second system, the verifier, is given an unlimited set of real news stories, plus fake stories from the adversary, and has to determine which are false. Based on the verifier’s results, the adversary tries again, and through repeated iterations both get better at their tasks.

With moderate training, Grover learned to distinguish neural fake news from human-written news with 71% accuracy. It did even better at detecting the fake news it generated itself, with an accuracy rate of 92%.

Grover is built on the same concept as other language modeling algorithms, such as Google’s Bidirectional Encoder Representations from Transformers (BERT) or Open AI’s Generative Pre-trained Transformer (GPT), which produce text that appears written

by humans. When Open AI produced its second iteration, GPT-2, it opted not to release it, saying the potential to create fake news was too dangerous. The company has since developed GPT-3, and while not fully releasing that, it has provided access to an application programming interface.

Just trying to keep such tools out of the hands of bad actors is not enough, says Franziska Roesner, a specialist on computational threat modeling at the University of Washington who took part in the Grover research. The researchers made their work available to help researchers understand how advances in language modeling algorithms can produce fake news and how they might detect it, “If one way of generating fake news is to do it automatically, then we have to assume that our adversaries are going to be doing that and they’re going to be training stronger models,” she says. “Security through obscurity is not ultimately effective.”

Speeding Verification

At the Duke University Reporter’s Lab, researchers string together a number of techniques to provide real-time fact checking on events such as presidential debates. They feed debate audio to Google’s speech-to-text tool, which uses machine learning to automatically transcribe the speech. They then hand off the text to ClaimBuster, an NLP system developed at the University of Texas at Arlington that examines each sentence and scores it according to the likelihood it contains an assertion of fact that can be checked. Duke then takes those checkable sentences and searches them against a database of fact checks done by humans to see if they match a previously checked claim. They send those results to human editors who, if they think the ruling looks reasonable, quickly post it to debate viewers’ screens.

It takes the system only about half a second to a full second from the time the audio comes in to passing its ruling to the editor for review, says Christopher Guess, lead technologist at the Reporter’s Lab, making real-time fact checking a viable option. Eliminating the slowest parts of the process, though—the initial human fact check and the editorial review—will not hap-

“A lot of the efficiency of fake news has to do with its emotional load, and a lot of the willingness to believe a crazy theory is more related to emotion than to thinking or reason.”

pen anytime soon, he says. The tendency of politicians to be deliberately vague, or to couch claims in terms that are favorable for them, makes it too difficult; human fact checkers often have to follow up with a politician to clarify just what claim he was trying to make. Automated fact checking of new assertions “with purely novel fact checking, isn’t even on our radar,” Guess says, “because how do you even have a computer determine what that person was saying?”

Even claim matching can be a challenge for computers. “Just seeing if somebody else said the same thing seems like a simple task,” Guess says. “But the fact is that in the vagaries of the English language, there’s a lot of different ways to say the same thing. And a lot of modern natural language processing is not good at determining the differences.”

At INRIA, France’s National Institute for Research in Digital Science and Technology, researcher Ioana Manolescu looks at fact checking as a data management problem for journalists. She is leading a team, with other research groups and journalists from the daily newspaper *Le Monde*, to develop ContentCheck, which uses NLP, automated reasoning, and data mining to provide fact checks of news articles. The system checks articles against data repositories such as France’s National Institute of Statistics and Economic Studies, and also helps journalists develop stories based on such data.

Manolescu’s goal is not so much to root out fake news as to help journalists find and make sense of data so they can use their storytelling skills to enlighten readers with valid information. “I am not as thrilled about fact checking as I used to be, because fact checking would assume that people yield to reason,” she says, and in many cases they do not. “A lot of the efficiency of fake news has to do with its emotional load, and a lot of the willingness to believe a crazy theory is more related to emotion than to thinking or reason.”

The best way for computer scientists to combat misinformation, she argues, is to find ways to provide more valid information. “Journalists do not have enough tools to process data at the speed and the efficiency that would serve society well,” she says. “So right now, that’s what I believe would be most useful.”

Further Reading

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, R., and Choi, Y. **Defending Against Neural Fake News, Proceedings of the Neural Information Processing Systems Conference, 32, (2019).** <https://papers.nips.cc/paper/9106-defending-against-neural-fake-news>

Duc Cao, T., Manolescu, I., and Tannier, X. **Extracting statistical mentions from textual claims to provide trusted content, 24th International Conference on Applications of Natural Language to Information Systems, (2019).** https://link.springer.com/chapter/10.1007/978-3-030-23281-8_36

Schuster, T., Schuster, R., Shah, D.J., and Barzilay, R. **The Limitations of Stylometry for Detecting Machine-Generated Fake News, Compute. Linguistics 46(2): 499-510 (2020).** <https://bit.ly/2FRxgIT>

Hassan, N., Arslan, F., Li, C., and Tremayne, M. **Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster, KDD '17: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2017).** <https://dl.acm.org/doi/10.1145/3097983.3098131>

Truth of Varying Shades, Analyzing Language in Fake News and Political Fact-Checking <https://vimeo.com/238236521>

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

© 2021 ACM 0001-0782/21/3 \$15.00

Can the Biases in Facial Recognition Be Fixed; Also, Should They?

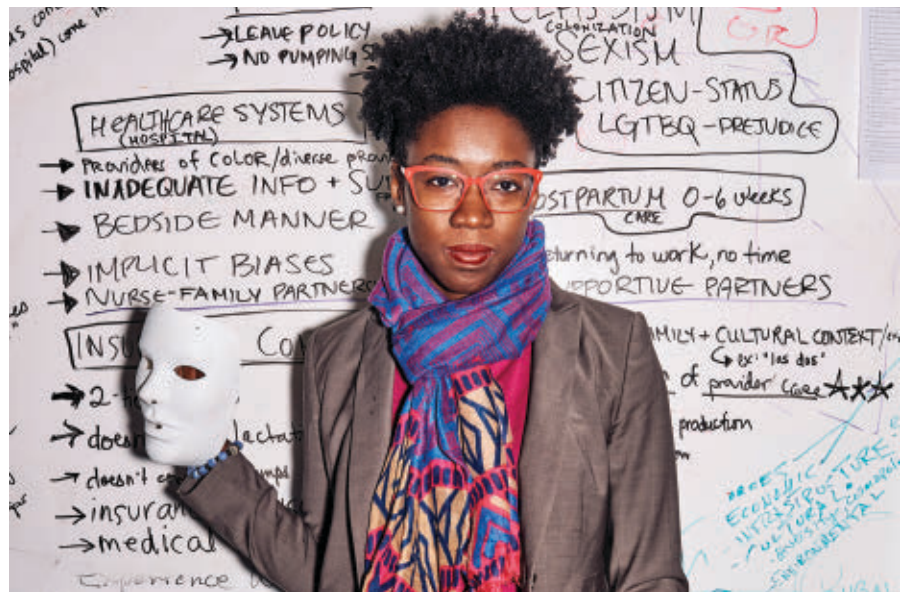
Many facial recognition systems used by law enforcement are shot through with biases. Can anything be done to make them fair and trustworthy?

IN JANUARY 2020, Robert Williams of Farmington Hills, MI, was arrested at his home by the Detroit Police Department. He was photographed, fingerprinted, had his DNA taken, and was then locked up for 30 hours. His crime? He had not committed one; a facial recognition system operated by the Michigan State Police had wrongly identified him as the thief in a 2018 store robbery. However, Williams looked nothing like the perpetrator captured in the surveillance video, and the case was dropped.

A one-off case? Far from it. Rewind to May 2019, when Detroit resident Michael Oliver was arrested after being identified by the very same police facial recognition unit as the person who stole a smartphone from a vehicle. Again, however, Oliver did not even resemble the person pictured in a smartphone video of the theft. His case, too, was dropped, and Oliver has filed a lawsuit seeking reputational and economic damages from the police.

What Williams and Oliver have in common is that they are both Black, and biases in deep-learning-based facial recognition systems are known to make such technology highly likely to incorrectly identify people of color. “This is not me. You think all Black people look alike?” an incredulous Williams asked detectives who showed him the CCTV picture of the alleged thief, according to *The New York Times*. In the *Detroit Free Press*, Oliver recalled detectives showing him the video of the perpetrator and realizing immediately, “It wasn’t me.”

It is such cases, borne out of the foisting of the privacy-invading mass-surveillance technology on whole populations, that continue to raise major questions over what role facial recognition should



Joy Buolamwini of the Massachusetts Institute of Technology Media Lab is one of many researchers that have found facial recognition technology to be deeply biased with regard to race, gender, age, and other factors.

have in a civilized society. Dubbed the “plutonium of artificial intelligence” in an appraisal in the ACM journal *XRDS*, Luke Stark of Microsoft Research’s Montreal lab described facial recognition as “inherently socially toxic.” Regardless of the intentions of its makers, he says, “it needs controls so strict that it should be banned for almost all practical purposes.”

Such controls are now the subject of ongoing legislative efforts in the U.S., the E.U., and the U.K., where lawmakers are attempting to work out how a technology that Washington, D.C.-based Georgetown University Law Center has characterized as placing populations in a “perpetual police lineup” should be regulated. At the same time, activist groups such as Amnesty International are monitoring the rollout of facial recognition at a human rights level, nam-

ing and shaming Western firms that provide the technologies to China’s surveillance state.

With politicians and pressure groups focused on facial recognition’s regulation, deployment, and human rights issues, where does that leave the technologists who actually make the stuff? Can software design and engineering teams charged with developing such systems address at least some of facial recognition technology’s deep-seated problems?

There’s certainly room for them to try. Kush Varshney, a senior researcher in trustworthy artificial intelligence at IBM’s T.J. Watson Research Center in Yorktown Heights, NY, says a raft of researchers have found facial recognition technology to be deeply biased with regard to race, gender, age, and disability, problems engineers can attempt to ad-

dress. Perhaps the best known of these researchers are Joy Buolamwini of the Massachusetts Institute of Technology Media Lab, and Timnit Gebru of Microsoft Research who, at a Conference on Fairness, Accountability, and Transparency at New York University in 2018, revealed just how badly commercial facial recognition systems fare when attempting to distinguishing gender across races.

The pair had tested three face-based gender classifiers (from IBM, China's Megvii, and Microsoft) and found the datasets the face recognition systems were trained on to be overwhelmingly (between 79% and 86%) comprised of faces of lighter-skinned people. As a result, they found the systems were skewed to better detect light-skinned people from the outset: the systems misclassified darker-skinned females as men 34% of the time, while lighter-skinned males were only misclassified as female 0.8% of the time.

"All classifiers performed best for lighter individuals and males overall. The classifiers performed worst for darker females," the researchers wrote in their paper *Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*.

The critiques did not end there: in late 2019, Patrick Grother and colleagues at the U.S. National Institute for Standards and Technology (NIST) published an exhaustive analysis of 189 face recognition algorithms from 99 developers. Although accuracy varied across algorithms, Grother's team found that in general, Asian and African faces garnered false positive matches 10 to 100 times more often than the faces of white people. Like Buolamwini and Gebru, they found African-American women experienced the highest rates of false positives. "Differentials in false positives in one-to-many matching are particularly important because the consequences could include false accusations," the NIST team said in its report.

The NIST team also found that where an algorithm is written can affect its performance. U.S.-developed software, they note, had the highest rates of false positives on faces of Asians, African-Americans, Native Americans, American Indians, Alaskan Indians, and Pacific Islanders. Algorithms developed in Asia, they found, did not have dramatic

Although accuracy varied across algorithms, Grother's team found Asian and African faces garnered false positives 10 to 100 times more often than white faces.

differences between matching accuracy of Asian and white (Caucasian) faces.

In a July 2020 report to Congress on facial recognition, the U.S. Government Accountability Office said this non-deterministic hodgepodge of unpredictable capabilities adds up to a technology that might, or might not, be accurate. As such, it generates performance differences where "higher error rates for certain demographic groups could result in disparate treatment, profiling, or other adverse consequences for members of these populations."

Worse, the GAO reports there is "no consensus" at all among academics, industry, standards bodies, or independent experts on how to fix the biases behind these "performance differences," which could have life-changing consequences for the mismatched. Facial recognition performance, the GAO says, depends on multiple algorithmic factors, such as the breadth of ethnicities used in the training data and variables like false-positive threshold settings, as well as photograph-related factors such as pose angle, illumination, skin tone, skin reflectance, expression, cosmetics, spectacle use, and image quality.

It was this proven propensity for dangerous biases (and, therefore, the potential for racist policing) that led IBM, Microsoft, and Amazon to halt entirely, or pause pending hoped-for legislation, their sales of facial recognition technology to police departments. That move was provoked by the police killing of George Floyd, a Black father of five, in Minneapolis, MN, in late May

2020, the event that sparked the global resurgence of the Black Lives Matter movement.

In an early June letter to Congress explaining its pullout from facial recognition sales and R&D, IBM CEO Arvind Krishna said his company "firmly opposes and will not condone uses of any technology, including facial recognition technology," for "mass surveillance, racial profiling, or violations of basic human rights and freedoms."

IBM's move was quickly followed by similar actions from Microsoft and Amazon, which in June 2020 each began one-year moratoria on sales of the technology to law enforcement agencies. The hope of all three firms is that legislation will be forthcoming to ensure facial recognition can only be used in ethical, unbiased ways that respect human rights and avoid racial or gender profiling.

Yet despite these moves, the global market for this biased technology is growing, as the GAO reported facial recognition system revenues were anticipated to grow from \$3 billion in 2016 to \$10 billion in 2024. In addition, innovation is rocketing: 631 U.S. patents were granted for facial recognition technologies in 2015, a number that grew to 1,497 in 2019, suggesting there is a lot more related (but potentially biased) technology to come.

Although IBM has departed from the facial recognition market, the runaway development of the technology concerns Varshney. After Buolamwini and Gebru showed the API for IBM's gender classifier to be so error-prone, Varshney said there are just too many points where biases can creep into the development process. "One is specifying the problem, which includes describing what the task is and describing the [facial recognition] metrics by which you'll be judging the task.

"And then there are the data understanding, data gathering, and data preparation stages. Following that, there is the modeling stage, which is when you're actually training a neural network, or some other type of model. Then there's the testing and evaluation phases, and then finally, there's the deployment phase. And there are issues that crop up in every single part of that complex cycle," Varshney says.

To fix such issues, he says, facial

recognition system developers need to “acquire as diverse a set of images as possible in order to not undersample certain groups.” The best way to do that, Varshney says, is to have as diverse a development team as possible, in terms of members’ races, genders, ages, and disabilities, so everyone can bring what he calls “their lived experience” to the task of specifying the facial recognition problem.

“The broader the set of stakeholders, the broader their set of perspectives and variety of experiences, and the more problems you can identify,” Varshney says.

Taking disability and health as an example, Varshney says a facial recognition system ought to be able to cope with people who have skin conditions, such as vitiligo, which can cause discolored patches on people’s faces. “That is something that you wouldn’t normally think about if you don’t bring in people with different perspectives. And people who have been victims of domestic abuse might have bruises that would create havoc with classification algorithms, too,” Varshney said.

NIST speculates its finding that algorithms developed in Asia are more accurate than those written in the U.S. may be due to some Asian development teams being more diverse. If so, says Grother, “The results are an encouraging sign that more diverse training data may produce more equitable outcomes.”

One facial recognition firm that continues to supply U.S. law enforcement, and which claims to use a very diverse development team, also happens to be the current *enfant terrible* of the field, Clearview AI of New York City. The firm hit the headlines because it scraped 2.8-billion face photos from publicly accessible Internet sites like Instagram, Facebook, Youtube, Twitter, and LinkedIn, all without user permission. Basically, the firm has created a search engine for any face image hosted on the public Internet.

That vast database already has landed Clearview in trouble with Google, Twitter, and LinkedIn, whose lawyers have issued cease-and-desist orders related to the scraping of their sites. That scraping also is likely to land Clearview AI in hot water in Europe, where GDPR data protection legislation requires in-

dividuals to opt in to permit the collection of their personal biometric data. The firm already has ceased operations in Canada, for similar reasons.

Clearview AI CEO Hoan Ton-That makes an extraordinary claim for the technology that company claims is in use by 600 U.S. law enforcement agencies to date: it is bias-free.

“When creating Clearview AI’s algorithm, we made sure to have trained our neural network with training data that reflects each ethnicity in a balanced way. So, unlike other facial recognition algorithms, which have misidentified people of color, an independent study indicates Clearview AI has no racial bias. As a person of mixed race, this is especially important to me,” Ton-That says.

The study he refers to is one Clearview AI commissioned itself—and it mimicked to a degree the methodology the American Civil Liberties Union (ACLU) used to test Amazon’s Rekognition system in 2018. ACLU had searched a database of 25,000 images of people who had been arrested using images of 535 members of Congress: Rekognition wrongly matched 28 Congresspersons to arrestees, with that total heavily skewed to politicians of color.

In its test, Clearview AI searched its database of 2.8 billion scraped faces using mugshots of 834 U.S. congressional and state legislators. “No incorrect matches were found...Accuracy was consistent across all racial and demographic groups,” the firm says in a six-page report signed off on by three independent observers: a former New York state judge, an expert in computational linguistics, and a management consultant.

Peter Fussey, director of the Centre for Research into Information, Surveillance, and Privacy (CRISP) at Essex University in the U.K., questions the accuracy of Clearview AI’s self-evaluation. Its brief report, he says, bears no comparison in length and detail to the “comprehensive” NIST facial recognition system studies, adding that the facial recognition expertise of the report’s three adjudicators is also unclear.

Fussey also questions the “ecological validity” of the methodology. “This is the idea that something tested in a lab can be replicated in wider society. For example, testing efficacy on U.S. Congress members that have a great deal of

searchable and publicly available photographs in circulation. This does not seem to approximate to the information we have about how the police are using Clearview AI on the public.”

Varshney thinks it’s time people stood back, as IBM has, and realized it is simply not a technology worth keeping. “Face recognition is a particularly thorny technology because it doesn’t have many beneficial uses. There’s just nothing good that can come out of it. It can be used in so many bad ways that even improving the technology could be worse for society,” he says. **□**

Further Reading

Hill, K.

Wrongfully Accused By An Algorithm, *The New York Times*, June 24, 2020, <https://nyti.ms/356Zt8D>

Anderson, E.

Facial Recognition Got Him Arrested for a Crime He Didn’t Commit, *Detroit Free Press*, July 11, 2020, <https://bit.ly/3bnpJwN>

Buolamwini, J. and Gebru, T.

Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification, *Proceedings of Machine Learning Research*, 81:1-15, 2018, Conference on Fairness, Accountability, and Transparency, <https://bit.ly/354ucDU>

Grother, P., Ngan, M., and Hanaoka, K.

Face Recognition Vendor Test Part 3: Demographic Effects U.S. National Institute of Standards and Technology, December 2019, <https://bit.ly/32Uv1vF>

Report to Congressional Requestors, Facial Recognition Technology: Privacy and Accuracy Issues Related to Commercial Uses, U.S. Government Accountability Office, July 2020, <https://bit.ly/2DrR5oV>

Krishna, A.

IBM CEO’s letter to the U.S. Congress on its abandonment of face recognition technology, June 8, 2020, <https://ibm.co/3hXDIM3>

Amazon: A one-year moratorium on police use of ‘Rekognition’ Amazon’s COVID-19 blog, June 10, 2020, <https://bit.ly/3gTOPUZ>

Smith, B.

Microsoft: Facial recognition: It’s Time for Action The Official Microsoft Blog, December 6, 2018, <https://bit.ly/3gVScuA>

Paul Marks is a technology journalist, writer, and editor based in London, U.K.

© 2021 ACM 0001-0782/21/3 \$15.00

Edmund M. Clarke (1945–2020)

EDMUND MELSON CLARKE, JR., a celebrated American academic who developed methods for mathematically proving the correctness of computer systems, died on December 22, 2020 at the age of 75 from complications of COVID-19. Clarke was awarded the A.M Turing Award in 2008 with his former student E. Allen Emerson and the French computer scientist Joseph Sifakis, for their work on model checking.

“I’ve never liked to fly, although I’ve done my share of it. I wanted to do something that would make systems like airplanes safer,” Clarke said in a 2014 video produced by the Franklin Institute when he was awarded their 2014 Bower Award and Prize for Achievement in Science^a “For his leading role in the conception and development of techniques for automatically verifying the correctness of a broad array of computer systems, including those found in transportation, communications, and medicine.”

Model checking is a practical approach for machine verification of mathematical models of hardware, software, communications protocols, and other complex computing systems. The technique is used to formally validate all of the states that a system can possibly reach, even when the number of states seems impossibly large—for example, more than the number of stars in the universe. These techniques, first developed when he was an assistant professor at Harvard University in 1981, are now widely used in the design of computing hardware and safety-critical systems and are increasingly being used for protocol validation and computer security.

Clarke grew up in rural Virginia and was the first person in his family to graduate from college. He earned a bachelor’s degree in mathematics in 1967 at the University of Virginia, a master’s degree in mathematics the following year at Duke University, and a Ph.D.

^a <https://www.fi.edu/laureates/edmund-m-clarke>



in computer science from Cornell in 1976. He then returned to Duke where he taught for two years before becoming an assistant professor at Harvard University in 1978. He joined CMU's faculty in 1982, was appointed full professor in 1989, University Professor in 2008, and became an Emeritus Professor in 2015.

As Clarke explained in the 2008 Turing Award Paper,¹ “Model checkers typically have three main components: (1) a *specification language*, based on propositional temporal logic. (2) a way of encoding a state machine representing the system to be verified, and (3) a *verification procedure*, that uses an *intelligent* exhaustive search of the state space to determine if the specification is true or not.” Most model checkers, upon finding that the specification is violated, provide the counterexample, which is invaluable in debugging complex systems.

His students remember Clarke for the way that he mentored them both professionally and personally. For instance, Somesh Jha, now a professor at the University of Wisconsin reminisced “Ed treated his students like family. Martha and Ed regularly invited students and their families to their house. I remember those parties quite fondly.”

“He used to insist that we should work on important problems of practical significance, but at the same time, he also had appreciation of basic research on foundational problems,” said A. Prasad Sistla, one of Clarke's students at Harvard who followed him to CMU and now a professor of Computer Science at the University of Illinois at Chicago.

“Ed was a perfect mentor for me,” recalls ACM Fellow David Dill, now an Emeritus Professor at Stanford, known for his work in formal verification and the security of electronic voting systems. “He only gave positive reinforcement. He would ask questions, refer to related work that would be useful to know, suggest directions and research topics, and carefully follow my presentations ... Once I started listening, I suddenly became productive.”

“On his 69th birthday, 100 of his students, postdocs, and visitors from all over the world gathered together in Pittsburgh to celebrate his planned retirement and to praise his enormous contribution to the Model Checking

“Ed was a perfect mentor for me. He only gave positive reinforcement. He would ask questions, refer to related work that would be useful to know, suggest directions and research topics, and carefully follow my presentations ... Once I started listening, I suddenly became productive.”

ACM FELLOW DAVID DILL
EMERITUS PROFESSOR,
STANFORD UNIVERSITY

area. He was surrounded with love and appreciation,” recalls ACM Fellow Orna Grumberg, a professor of computer science at Technion in Israel.

Some of his colleagues also noted Clarke's mentoring. Randal Bryant mentioned it first when asked for recollection. “He always had a group of graduate students and post-docs who collaborated with him and with each other very effectively. He launched them into very successful careers in both industry and academia.” Bryant also noted the scope of Clarke's life-long work: hardware to software to protocols—“He was intellectually broad and open-minded.”

Indeed, academia was the family business, with Clarke's wife Martha serving as the graduate admissions coordinator for the CMU Computer Science Department and the School of Computer Science, where she worked for 28 years until her retirement in 2014. The two were high-school sweethearts, marrying immediately after they graduated. They celebrated their 52nd anniversary in 2020.

Clarke's son, Jonathan, said that he gave his three sons a “joy of learning” and encouraged them to take courses in science and mathematics. Jonathan earned a Ph.D. in Finance and is now a professor and senior associate dean at Georgia Tech's Business School. His younger brother, Jeffrey, earned a medical degree and is an oncologist and assistant professor at Duke University School of Medicine. Their older brother, James, earned a Ph.D. in Chemistry and is the Director of Quantum Hardware at Intel in Portland, Oregon. In addition to his wife and sons, Clarke is also survived by six grandchildren.

His wife Martha reflected on Clarke's own joy of learning, “He was always reading. I remember him even taking scientific papers to read during high school football games.” His interests were wider than simply academics, with members of his family noting his fondness for fishing, photography, and flying kites.

A founder of the *Computer Aided Verification Conference* in 1989, Clarke was also the former editor-in-chief of the Springer journal *Formal Methods in Systems Design*. He was a Fellow of the ACM and the IEEE, and a member of both Sigma Xi and Phi Beta Kappa. He was inducted into the National Academy of Engineering in 2005, and the American Academy of Arts and Sciences in 2011.

Clarke was the co-recipient of the 1998 ACM Paris Kanellakis Theory and Practice Award, CMU's 1999 Allen Newell Award for Excellence in Research, the 2004 IEEE Harry H. Goode Memorial Award, and the Conference on Automated Deduction's 2008 Herbrand Award for Distinguished Contributions to Automated Reasoning. **C**

Reference

1. Clarke, E.M., Emerson, E.A., and Sifakis, J. Model checking: Algorithmic verification and debugging. *Commun. ACM* 52, 11 (Nov. 2009), 74–84; <https://doi.org/10.1145/1592761.1592781>

Simson Garfinkel is the U.S. Census Bureau's Senior Computer Scientist for Confidentiality and a part-time faculty member at George Washington University in Washington, D.C., USA. He is an ACM Fellow.

Eugene H. Spafford is a professor of computer science and the founder and executive director emeritus of the Center for Education and Research in Information Assurance and Security at Purdue University, W. Lafayette, IN, USA. He is an ACM Fellow.

Copyright held by authors/owners.

ACM Transactions on Evolutionary Learning and Optimization (TELO)

Open for Submissions

Publishes papers at the intersection of optimization and machine learning, making solid contributions to theory, method and applications in the field.



ACM Transactions on Evolutionary Learning and Optimization (TELO) publishes high-quality, original papers in all areas of evolutionary computation and related areas such as population-based methods, Bayesian optimization, or swarm intelligence.

We welcome papers that make solid contributions to theory, method and applications. Relevant domains include continuous, combinatorial or multi-objective optimization. Applications of interest include but are not limited to logistics, scheduling, healthcare, games, robotics, software engineering, feature selection, clustering as well as the open-ended evolution of complex systems.

We are particularly interested in papers at the intersection of optimization and machine learning, such as the use of evolutionary optimization for tuning and configuring machine learning algorithms, machine learning to support and configure evolutionary optimization, and hybrids of evolutionary algorithms with other optimization and machine learning techniques.

For more information and to submit your work, please visit:

telo.acm.org



Association for Computing Machinery



DOI:10.1145/3447251

Pamela Samuelson

Legally Speaking

The Push for Stricter Rules for Internet Platforms

Considering the origins, interpretations, and possible changes to Communications Decency Act § 230 amid an evolving online environment.

ONE OF THE few things about which U.S. Republican and Democratic politicians generally agree these days is that the law widely known as § 230 of the Communications Decency Act needs to be repealed, amended, or reinterpreted.

Section § 230(c)(1) provides Internet platforms with a shield from liability as to information content posted by others. It states that “[n]o provider or user of an interactive computer service shall be treated as the publisher or speaker of any information provided by another information content provider.”

Although computing professionals might question whether these 26 words truly “created the Internet,”^a Internet platform companies and most technology law specialists would say this characterization is only a slight exaggeration, at least as to sites that host user-posted content.

Although Donald Trump and Joe Biden have both recommended that

Congress repeal this provision, their reasons are starkly different. Trump and other Republican critics think Internet platforms take down *too much* content in reliance on this law. They claim platforms are biased against conservative viewpoints when they remove or demote such postings. Democratic critics of § 230 blame Internet platforms for not taking down *more* harmful content, such as disinformation about COVID-19 or elections. They think repealing or amending § 230 would make platforms more responsible participants in civil society.

Short of repeal, several initiatives aim to change § 230. Eleven bills have been introduced in the Senate and nine in the House of Representatives to amend § 230 in various ways. President Trump issued an Executive Order directing the National Telecommunications and Information Administration (NTIA) to petition the Federal Communications Commission (FCC) to engage in rule-making to interpret § 230 more narrowly than courts have done. Moreover, Justice Clarence Thomas of the U.S. Supreme Court recently criticized

court decisions giving a broad interpretation of § 230, signaling receptivity to overturning them.

This column explains the origins of § 230 and its broad interpretation. It then reviews proposed changes and speculates about what they would mean for Internet platforms.

Overview of § 230

In saying Internet platforms are neither “speakers” nor “publishers” of information posted by others, § 230(c)(1) protects platforms from lawsuits for unlawful content, such as defamation, posted by their users. Victims can sue the “speakers” who posted the unlawful content, but courts almost always dismiss victims’ lawsuits against platforms shortly after filing.

Why would victims sue platforms? For one thing, victims may not be able to identify wrongdoers because harmful postings are often anonymous. Second, victims typically want judges to order the platforms to take down harmful content. Third, platforms generally have more resources than wrongdoers. Victims who want compensation for

^a Jeff Kosseff, *The Twenty-Six Words That Created the Internet* (2019).



harms could likely get more money from platforms than from wrongdoers.

The 1995 *Stratton Oakmont v. Prodigy* case, which catalyzed the enactment of § 230, illustrates. A Prodigy user claimed Stratton-Oakmont engaged in securities fraud on a Prodigy bulletin board. Stratton-Oakmont responded by suing Prodigy and the anonymous user for defamation, initially asking for \$100 million in damages. Even though Prodigy did not know of or contribute to the defamatory content, the court refused to dismiss the case. It regarded Prodigy as a “publisher” of the defamation because of its stated policy of exercising editorial control over content on its site. (The case settled after Prodigy apologized.)

In addition, § 230(c)(2) says platforms are not liable for any action they take “in good faith to restrict access to or availability of material that the provider ... considers to be obscene, lewd, lascivious, filthy, excessively violent, harassing, or otherwise objectionable, whether or not such material is constitutionally protected.”

Section 230 was enacted as part of a 1996 overhaul of U.S. telecommunications law. Its goal was to encourage emerging Internet services to monitor their sites for harmful content without the risk of being treated as publishers

of user-posted content. Congress thought this law would foster the growth of the Internet sector of the economy, as indeed it has.

Zeran’s Broad Interpretation of § 230

Zeran v. America Online was the first court decision to interpret § 230. The dispute arose over someone posting Ken Zeran’s telephone number on AOL in advertisements for T-shirts glorifying the 1995 Oklahoma City terrorist bombing. It directed interested AOL users to call Ken about the shirts. Zeran got hundreds of telephone calls, including death threats, even though he knew nothing about the ads and was not even an AOL user.

Zeran asked AOL staff on several occasions to remove the ads. After AOL failed to follow through on assurances the ads would be deleted, Zeran sued AOL for negligently failing to protect him from harms resulting from this post.

AOL asked the court to dismiss Zeran’s lawsuit based on § 230. Although Zeran did not claim AOL was a publisher who had defamed him by not taking down the ads, the court construed his negligence claim as an attempted evasion of Congress’ intent to protect online services from law-

suits for content posted by third parties. Hence, the court granted AOL’s motion to dismiss.

Relying on *Zeran*, online platforms have routinely avoided legal liability through § 230 defenses. Numerous cases have featured very sympathetic plaintiffs, such as victims of revenge porn, fraudulent ads, and professional defamation, and some unsympathetic defendants who seem to have encouraged or tolerated harmful postings.

Proposals to Amend § 230

In late 2020, the Senate introduced a bill that would repeal § 230 outright. However, numerous bills would give § 230 a significant modification (bill names and numbers, sponsors, and links can be found at <https://bit.ly/3iHUtW8>).

Members of Congress have taken several different approaches to amending § 230. Some would widen the categories of harmful conduct for which § 230 immunity is unavailable. At present, § 230 does not apply to user-posted content that violates federal criminal law, infringes intellectual property rights, or facilitates sex trafficking. One proposal would add to this list violations of federal civil laws.

Some bills would condition § 230 immunity on compliance with certain

conditions or make it unavailable if the platforms engage in behavioral advertising. Others would require platforms to spell out their content moderation policies with particularity in their terms of service (TOS) and would limit § 230 immunity to TOS violations. Still others would allow users whose content was taken down in “bad faith” to bring a lawsuit to challenge this and be awarded \$5,000 if the challenge was successful.

Some bills would impose due process requirements on platforms concerning removal of user-posted content. Other bills seek to regulate platform algorithms in the hope of stopping the spread of extremist content or in the hope of eliminating biases.

Possible Ambiguities in § 230?

NTIA’s petition asserts there is an ambiguity in § 230 about the relationship between § 230(c)(1) and § 230(c)(2) that the FCC should resolve through rulemaking. It posits that the function of § 230(c)(1) should be to shield platforms from liability for allowing user-posted content to remain on their sites. Take-downs of user-posted content should be governed, however, under its sister provision, § 230(c)(2).

The NTIA petition asserts that takedowns of “otherwise objectionable” content would not be sheltered by § 230(c)(2) unless the content was similar in nature to the named categories (for example, lewd or harassing). NTIA does not accept that platforms can construe that term broadly. Take-downs of “disinformation,” for instance, would under this interpretation be ineligible for the § 230(c)(2) immunity shield. The FCC is unlikely to proceed with the proposed rulemaking under the Biden administration.

Equally ambiguous, in NTIA’s view, is the meaning of “good faith” in §230(c)(2). The NTIA petition asserts this standard cannot be satisfied if the take-down is “deceptive, pretextual, or inconsistent with [the platform’s] terms of service.” Moreover, it regards “good faith” as requiring due process protections. In NTIA’s view, user-posted content cannot be taken down unless the platform notified users, explained their basis for take-down decisions, and provided users with a meaningful opportunity to be heard about it.

Section 230 was enacted as part of a 1996 overhaul of U.S. telecommunications law. Its goal was to encourage emerging Internet services to monitor their sites for harmful content without the risk of being treated as publishers of user-posted content.

Narrowing § 230 By Interpretation?

Neither legislation nor an FCC rulemaking may be necessary to significantly curtail § 230 as a shield from liability. Conservative Justice Thomas has recently suggested a reinterpretation of § 230 that would support imposing liability on Internet platforms as “distributors” of harmful content.

A key precedent on distributor liability dates back to a 1950s era decision, *Smith v. California*. Smith owned a bookstore that sold books, candy, and other sundries. A Los Angeles ordinance forbade sale of obscene or indecent books at such stores. Smith was convicted of selling obscene books, even though he had not read the books at issue and didn’t know of their contents. The Supreme Court reversed Smith’s conviction holding that LA’s strict liability ordinance violated the First Amendment of the U.S. Constitution. Distributors of obscene books must know or have reason to know of illegal contents to be subject to prosecution.

Applying *Smith* to platforms under § 230 could result in Internet platforms being considered “distributors” of un-


lawful content once on notice of such content. Section 230, after all, shields these services from liability as “speakers” and “publishers,” but is silent about possible “distributor” liability.

Endorsing this interpretation would be akin to adopting the notice-and-takedown rules that apply when platforms host user-uploaded files that infringe copyrights. Notice-and-takedown regimes have long been problematic because false or mistaken notices are common and platforms often quickly take-down challenged content, even if it is lawful, to avoid liability.

Conclusion

Civil liberties groups, Internet platforms, and industry associations still support § 230, as do Senator Wyden and former Congressman Chris Cox, who co-sponsored the bill that became § 230. Wyden and Cox have pointed out that an overwhelming majority of the 200 million U.S.-based Internet platforms depend on § 230 to protect them against unwarranted lawsuits by disgruntled users and those who may have been harmed by user-posted content of which the platforms were unaware and over which they had no control.

For the most part, these platforms promote free speech interests of their users in a responsible way. Startup and small nonprofit platforms would be adversely affected by some of the proposed changes insofar as the changes would enable more lawsuits against platforms for third-party content. Fighting lawsuits is costly, even if one wins on the merits.

Much of the fuel for the proposed changes to § 230 has come from conservative politicians who are no longer in control of the Senate. The next Congress will have a lot of work to do. Section 230 reform is unlikely to be a high priority in the near term. Yet, some adjustments to that law seem quite likely over time because platforms are widely viewed as having too much power over users’ speech and are not transparent or consistent about their policies and practices. 

Pamela Samuelson (pam@law.berkeley.edu) is the Richard M. Sherman Distinguished Professor of Law and Information at the University of California, Berkeley, CA, USA.

Copyright held by author.

Privacy

Informing California Privacy Regulations with Evidence from Research

Designing and testing 'Do Not Sell My Personal Information' icons.

EXERCISING PRIVACY CHOICES is akin to a scavenger hunt: information about available choices is hard to find and mechanisms can be difficult to use. My research group has been examining ways to improve privacy user experiences (UX).¹² We started exploring website privacy “nutrition labels”^{9,10} a decade before Apple introduced them in their app store in December 2020, and recently we proposed a privacy and security label for IoT devices.⁵ When the State of California passed the California Consumer Privacy Act (CCPA) mandating a “Do Not Sell My Personal Information” website opt-out link and optional icon, we developed and evaluated icon designs and submitted recommendations in response to the Office of the Attorney General (OAG) call for public comments.^a After several twists and turns, in December 2020 the OAG issued proposed regulations with our recommended icon.

Icon Development and Evaluation

In fall 2019, our team of researchers^b

a All documents pertaining to the CCPA rule-making activities can be found at <https://oag.ca.gov/privacy/ccpa/current>

b Members of our team included Alessandro Acquisti, Michelle Chou, Lorrie Cranor, Hana Habib, Norman Sadeh, and Yaxing Yao from Carnegie Mellon University; Florian Schaub and Yixin Zou from the University of Michigan School of Information; and Joel Reidenberg from Fordham University School of Law.



began brainstorming possible icon designs. We developed 11 icons that could represent one of three concepts: *choice*, *opting out*, and *do not sell personal information*. We focused on representing these concepts rather than on representing privacy itself, as privacy is difficult to visualize and popular privacy visualizations (locks, shields, keys, masks, eyes) are already used in Web security and privacy tools.

We conducted an initial evaluation of our 11 icons as well as the green “privacy rights” icon promoted by the Digital

Advertising Alliance industry group for use as a CCPA icon. We recruited participants from Amazon’s Mechanical Turk (MTurk) and showed one randomly selected icon to each participant. Half the participants saw the icon with the text “Do Not Sell My Personal Information” and half saw the icon alone. We asked participants to tell us what they thought the icon communicated and what they thought would happen if they clicked on it. Then we showed them all 12 icons, shown in Figure 1a, and asked them to select the icons that

Figure 1a. Icons evaluated in first user study.

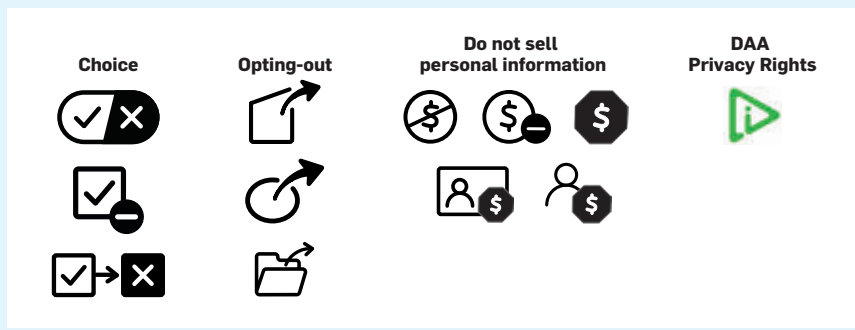


Figure 1b. Refined icons evaluated in second user study. The three on the left were also evaluated in the combined study with link texts.



Figure 1c. Link texts evaluated in user study. The five shown in bold were also evaluated in the combined study with icons.

- | | |
|--|------------------------------|
| Do Not Sell My Personal Information | Privacy Choices |
| Do Not Sell My Info | Privacy Options |
| Don't Sell My Info | Privacy Opt-Outs |
| Do Not Sell | Personal Info Choices |
| Don't Sell | Personal Info Options |
| Do-Not-Sell Choices | Personal Info Opt-Outs |
| Do-Not-Sell Options | Do Not Sell My Info Choices |
| Do-Not-Sell Opt-Outs | Do Not Sell My Info Options |

Figure 1d. Toggle icon variants evaluated in user study. All icons were tested in both red and blue.



Figure 1e. OAG icons evaluated in final user study.



best conveyed the presence of privacy choices and do-not-sell choices.

Based on the results of the initial evaluation, we refined five of the icons (see Figure 1b) and conducted another MTurk evaluation. The slash-dollar icon was misinterpreted as relating to money when it appeared without link text, but it was most preferred by participants as an icon for representing do-not-sell. The DAA icon was often misinterpreted as a play button or an information button. The stylized-toggle led to the fewest misconceptions.

Our initial evaluations demonstrated the importance of placing link text next to the icons, and our prior research showed the specific wording of this text can have a large impact.⁶ We brainstormed possible link texts and evaluated 16 of them (see Figure 1c), including “Do Not Sell My Personal Information” and “Do Not Sell My Info,” which were in the CCPA legislation. Our evaluation identified three new promising link texts: “Privacy Choices,” “Privacy Options,” and “Personal Info Choices.”

Our next step was to evaluate three icons and five link texts together in the context of a fictitious shoe retailer website. We tested 23 icon-link text combinations, including link texts without an icon and the icons without a link text. We recruited 1,468 MTurk participants and randomly assigned them to view the shoe website with one icon-link text combination shown in the footer (see Figure 2).

We found the link texts had more of an impact on participant expectations than the icons, and the icons continued to convey misconceptions. The combination of stylized-toggle and the “Privacy Options” link text best conveyed choices about personal information. The CCPA link texts best conveyed do-not-sell choices. In February 2020, we sent a detailed report to the OAG and recommended adoption of the stylized-toggle icon with either the “Privacy Options” link text or the CCPA link texts.²

More Research Needed

Shortly after receiving our report, the OAG released the first set of modifications to the CCPA regulations with an icon that was similar to our stylized-toggle but differed in significant ways.

While our icon was blue and contained both a checkmark and an X arranged to convey choices without suggesting a toggle in a particular state, the OAG's icon was red, contained only an X, and strongly resembled an actual toggle button. Comments on Twitter raised concerns that the OAG's icon might be misinterpreted as representing the state of a user's opt-out selection.

We quickly ran another MTurk study to compare our stylized toggle icon with the OAG's toggle icon and a variant of the OAG's toggle icon with a larger X—each tested in both red and blue (see Figure 1d). We found our stylized toggle better conveyed do-not-sell choices than the OAG's icon and led to fewer misconceptions. The larger X and the color had minimal impact. After we submitted a report on these results to the OAG,³ they released their second set of modifications to the CCPA regulations, removing their recommendation for an icon altogether.

Later the OAG asked us if we would evaluate a set of four new icons (see Figure 1e) with 1,000 California residents. Besides evaluating each icon's ability to communicate the presence of do-not-sell choices, they asked us to test the ability of each icon to stand out on websites and motivate users to click. This necessitated some changes to our study protocol.

To ensure participants viewed the area of the fictitious shoe store website where the CCPA link appeared, we showed the website with the CCPA link text and one of the four icons or no icon and asked participants to find a link where they could get information about shipping shoes overnight. We then hid the shoe store website image and instructed participants to imagine they were concerned about an online store selling their personal information. We then asked, "Do you remember seeing any feature in the screenshot that you could use to prevent this from happening?" Next, we showed the screenshot again, calling attention to the icon and link text. We instructed participants to imagine this was the first time they had noticed the icon and link text on a website, and we asked how likely they would be to click on them. We followed up with questions about what would happen if they clicked and then showed them all four icons and asked them

This story provides a case study of how academic researchers can refocus their research to answer policymakers' questions.

how well each conveyed the presence of do-not-sell choices.

Our results showed the icons successfully increased users' attention to the link text but did not create a significantly higher motivation to click. Interestingly, we found participants who were not shown any icon were most likely to have correct expectations about what would happen if they clicked; all four icons introduced misconceptions. Furthermore, participants did not rate any of the icons well. We submitted our report to the OAG in May 2020 and recommended

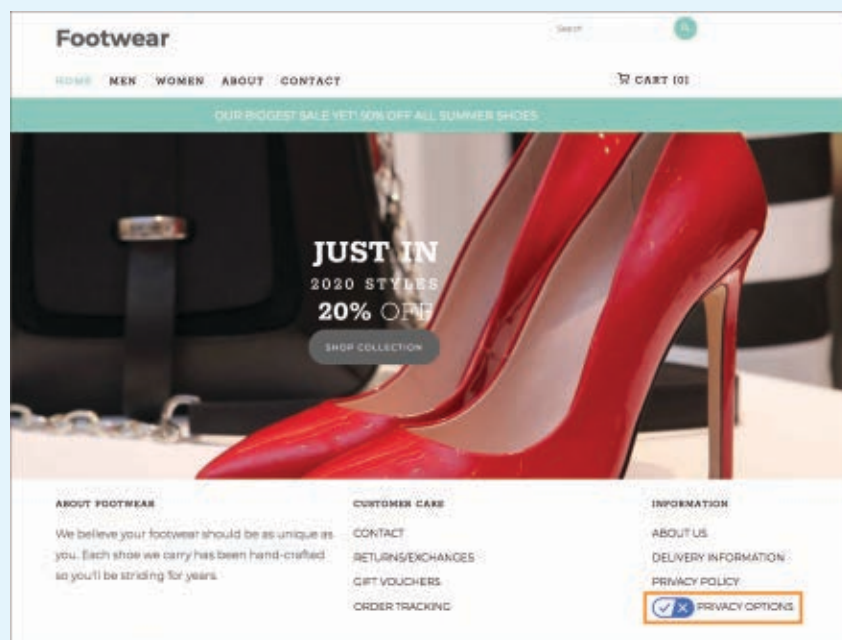
they evaluate other icons and conduct public education to increase awareness of do-not-sell choices.¹

Informing Policymaking with Research

Over six months passed before the OAG released their fourth set of modified regulations in December 2020, this time recommending the optional use of our blue stylized-toggle icon.

This story provides a case study of how academic researchers can refocus their research to answer policymakers' questions. When our team realized the OAG had a need for a specific privacy icon, we quickly pivoted from studying website privacy choices generally, to designing and evaluating a privacy icon to meet this need. After a three-month sprint to meet the public comment deadline we turned our attention to writing a research paper on this project. However, the OAG's recommendation of an untested icon triggered more quick action from our team, and we conducted another study to demonstrate that small changes in the icon could make a big difference in how it would be interpreted. Just when we thought we were done, the OAG reached out to us again and we put other work on hold so that we could redesign our experimental protocol and perform another evaluation.

Figure 2. In the combination icon and link text study participants were shown this screen shot of a fictitious footwear website with an icon and link text highlighted.





Advertise with ACM!

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



Request a media kit with specifications and pricing:

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



In a world where privacy is increasingly threatened by online trackers and ubiquitous sensors, how much can a little blue privacy icon accomplish, especially when its use is entirely optional?

The combined expertise of our research team, the availability of flexible research funding, and the ability to conduct studies quickly and inexpensively using crowd workers allowed us to provide timely research that informed public policymaking. While crowd working platforms that do not offer demographically representative samples have their limitations, they are useful inexpensive tools for carrying out studies like these where the focus is on comparing alternatives.¹⁰

In the end, this project has resulted in a forthcoming CHI 2021 paper,⁸ a case study I will use in my usable privacy and security class, and an icon that may soon appear on websites. Moving forward, I am hopeful that websites will adopt the stylized-toggle icon not only for CCPA compliance, but also to point users toward a “Privacy Options” page with all of their privacy choices and settings in one place.

In a world where privacy is increasingly threatened by online trackers and ubiquitous sensors, how much can a little blue privacy icon accomplish, especially when its use is entirely optional? While an icon alone will not protect privacy, it can make it easy for users to find information about their privacy choices. We have seen in our research that Internet users are not always aware they have privacy choices, and they struggle to figure

out how to exercise them.⁷ A standardized icon is a good first step toward increasing the discoverability of privacy choices and raising awareness about them. Ultimately, the use of standardized protocols interfacing with usable “personal privacy assistants”⁴ will allow users to make flexible fine-grained privacy choices that adjust according to each user’s preferences and context across all websites, apps, and devices. ■

References

1. Cranor, L.F. et al. CCPA Opt-out icon testing—phase 2. May 28, 2020; <https://oag.ca.gov/sites/all/files/agweb/pdfs/privacy/dns-icon-study-report-052822020.pdf>
2. Cranor, L.F. et al. Design and Evaluation of a Usable Icon and Tagline to Signal an Opt-Out of the Sale of Personal Information as Required by CCPA. February 4, 2020; <http://cup.cs.cmu.edu/pubs/CCPA2020Feb04.pdf>
3. Cranor, L.F. et al. User Testing of the Proposed CCPA Do-Not-Sell Icon. February 24, 2020. <http://cup.cs.cmu.edu/pubs/CCPA2020Feb24.pdf>
4. Das, A. et al. Personalized privacy assistants for the Internet of Things: Providing users with notice and choice. *IEEE Pervasive Computing* 17, 3 (Jul.–Sep. 2018), 35–46; DOI:10.1109/MPRV.2018.03367733
5. Emami-Naeini, P. et al. Ask the experts: What should be on an IoT privacy and security label? In *Proceedings of the 2020 IEEE Symposium on Security and Privacy* (San Francisco, CA, USA, 2020), pp. 447–464. DOI:10.1109/SP40000.2020.00043
6. Giovanni Leon, P. What do online behavioral advertising privacy disclosures communicate to users? In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society (WPES '12)*. ACM, New York, NY, USA, 2012, 19–30; DOI:10.1145/2381966.2381970
7. Habib, H. It's a scavenger hunt: Usability of Websites' opt-out and data deletion choices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, New York, NY, USA, 2020, 1–12. DOI:10.1145/3313831.3376511
8. Habib, H. et al. Toggles, dollar signs, and triangles: How to (in)effectively convey privacy choices with icons and link texts. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '21)* (May 2021, Yokohama, Japan). ACM, New York, NY, USA, 2021, 19–30; DOI:10.1145/3411764.3445387
9. Kelley, P.G., Cranor, L.F., and Sadeh, N. Privacy as part of the app decision-making process. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2013, 3393–3402; DOI:10.1145/2470654.2466466
10. Kelley, P.G. et al. Standardizing privacy notices: an online study of the nutrition label approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2010, 1573–1582. DOI:10.1145/1753326.1753561
11. Redmiles, E.M., Kross, S., and Mazurek, M.L. How well do my results generalize? Comparing security and privacy survey results from mturk, web, and telephone samples. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy*, 1326–1343; DOI:10.1109/SP.2019.00014.
12. Schaub, F. and Cranor, L.F. Usable and useful privacy interfaces. In *An Introduction to Privacy for Technology Professionals*, Travis D. Breaux, Ed., IAPP (2020), 176–238; <https://iapp.org/media/pdf/certification/IAPP-Intro-to-Privacy-for-Tech-Prof-SAMPLE.pdf>

Lorrie Faith Cranor (lorrie@cmu.edu) is Director and Bosch Distinguished Professor in Security and Privacy Technologies, CyLab Security and Privacy Institute and FORE Systems Professor, Computer Science and Engineering & Public Policy, Carnegie Mellon University, Pittsburgh, PA, USA.

Copyright held by author.

► Susan J. Winter, Column Editor

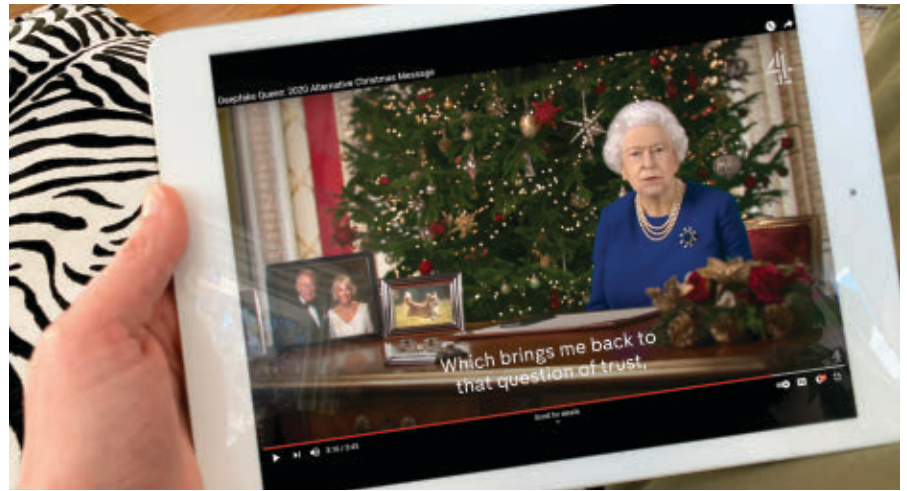
Computing Ethics

What To Do About Deepfakes

Seeking to reap the positive uses of synthetic media while minimizing or preventing negative societal impact.

SYNTHETIC MEDIA TECHNOLOGIES are rapidly advancing, making it easier to generate nonveridical media that look and sound increasingly realistic. So-called “deepfakes” (owing to their reliance on deep learning) often present a person saying or doing something they have not said or done. The proliferation of deepfakes^a creates a new challenge to the trustworthiness of visual experience, and has already created negative consequences such as nonconsensual pornography,¹¹ political disinformation,¹⁹ and financial fraud.³ Deepfakes can harm *viewers* by deceiving or intimidating, harm *subjects* by causing reputational damage, and harm *society* by undermining societal values such as trust in institutions.⁷ What can be done to mitigate these harms?

It will take the efforts of many different stakeholders including platforms, journalists, and policymakers to counteract the negative effects of deepfakes. Technical experts can and should play an active role. Technical experts must marshal their expertise—their understanding of how deepfake technologies work, their insights into how the technology can be further developed and used—and direct their efforts to find solutions that allow the beneficial uses of synthetic media technologies and mitigate the negative effects. While



A deepfake video from a December 25, 2020, posting “Deepfake Queen: 2020 Alternative Christmas Message” (source <https://youtu.be/ivY-Abd2FfM>).

successful interventions will likely be interdisciplinary and sociotechnical, technical experts should play a role by designing, developing, and evaluating potential technical responses and in collaborating with legal, policy, and other stakeholders in implementing social responses.

The Responsibilities of Technical Experts

Deepfakes pose an age-old challenge for technical experts. Often as new technologies are being developed, their dangers and benefits are uncertain and the dangers loom large. This raises the question of whether technical experts should even work on or with a technology that has the potential for great harm. One of the

most well known and weighty versions of this dilemma was faced by scientists involved in the development and use of the atomic bomb.¹⁸ The dilemma also arose for computer scientists as plans for the Strategic Defense Initiative were taking shape¹⁴ as well as when encryption techniques were first debated.¹³

Although some technical experts may decide not to work on or with the synthetic media technologies underlying deep fakes, many will likely attempt to navigate more complicated territory, trying to avoid doing harm and reap the benefits of the technology. Those who take this route must recognize they may actually enable negative social consequences and take steps to reduce this risk.

^a See <https://bit.ly/3qY0Lua>

Responsibility can be diffuse and ambiguous. Any deepfake involves multiple actors who create the deepfake, develop the tool used to make it, provide the social media platform for amplification, redistribute it, and so on. Since multiple actors contributed, accountability is unclear, setting the stage for a dangerous blame game where no one is held responsible. Legal interventions will also be stymied by difficulties in determining jurisdiction for punishing deepfake creators,⁵ and by the need to strike a balance with free speech concerns for platform publication.¹⁸ Still, ethically, each actor is responsible for what they do as well as what they fail to do, particularly if a negative consequence might have been averted. Technical experts have an ethical responsibility to avoid or mitigate the potential negative consequences of their contributions.

Consider DeepNude, an app that converts images of clothed women into nude images. It is not only end users that are doing harm with the app. The developer is reported to have said that he did not expect the app to go viral, and later withdrew it from the marketplace.⁶ In defense of the developer, some could consider him thoughtless but not ill-intended. This, however, misses the fact that the tool was designed for a purpose that inherently objectifies women. The negative outcome of the app was not difficult to foresee, and the designer bears some responsibility for the harm caused.

Many technical experts will work on more generic synthetic media technologies that have diverse applications and uses even they cannot foresee. But despite the uncertainty of future uses they still are not entirely off the hook ethically. Responsibility in this case is less about blame than about making conscientious efforts to identify the potential uses of their creations in the hands of a variety of users with ill as well as good intent.⁴ NeurIPS, a premier conference in the field of AI, is trying to enforce this ethical responsibility by requiring submissions to include a “Broader Impact” section that addresses both potential positive and negative social impacts.^b Technical experts must go a step further though:

not to just think or write about social impacts, but to design tools and techniques that limit the possibility of harmful or dangerous use.

How to Be Part of the Solution

Individually and collectively, the behavior of technical experts in the field of synthetic media is coming under scrutiny. They should be expected to, and should expect one another to, behave in ways that diminish the negative effects of deepfakes. Research and development of synthetic media will be better served if technical experts see themselves as part of the solution, and not the problem. Here are three areas where technical experts can make positive contributions to the development of synthetic media technologies: education and media literacy, subject defense, and verification.

Education and Media Literacy.

Technical experts should speak out publicly (as some already have) about the capabilities of new synthetic media. Deepfakes have enormous potential to deceive viewers and undermine trust in what they see, but the possibility of such deception is diminished when viewers understand synthetic media and what is possible. For example, were individuals taught to spot characteristic flaws that might give deepfakes away, they would be empowered to use their own judgment about what to believe and what not to believe. More broadly, media literate people

Deepfakes pose an age-old challenge for technical experts. Often as new technologies are being developed, their dangers and benefits are uncertain and the dangers loom large.

can verify and fact check the media they consume and are, therefore, less likely to be misled. While many stakeholders, from journalists to platforms and policymakers, can contribute to increased education and media literacy, technical experts are crucial.

Because of their knowledge, technical experts are in the best position to identify the limitations of deepfakes and recommend ways that viewers and fact checkers can learn to recognize those limitations. For example, some of the early deepfake methods were not able to convincingly synthesize eyes, and so individuals could be taught to carefully examine eyes and blinking. Of course, the technology is changing rapidly (newer methods can synthesize eyes accurately), so technical experts must be at the forefront of translating the latest technical capabilities into guidelines. Technical experts could also facilitate media literacy by pushing a norm that those who publish new methods for media synthesis always include a section specifying how synthesis using the new method could be detected. Including this information in publicly available publications would facilitate media literacy.

Subject Defense. Technical experts should contribute to the development of technical strategies that help individuals avoid becoming victims of malicious deepfakes. While viewers can be deceived by deepfakes, those who are depicted in deepfakes can also be harmed. Their reputations can be severely damaged when they are falsely shown to be speaking inappropriately or engaged in sordid behavior. As well, the subjects of deepfakes have their persona (their likeness and voice) taken and used without their consent, resulting in misattribution that either exploits or denigrates their reputation according to the goals of the deepfake creator. Deepfakes may also be used to threaten and intimidate subjects.

Here there are a variety of technical approaches that experts could take. They can develop more sophisticated identity monitoring technology that could alert individuals when their likeness appears online. An individual could enroll using a sample photo, video, or audio clip, and be notified if their likeness (real or synthetic) ap-

^b See <https://bit.ly/3qh8AuC>

peared on particular platforms. Of course, this type of response would come with difficult sociotechnical challenges, including obtaining the cooperation of platforms to provide data for monitoring and addressing the resulting privacy implications. Other approaches to subject defense could involve everything from watermarking and blockchain to new techniques to limit the accessibility, usability, or viability of training data for deepfake model development. Chesney and Citron⁵ suggest the development of immutable life logs tracking subjects' behavior so that a victim can "produce a certified alibi credibly proving that he or she did not do or say the thing depicted." These are only a few suggestions; the point is that technical experts should help develop ways to counteract the negative effects of deepfakes for individuals who may be targeted.

Verification. Technical experts should develop and evaluate verification strategies, methods, and interfaces. The enormous potential of deepfakes to deceive viewers, harm subjects, and challenge the integrity of social institutions such as news reporting, elections, business, foreign affairs, and education, makes verification strategies an area of great importance.

Verification techniques can be a powerful antidote because they make it possible to identify when video, audio, or text has been manipulated. While state-of-the-art detection systems may reach accuracy in the 90%+ range,¹ they are also typically limited in scope, that is, they may work on familiar datasets but struggle to achieve comparable accuracy on unseen data or media "in the wild."⁸ For instance, a reduction in visual encoding quality, or the fine-tuning of a model on a new dataset may challenge the detector.^{2,16} Technical research on automated detection continues, with the recent Deepfake Detection Challenge drawing thousands of entries and resulting in the release of a vast dataset to help develop new algorithms.⁸ To spur work on in this area NIST has organized the Media Forensics Challenge over the past several years,^c and other workshops on Media Forensics have also convened to advance research and

Research and development of synthetic media will be better served if technical experts see themselves as part of the solution, and not the problem.

share best practices.^d Another avenue for further technical work is in building human-centered interactive tools to support semiautomated detection and verification workflows.^{9,10,17}

In practice a combination of automated and semiautomated detection may be most prudent.¹⁵ Ultimately, once verification tools are developed there will be yet another layer of sociotechnical challenges for tool deployment, from considering adversarial scenarios and access issues, to output explanations and integration with broader media verification workflows.¹²

There is no doubt that synthetic media can be used for beneficial purposes, such as in entertainment, historical reenactment, education, and training. The pressing challenge is to reap the positive uses of synthetic media while preventing or at least minimizing the harms. We are encouraged by efforts in industry and academia to grapple directly with ethics and societal impact as new innovations in synthetic media advance.^e And, as we laid out in this column, there are numerous opportunities to direct effort in butressing against some of the worst outcomes. The challenge can only be met with the sustained efforts of technical experts. Let's get to it! □

c See <https://bit.ly/3qduL5c>

d Workshop on Media Forensics; <https://bit.ly/2KCWVYb>

e For industry, see for example: <https://bit.ly/3iDIVdk>; for academia, see for example Fried, Ohad, et al. Text-based editing of talking-head video. *ACM Transactions on Graphics* 38, 4 ACM (2019), 1–14; doi:10.1145/3306346.3323028

References

1. Agarwal, S. et al. Protecting world leaders against deep fakes. Workshop on Media Forensics at CVPR. (2019).
2. Bakhtin, A. et al. Real or fake? Learning to discriminate machine from human generated text. (2019) <https://bit.ly/3iy15Q9>
3. Bateman, J. Deepfakes and synthetic media in the financial system: Assessing threat scenarios, cyber policy initiative working paper series. Carnegie Endowment for International Peace, July 2020; <https://bit.ly/3sN2DYM>
4. Brey, P. Anticipatory ethics for emerging technologies. *NanoEthics* 6, 1 (2012), 1–13.
5. Chesney, R. and Citron, D.K. Deep fakes: A looming challenge for privacy, democracy, and national security. *California Law Review* 107. (2019)
6. Cole, S. Creator of DeepNude, app that undresses photos of women, takes it offline. *Motherboard* (June 27, 2019); <https://bit.ly/393MMgy>
7. Diakopoulos, N. and Johnson, D. Anticipating and addressing the ethical implications of deepfakes in the context of elections. *New Media & Society* (2019).
8. Dolhansky, B. et al. The DeepFake Detection Challenge Dataset (2020); <https://bit.ly/3sNMGRN>
9. Gehrmann, S., Strobel, H., and Rush, A.M. GLTR: Statistical Detection and Visualization of Generated Text. (2019).
10. Groh, M. et al. Human detection of machine manipulated media. (2019); <https://bit.ly/3p6L1ot>
11. Harris, D. Deepfakes: False pornography is here and the law cannot protect you. *Duke L. & Tech. Rev.* 17 (2018), 99.
12. Leibowicz, C., Stray, J., and Saltz, E. Manipulated Media Detection Requires More Than Tools: Community Insights on What's Needed. July, 2020; <https://bit.ly/3iCsUV2>
13. Levy, S. Battle of the Clipper chip. *New York Times Magazine* 44, (1994).
14. Parnas D.L. SDI: A violation of professional responsibility. In Weiss, E.A., Ed. *A Computer Science Reader*. Springer, New York, NY, 1988.
15. Partnership on AI. A Report on the Deepfake Detection Challenge. (2020); <https://bit.ly/39STDZ0>
16. Rössler, A. et al. FaceForensics++: Learning to Detect Manipulated Facial Images. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)* (2019).
17. Sohrawardi S.J. DeFaking Deepfakes: Understanding journalists' needs for deepfake detection. In *Proceedings of the Computation + Journalism Symposium*. (2020).
18. Schweber, S. In *the Shadow of the Bomb: Bethe, Oppenheimer, and the Moral Responsibility of the Scientist* 39. Princeton University Press, 2000.
19. Tsukayama, H. McKinney, I., and Williams, J. Congress should not rush to regulate deepfakes. Electronic Frontier Foundation (June 24, 2019); <https://bit.ly/396rW03>
20. Vaccari, C. and Chadwick, A. Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social Media + Society* 6, 1 (2020).

Deborah G. Johnson (dgi7p@virginia.edu) is Olsson Professor of Applied Ethics, Emeritus, in the Department of Engineering and Society University of Virginia in Charlottesville, VA, USA.

Nicholas Diakopoulos (nad@northwestern.edu) is an Associate Professor in Communication Studies and Computer Science (by courtesy) at Northwestern University in Evanston, IL, USA.

The Profession of IT Science Is Not Another Opinion

*The issue is not who has the “truth,”
but whose claims deserve more credence.*

IS SCIENCE JUST another opinion? As the weeks unfold into months in the COVID-19 pandemic, scientists have struggled to understand the disease, how best to treat it, and how to find a vaccine. The frustration over new outbreaks and the difficulties of containing the disease have embroiled mainstream politics. Some politicians, claiming their policies are science-based, handpick scientists whose expert opinions align with their political views. Scientists appear on talk-show panels where their expert opinions are treated like the political opinions—with admiration if they agree with yours, disdain if they do not.

Treating science as if it is just another opinion is a disservice to science and to humanity. As computing professionals, we rely on science to support our work and give confidence that our systems can be trusted. What makes science different from political, journalistic, barroom, or dinner-table opinions?

Scientists investigate the natural and social worlds to understand how things work and learn their laws of operation. Many scientific laws begin as professional opinions, or hypotheses, that evolve into statements that are so well supported by evidence that no one doubts them. When this happens, the statements are called “settled science.” The profession of science has adopted a “scientific method”—a standard way of formulating and proving or disproving



scientific hypotheses. Science is open to the possibility that new evidence may disrupt settled science. In other words, science is never sure it has discovered “truth.” To look at science as a method of finding truth is hubris. The issue is not who has the “truth,” but whose claims deserve more credence.

Let us investigate why that scientists doing their best work on new questions may disagree. The disagreements hasten the journey to settling the scientific questions.

Science and Public Policy

When making important decisions governments around the world usually seek the advice of the best scientists available to apply relevant theory and data to draw conclusions on the likely outcomes of policies. Scientists, the media and the general public usually approve and applaud this commitment to “evidence-based policy.” Science is seen as stronger than the multitude of opinions from ideologies and dogmas. In this spirit governments around the

world initially took a science-based approach to the coronavirus pandemic of 2020 and frequently said, “Our policy follows the science.”

This arrangement endured for some time, especially in the early days when nobody really understood what was going on as the number of infected people soared exponentially and hospitals filled up alarmingly with sick and dying people. In the face of this existential crisis most people wanted to pull together and stand behind their governments.

Scientists in many countries built models to predict the spread of the disease and evaluate which possible interventions were most likely to contain it. In the U.K., the government turned to its Scientific Advisory Group of Experts (SAGE), which is made up of some of the most eminent and respected scientists in the country. Relying on models developed by Neil Ferguson of London’s prestigious Imperial College, SAGE advised that severely restricting citizen movement and contact was the most effective means to “flatten the curve” and keep hospitals from being overwhelmed. The U.K. government introduced a severe lockdown on March 23, 2020 that caused the most massive social and economic shock in 70 years. Many other countries followed suit and a worldwide economic depression quickly followed.

As these draconian measures were introduced dissenting voices began to be heard from citizens groups, economists, the wider scientific community, and even from within SAGE. It became common to hear scientists disagreeing with each other on the radio and TV. Now “following the science” lost its certainty as the politicians, media, and public realized that there was no single clear and authoritative scientific account. The media depicted the scientific community as a squabbling rabble. In the eyes of some science became discredited and it became obvious that government policy was being driven by political ideologies that overrode the science.

Had Science Failed?

Definitely not. Worldwide science has performed magnificently during the pandemic. In January 2020, we knew almost nothing about COVID-19. Its spread was declared to be a pandemic

Disagreement between scientists is a normal, healthy part of the scientific process.

by the World Health Organization three months later. Scientists have accumulated and synthesized a huge amount of knowledge in a very short period, much of which is not contested. With the help of massive political support, science and industry found three effective vaccines for COVID-19 in a record time of less than a year. Science is working and doing what it is supposed to do. To understand the achievements and contribution of science to the existential threat of COVID-19 it is necessary to understand the scientific process and the contingent nature of scientific knowledge.

In his 1934 book *Logik der Forschung* (The Logic of Scientific Discovery), Karl Popper established the principle of falsifiability as the main criterion distinguishing science from non-science. Falsifiability means a scientific claim is open to being shown wrong. Newton’s laws (1687) were unchallenged for the next 200 years because no one found any contrary evidence until the Michelson-Morley experiment in 1887. Then in 1905 Albert Einstein’s theory of relativity falsified Newton’s laws for objects moving close to the speed of light. Einstein’s theory inspired great skepticism. The skepticism broke in 1919 when Arthur Eddington’s solar eclipse experiment exactly confirmed the bending of light passing near the sun, an important prediction of Einstein’s theory.² Then, some of Einstein’s theory was overthrown by quantum mechanics in the mid-1920s. Popper claimed that other theories such as Marx’s economics and Freud’s psychoanalysis cannot be empirically falsified and are not science.

The falsifiability principle is not as definitive as Popper made it out to be. Scientists frequently argue over whether apparently contradictory evi-

dence is strong enough to be taken as falsification. The social sciences, rejected by Popper as sciences, by and large accept the need for evidence and for rigorous statistical testing of their hypotheses about human behavior. Statistical testing has limitations. The usual “95% confidence” means that one in 20 conclusions may not be supported by the evidence. Similarly, the statistical methods of medical sciences in double-blind clinical testing allow a small number of trials to fail as long as the vast majority support the claims.

Disagreement between scientists is a normal, healthy part of the scientific process. When something completely new like COVID-19 appears scientists will explain the early observations with a variety of theories and explanations. The scientific process culls out the theories that can be refuted and moves to a consensus on the ones strongly supported by evidence. Even among those that fit the existing observations some can be falsified by new observations. All these early theories are scientific, even though they may contradict each other. The worldwide search for a vaccine was scientific even though its outcome was uncertain because everyone was prepared to abandon a candidate vaccine if the evidence showed it did not work.

The Dual Nature of Science

There is a paradox in science that you might not have much thought about. On the one hand, the historical accounts of settled science are familiar, reasoned, and methodical. On the other hand, the actual work of investigating and verifying hypotheses is fraught and often chaotic. How can science be methodical and chaotic at the same time? How does science resolve chaos into order?

This paradox has serious implications during times when science is searching for answers that have yet to be found, as in the COVID crisis. To outsiders, it may seem the chaos indicates that the scientific process has collapsed and is not working, and that the claims of scientists cannot be trusted. In fact, the chaos is an integral part of the workings of science and dealing with it is necessary to achieve settled science.

Bruno Latour addresses this paradox in his book *Science in Action*.¹ He

makes a fundamental distinction between “ready-made-science” and “science-in-the-making.” Ready-made science, also called settled science, is the models, theories, and laws that are now taken for granted and are ready for use to build systems and make predictions about nature and the heavens. Science-in-the-making is quite unsettled because scientists do not yet know if a hypothesis is verifiable or how to go about verifying it. It is infused with uncertainties, controversies, dead ends, and fierce debates among scientists. It is highly emotional, passionate, and unsettling. Latour illustrates his point with a detailed analysis of the hypothesis that DNA is a double helix. Today that statement is taken for granted and is the basis of gene sequencing, genetic engineering, DNA analysis, CRISPR gene editing, and more. But the process of coming to this conclusion was fraught with disputes among the leading scientists, emotional name-calling, misfired claims of first discovery, and supreme disappointments about being upstaged. Latour grounds this with extensive quotes from the writings of the scientists involved at the time.

Latour depicts this dual nature of science with an image the two-faced Roman God, Janus. One face, seasoned and creased with lines of wisdom, looks back over all that has happened and tells us what is true and repeatable. The other face, youthful and brash, looks forward and tries to make sense of the unknown ahead. These opposing faces embody inverted interpretations of the world. Latour illustrates with contrasting aphorisms such as in the table here.

How does a hypothesis move from uncertainty to settled science? Favorable evidence increases confidence in the hypothesis. Unfavorable evidence decreases it. The processes of science—such as publication, exchanges at professional meetings, debates, round-

Different communities can and do evolve different statements of scientific facts based on the same evidence.

tables, extensive experimentation and testing—all contribute to removing doubt about the original hypothesis. When all the doubt has been removed and there are no remaining dissenters, the scientific community accepts the hypothesis as a fact. Latour says that the process of scientific settlement is one of hypotheses accumulating allies until there are no more dissenters. It is intensely social.

Some people do not like the notion that scientific interpretations are social inventions. They think that scientists are teasing out fundamental, immutable truths about the world. Social construction allows the possibility that different communities could adopt different systems of interpretation of the same phenomena. And exactly that has happened. Western and Chinese medicine are different systems for interpreting and treating symptoms of diseases. Biology includes a community that accepts the theory of evolution and another that accepts the theory of intelligent design by a higher being. Strong and weak artificial intelligence are different systems for interpreting machine implementations of cognitive processes. Within computing there are different communities around differ-

ent programming languages or software development processes. These communities and their interpretations are durable—when a community tries to present falsifying evidence to the other side, the other side instead finds a way to interpret that evidence as supportive of its interpretation.

A scientific interpretation must be accepted by a scientific community to be considered settled. Scientific facts are interpretations accepted by a whole scientific community with no dissenters. Different communities can and do evolve different statements of scientific facts based on the same evidence.

Working Science Pierces the Fog of Uncertainty

These two faces are integral parts of science. Science has a dual nature. We all need to be aware of it and respect it. The chaos of science-in-the-making will evolve either into settled hypotheses or rejected hypotheses.

When outsiders look in at a time of chaos, they will see hypotheses floating around but no general agreement. It will seem that scientists don’t agree—and that is exactly right. However, the disagreements do not mean that science is not working. The debates and controversies are essential to settle or reject hypotheses.

The front edge of science—the boundary region between the known and the unknown—seethes with uncertainties. Scientists must be prepared not only to apply wisely what is known, but also to find their way through the fog of uncertainty as they search for what can be known. ■

References

1. Latour, B. *Science in Action: How to Follow Scientists and Engineers through Society*. Harvard University Press, 1987.
2. *The Times Newspaper*. Revolution in science: New theory of the universe: Newtonian ideas overthrown; <https://bit.ly/3bwaV0b>

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.

Jeffrey Johnson (jeff.johnson@open.ac.uk) is Professor of Complexity Science and Design at the UK Open University. He is Vice President of the UNESCO UniTwin Complex Systems Digital Campus, an Associate Editor of ACM Ubiquity, and past president of the Complex Systems Society.

Copyright held by authors.

Contrasting aphorisms (from Latour¹).

Ready-made-science	Science-in-the-making
When things are true, they hold	When things hold, they start becoming true.
Find the most efficient system	Decide what efficiency means
Innovation is the adoption of new ideas	New ideas are the results of adoptions
Science is stronger than the multitude of opinions	How to tell which opinions hold

Viewpoint

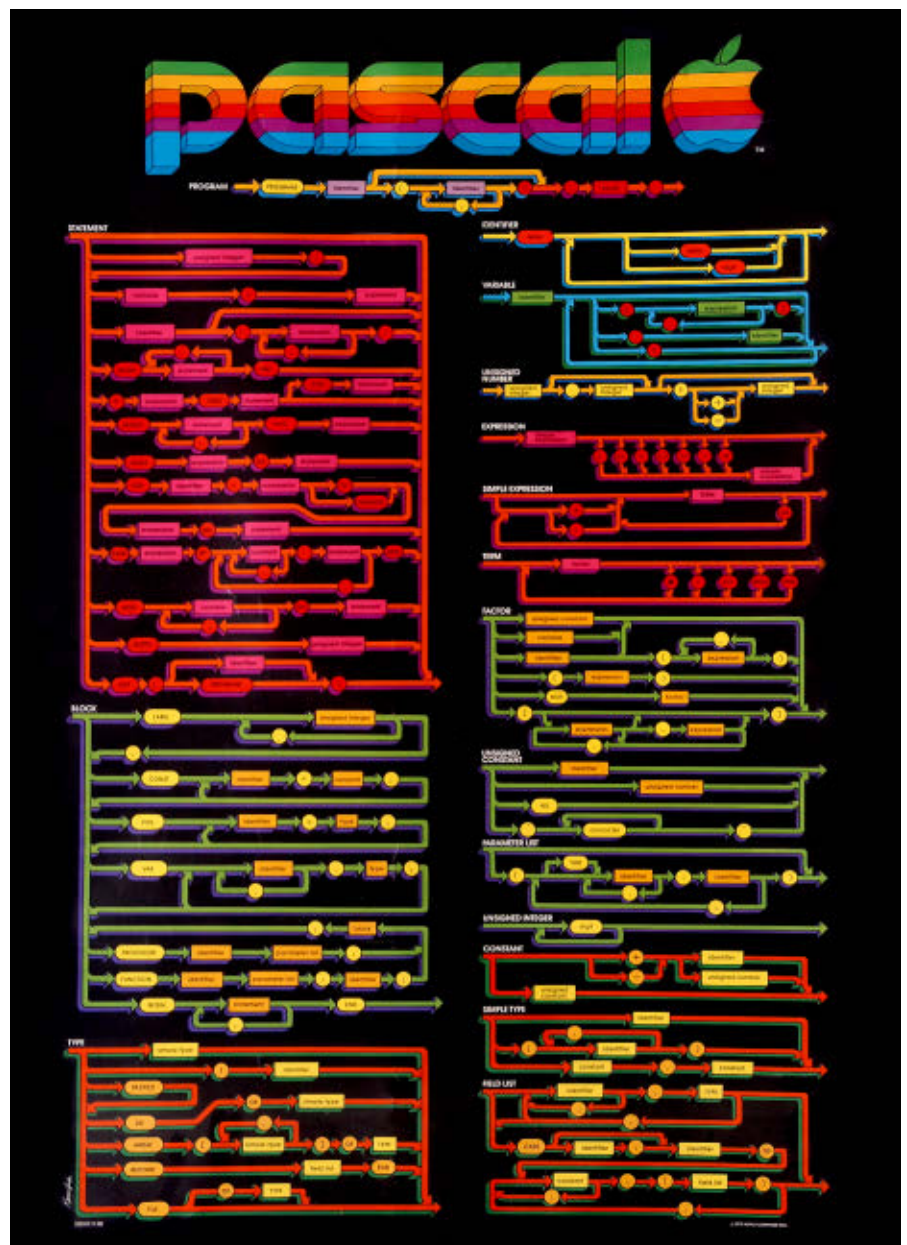
50 Years of Pascal

The Pascal programming language creator Niklaus Wirth reflects on its origin, spread, and further development.

IN THE EARLY 1960s, the languages Fortran (John Backus, IBM) for scientific, and Cobol (Jean Sammet, IBM, and DoD) for commercial applications dominated. Programs were written on paper, then punched on cards, and one waited a day for the results. Programming languages were recognized as essential aids and accelerators of the programming process.

In 1960, an international committee published the language Algol 60.¹ It was the first time a language was defined by concisely formulated constructs and by a precise, formal syntax. Two years later, it was recognized that a few corrections and improvements were needed. Mainly, however, the range of applications should be widened, because Algol 60 was intended for scientific calculations (numerical mathematics) only. Under the auspices of IFIP a Working Group (WG 2.1) was established to tackle this project.

The group consisted of about 40 members with almost the same number of opinions and views about what a successor of Algol should look like. There ensued many discussions, and on occasions the debates ended even bitterly. Early in 1964 I became a member, and soon was requested to prepare a concrete proposal. Two factions had developed in the committee. One of them aimed at a second, after Algol 60, milestone, a language with radically new, untested concepts and pervasive flexibility. It later became known as Algol 68. The other faction remained more modest and focused on realistic improvements of known concepts.



A poster of Pascal's syntax diagrams strongly identified with Pascal.

After all, time was pressing: PL/1 of IBM was about to appear. However, my proposal, although technically realistic, succumbed to the small majority that favored a milestone.

It is never sufficient to merely postulate a language on paper. A solid compiler also had to be built, which usually was a highly complex program. In this respect, large industrial firms had an advantage over our Working Group, which had to rely on enthusiasts at universities. I left the Group in 1966 and devoted myself together with a few doctoral students at Stanford University to the construction of a compiler for my proposal. The result was the language Algol W,² which after 1967 came into use at many locations on large IBM computers. It became quite successful. The milestone Algol 68 did appear and then sank quickly into obscurity under its own weight, although a few of its concepts did survive into subsequent languages.

But in my opinion Algol W was not perfectly satisfactory. It still contained too many compromises, having emerged from a committee. After my return to Switzerland, I designed a language after my own preferences: Pascal. Together with a few assistants, we wrote a user manual and constructed a compiler. In the course of it, we had a dire experience. We intended to describe the compiler in Pascal itself, then translate it manually to Fortran, and finally compile the former with the latter. This resulted in a great failure, because of the lack of data structures (records) in Fortran, which made the translation very cumbersome. After this unfortunate, expensive lesson, a second try succeeded, where in place of Fortran the local language Scallop (M. Engeli) was used.

Pascal

Like its precursor Algol 60, Pascal² featured a precise definition and a few lucid, basic elements. Its structure, the syntax, was formally defined in Extended BNF.³ Statements described assignments of values to variables, and conditional and repeated execution. Additionally, there were procedures, and they were recursive. A significant extension were data types and structures: Its elementary data types were integers and real numbers, Boolean values, charac-

ters, and enumerations (of constants). The structures were arrays, records, files (sequences), and pointers. Procedures featured two kinds of parameters, value- and variable-parameters. Procedures could be used recursively. Most essential was the pervasive concept of data type: Every constant, variable, or function was of a fixed, static type. Thereby programs included much redundancy that a compiler could use for checking type consistency. This contributed to the detection of errors, and this before the program's execution.

Just as important as addition of features were deletions (with respect to Algol). As C.A.R. Hoare once remarked: A language is characterized not only by what it permits programmers to specify, but even more so by what it does not allow. In this vein, Algol's name parameter was omitted. It was rarely used, and caused considerable complications for a compiler. Also, Algol's own concept was deleted, which allowed local variables to be global, to "survive" the activation of the procedure to which it was declared local. Algol's for statement was drastically simplified, eliminating complex and hard to understand constructs. But the while and repeat statements were added for simple and transparent situations of repetition. Nevertheless, the controversial goto statement remained. I considered it too early for the programming community to swallow its absence. It would have been too detrimental for a general acceptance of Pascal.

Pascal was easy to teach, and it covered a wide spectrum of applications, which was a significant advantage over Algol, Fortran, and Cobol. The Pascal System was efficient, compact, and easy to use. The language was strongly

Rapidly computers became faster, and therefore demands on applications grew, as well as those on programmers.

influenced by the new discipline of structured programming, advocated primarily by E.W. Dijkstra to avert the threatening software crisis (1968).

Pascal was published in 1970 and for the first time used in large courses at ETH Zurich on a grand scale. We had even defined a subset Pascal-S and built a smaller compiler, in order to save computing time and memory space on our large CDC computer, and to reduce the turnaround time for students. Back then, computing time and memory space were still scarce.

Pascal's Spread and Distribution

Soon Pascal became noticed at several universities, and interest rose for its use in classes. We received requests for possible help in implementing compilers for other large computers. It was my idea to postulate a hypothetical computer, which would be simple to realize on various other mainframes, and for which we would build a Pascal compiler at ETH. The hypothetical computer would be quickly implementable with relatively little effort using readily available tools (assemblers). Thus emerged the architecture Pascal-P (P for portable), and this technique proved to be extremely successful. The first clients came from Belfast (C.A.R. Hoare). Two assistants brought two heavy cartons of punched cards to Zurich, the compiler they had designed for their ICL computer. At the border, they were scrutinized, for there was the suspicion that the holes might contain secrets subject to custom fees. All this occurred without international project organizations, without bureaucracy and research budgets. It would be impossible today.

An interesting consequence of these developments was the emergence of user groups, mostly of young enthusiasts who wanted to promote and distribute Pascal. Their core resided under Andy Mickel in Minneapolis, where they regularly published a Pascal Newsletter. This movement contributed significantly to the rapid spread of Pascal.

Several years later the first microcomputers appeared on the market. These were small computers with a processor integrated on a single chip and with 8-bit data paths, affordable by private persons. It was recognized that Pascal was suitable for these processors, due to

its compact compiler that would fit into the small memory (64K). A group under Ken Bowles at the University of San Diego, and Philippe Kahn at Borland Inc. in Santa Cruz surrounded our compiler with a simple operating system, a text editor, and routines for error discovery and diagnostics. They sold this package for \$50 on floppy disks (Turbo Pascal). Thereby Pascal spread immediately, particularly in schools, and it became the entry point for many to programming and computer science. Our Pascal manual became a best-seller.

This spreading did not remain restricted to America and Europe. Russia and China welcomed Pascal with enthusiasm. This I became aware of only later, during my first travels to China (1982) and Russia (1990), when I was presented with a copy of our manual written in (for me) illegible characters and symbols.

Pascal's Successors

But time did not stand still. Rapidly computers became faster, and therefore demands on applications grew, as well as those on programmers. No longer were programs developed by single persons. Now they were being built by teams. Constructs had to be offered by languages that supported teamwork. A single person could design a part of a system, called a module, and do this relatively independently of other modules. Modules would later be linked and loaded automatically. Already Fortran had offered this facility, but now a linker would have to verify the consistency of data types also across module boundaries. This was not a simple matter!

Modules with type consistency checking across boundaries were indeed the primary extension of Pascal's first successor Modula-2⁴ (for modular language, 1979). It evolved from Pascal, but also from Mesa, a language developed at Xerox PARC for system programming, which itself originated from Pascal. Mesa, however, had grown too wildly and needed "taming." Modula-2 also included elements for system programming, which admitted constructs that depended on specific properties of a computer, as they were necessary for interfaces to peripheral devices or networks. This entailed sacrificing the essence of higher languages, namely machine-independent programming.

No reference to any computer or mechanism should be necessary to understand it.

Fortunately, however, such parts could now be localized in specific "low-level" modules, and thereby be properly isolated.

Apart from this, Modula contained constructs for programming concurrent processes (or quasiparallel threads). "Parallel programming" was the dominant theme of the 1970s. Overall, Modula-2 grew rather complex and became too complicated for my taste, and for teaching programming. An improvement and simplification appeared desirable.

From such deliberations emerged the language Oberon,⁵ again after a sabbatical at Xerox PARC. No longer were mainframe computers in use, but powerful workstations with high-resolution displays and interactive usage. For this purpose, the language and interactive operating system Cedar had been developed at PARC. Once again, a drastic simplification and consolidation seemed desirable. So, an operating system, a compiler, and a text editor were programmed at ETH for Oberon. This was achieved by only two programmers—Wirth and Gutknecht—in their spare time over six months. Oberon was published in 1988. The language was influenced by the new discipline of object-oriented programming. However, no new features were introduced except type extension. Thereby for the first time a language was created that was not more complex, but rather simpler, yet even more powerful than its ancestor. A highly desirable goal had finally been reached.

Even today Oberon is successfully in use in many places. A breakthrough like Pascal's, however, did not occur. Complex, commercial systems are too widely used and entrenched. But it can be claimed that many of those lan-

guages, like Java (Sun Microsystems) and C# (Microsoft) have been strongly influenced by Oberon or Pascal.

Around 1995 electronic components that are dynamically reprogrammable at the gate level appeared on the market. These field programmable gate arrays (FPGA) can be configured into almost any digital circuit. The difference between hardware and software became increasingly diffuse. I developed the language Lola (logic language) with similar elements and the same structure as Oberon for describing digital circuits. Increasingly, circuits became specified by formal texts, replacing graphical circuit diagrams. This facilitates the common design of hardware and software, which has become increasingly important in practice.

Comments and Conclusion

The principal purpose of a higher-level language is to raise the level of abstraction from that of machine instructions. Examples are data structures vs. word arrays in memory, or conditional and repetitive statements vs. jump instructions. A perfect language should be defined in terms of mathematical logic, of axioms and rules of inference. *No reference to any computer or mechanism should be necessary to understand it.* This is the basis of portability. Algol's designers saw this goal; but it is most difficult to achieve without sacrificing power of expression. Yet, any new language must be measured on the degree to which it comes close to this goal. The sequence Pascal—Modula—Oberon is witness to my attempts to achieve it. Oberon is close to it. Yet, nothing is perfect. ■

References

1. Naur, P. Revised report on the algorithmic language Algol 60. *Commun. ACM* 6, (Jan. 1963), 1–17.
2. Wirth, N. and Hoare, C.A.R. A contribution to the development of ALGOL. *Commun. ACM* 9 (June 1966), 413–432.
3. Wirth, N. The programming language Pascal. *Acta Informatica* 1, (1971), 35–63; <https://doi.org/10.1007/BF00264291>.
4. Wirth, N. What can we do about the unnecessary diversity of notation for syntactic definitions? *Commun. ACM* 20, 11 (Nov. 1977).
5. Wirth, N. *Programming in Modula-2*. Springer-Verlag 1982.
6. Wirth, N. *The Programming Language Oberon. Software-Practice and Experience* 18, (Jul. 1988), 671–690; <https://doi.org/10.1002/spe.4380180707>

Niklaus Wirth (wirth@inf.ethz.ch) is a former Professor of Informatics at ETH Zurich, Switzerland.

Copyright held by author.

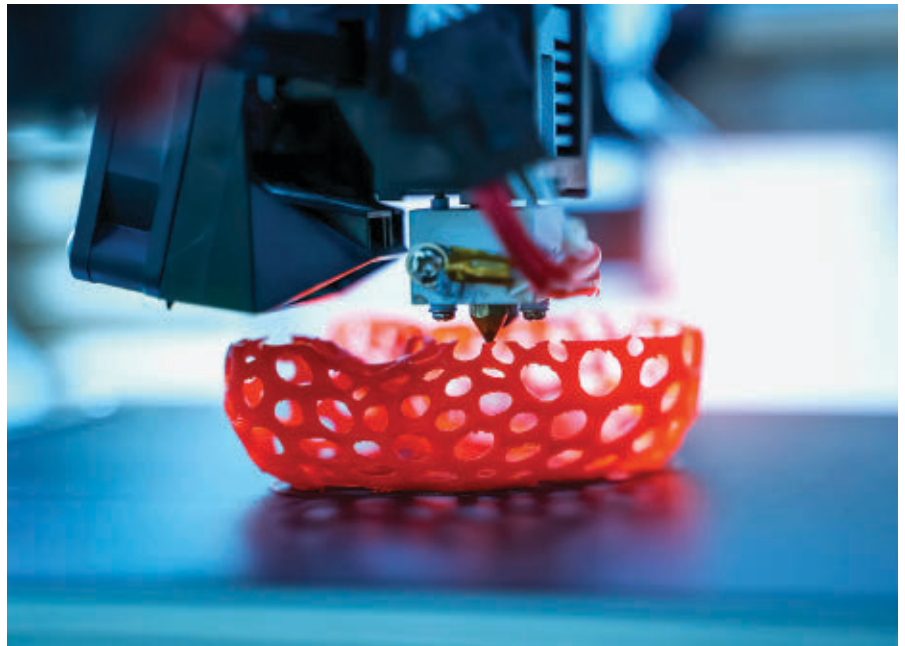
Viewpoint

What Can the Maker Movement Teach Us About the Digitization of Creativity?

Experimenting with the creative process.

IN RECENT YEARS, the ‘maker movement’ has emerged as a social phenomenon driven by novel technological possibilities.¹ With the help of inexpensive, yet highly versatile means of production (for example, CNC milling machines, 3D printers) and easy-to-use software tools, makers free themselves from their traditional role as passive consumers and evolve into innovators and producers. Although the act of physical production seems to be at the center of the movement, a large part of the creative work takes place in the online sphere. These digital activities and their outcomes provide a rich source of information that can be used to gain a more nuanced understanding of how the digitization affects the creative process itself.

Of all the production methods available to makers, 3D printing is probably the most versatile and requires only a limited understanding of the production process. Several 3D design software packages allow even lay people to turn their ideas into printable designs. This combination of flexibility and usability has led to an abundance of 3D object models over the past years, which are shared and jointly refined with the community on digital maker platforms. As part of a multi-year research project on the use of 3D printing by the maker community, we found that the use of these platforms in the



creative process blurs the boundaries between the digital and the physical and ultimately changes the way ideas are expressed, curated, and eventually translated into physical reality. In particular, we saw how makers with entirely different backgrounds (for example, HW/SW developers, designers, business and social entrepreneurs) traverse across the startup world, software development, and open online communities, to combine concepts through a novel digitized creative process.

When interviewing makers on why they started particular projects we

found their creative processes were initiated by one of two triggers: a *problem trigger* or a *curiosity trigger*. In the first case, creative efforts are made to solve a particular problem, whereas in the latter case, curiosity about the technology and enjoyment of the creative act itself are motivating factors. Traditional creativity methods like ‘design thinking’ emphasize that a creative process needs to start with an in-depth understanding of a given problem.^a Among makers we see this precondition is not

^a See <https://bit.ly/2LKkoY3>

imperative as they iteratively improve their problem-solution-fit along the way.⁷ Regardless of the trigger, the subsequent creative process can be divided into three phases: an *inspiration phase*, a *distribution phase*, and an *iteration phase*.

Inspiration Phase: Where Ideas Come From

Creativity in a digital context often makes use of prior art. It embodies the past in the present and is therefore a reflection of the social context in which it takes place. It is inherently social when makers of an online community build upon each other's work through 'remixing', a process that resembles versioning and code sharing in software repositories. An example of this can be found on Thingiverse, the world's largest 3D printing community. Here, designs are shared under open licenses (for example, CC BY) that explicitly allow remixing. We spoke to many designers who browse the platform in search for inspiration. And once they find an inspiring design they employ remixing to turn it into something new.

We found the creative processes behind these remixes are not as chaotic as one might expect, they follow distinct patterns. Remixing is either *additive*, that is, multiple ideas are combined into something new, or *subtractive*, that is, something is omitted to focus on key elements. An example of an additive remix is the debate coin in Figure 1 (a), maker *Karr* placed the mascots of the U.S.'s Republican and Democratic Party on a printable coin. By tossing the coin, a user can decide between the parties.

Remixing is also used to bring together knowledge from separate domains. On Thingiverse, designs are grouped into categories like 'Household' or 'Learning'. In many cases makers transfer ideas from one category to another where these ideas are not yet known. An example can be found in Figure 1 (b). Maker *skarab* found plant signs that allow gardeners to remember which pot contains which plant. He transferred the idea from 'Outdoor & Garden' to the 'Office' category by turning the signs into bookmarks that can be clipped to magazines, documents, or books. Instead of plant names, the bookmarks provide prompts like 'to

When interviewing makers on why they started particular projects we found their creative processes were initiated by one of two triggers: a problem trigger or a curiosity trigger.

do', 'please sign', or 'read this'.

Introducing new aspects to a field is a balancing act. If makers introduce too much newness, their designs might be hard to understand and ultimately fail. If they introduce too few novel aspects their designs are considered "nothing new" and fail as well. A similar situation is well documented in research on scientific impact where "science follows a nearly universal pattern: The highest-impact science is primarily grounded in exceptionally conventional combinations of prior work yet simultaneously features an intrusion of unusual combinations."⁶

Distribution Phase: Reaching Users Early

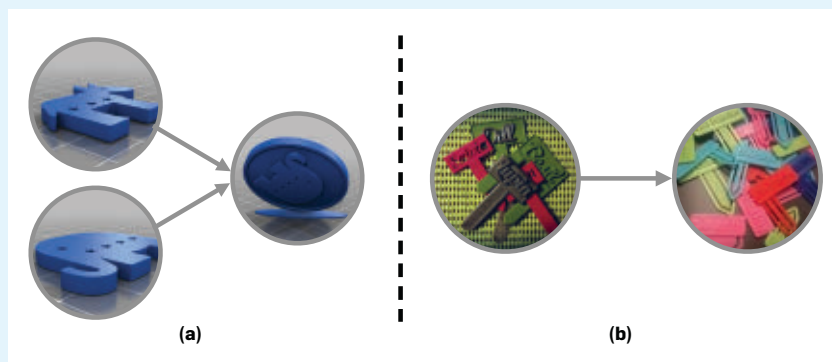
It is a notorious problem in new prod-

uct development that many new ideas fail. In some industries, like fast-moving consumer goods, failure rates are typically above 50%.⁵ The maker community is not much different, many designs fail to reach a large user base and receive little feedback. However, we saw that makers adapt to this, they understand that other people's future needs are hard to predict and in order to make meaningful contributions, they actively seek community feedback as early as possible.

In the analog age, one of the biggest obstacles for many creatives was to find an effective means for reaching potential users. Distribution was expensive and controlled by a few gatekeepers (for example, publishers or record companies). And once a product was in distribution, it was impractical to change it especially with large-scale production. However, in the maker community, production and distribution often go hand in hand. Online platforms such as *Shapeways* or *MyMiniFactory* make distribution easy for small-scale production. And the platforms typically provide the producers with a medium for receiving feedback. This is in stark contrast to traditional creative processes where only a finished solution is distributed. The digitization of creativity does not end with distribution. Rather, online platforms allow makers to distribute early in the process, which in turn allows them to iteratively improve their problem-solution-fit.

Maker Jonathan Bobrow provides

Figure 1. On the left a debate coin* that is remixed from two political mascots,** on the right plant signs*** that were remixed into bookmarkst.****



* See <http://www.thingiverse.com/thing:495777>

** See <http://www.thingiverse.com/thing:32971> and <http://www.thingiverse.com/thing:32970>

*** See <http://www.thingiverse.com/thing:1013494>

**** See <http://www.thingiverse.com/thing:1039106>

an example (see Figure 2). In 2013, Apple changed the charging ports on their laptops. To use an old charger a tiny adapter was needed, Bobrow lost his frequently. To solve this problem, he developed a key ring to hold the

adapter. Assuming that others had the same problem, he offered his key ring on the 3D printing marketplace Shapeways. As the product took off, Bobrow founded a company, improved the design and launched a

campaign on the crowdfunding platform Kickstarter.^b Hundreds of Mac users supported the campaign and made ‘keybit’ a success.

Going to users early to collect initial feedback resembles the so-called ‘Lean Startup’ concept from the startup world.⁴ Here, companies enter a market as early as possible to test their business models. They pay close attention to feedback and orientate their business to what they hear. This strategy allows them to prioritize development efforts more effectively, as the feedback they receive gives them suggestions on what to do next. All across the maker community we found similar behaviors: makers offering solutions free of charge act this way, but so do hardware developers. New printers or tools are typically introduced via crowdfunding platforms like *Kickstarter* or *Indiegogo*. Here, developers can judge the market’s reaction to their solution. Successful ideas can receive full attention, while those that do not attract audiences can offer learning opportunities before significant production costs accrue.

Iteration Phase: Steadily Improving Solutions

In the third phase of the creative process, makers iteratively improve their concepts. To do so they rely on intensive interactions with users. These interactions serve two purposes.

First, they increase the fit between ideas and user needs. In this regard, makers update existing designs to improve the problem-solution-fit. This behavior is in line with other creative communities and well documented in the field of innovation management where it is often referred to as ‘user integration.’ In one case maker *AdamStag* developed 18 versions of a single bird whistle, improving the product and answering user comments. Users for instance asked for water level marks to make the whistle more intuitive, a feature *AdamStag* promptly implemented (see Figure 3).

Secondly, makers benefit from user feedback to speed up their creative activities. Getting feedback early makes their creative processes more focused and at the same time more flexible. This

Figure 2. Jonathan Bodrow’s keybit: From a sketch to a product.

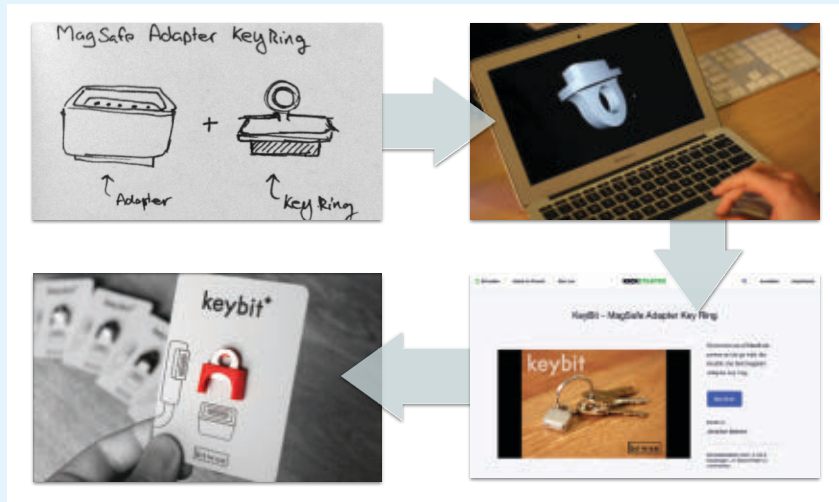


Figure 3. Feedback and new iterations of a bird whistle on the 3D printing website Thingiverse.*

AdamStag:

“Get ready for some chirping! This simple bird whistle was designed for the #MakeItLoud challenge.”



olendorf:

“Maybe add a water level mark, and text, on the outside so recipients of the whistles will know what to do.”

AdamStag:

“Great idea! I was thinking it would be really fun to print it in a semi-transparent filament ...”



tibuck:

“im sure you mind is working over drive on how , different shapes on can make unique sounds.”

AdamStag:

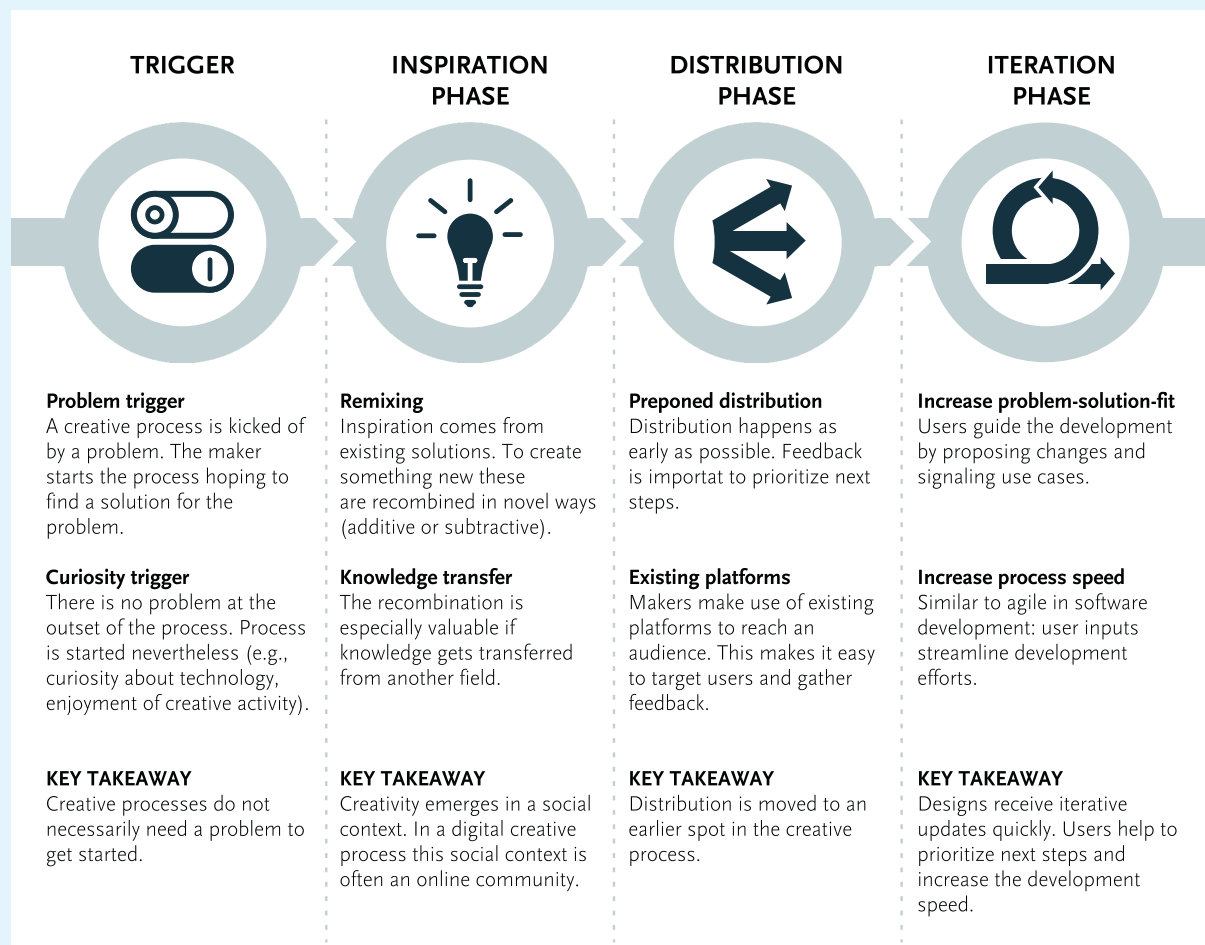
“... In December I am planning on posting a slightly modified version for the holiday season.”



* See <http://www.thingiverse.com/thing:1155687>

^b <https://bit.ly/3sOnjzB>


Figure 4. Overview of the digitized creative process



refocusing during the development process resembles concepts from software development like ‘agile.’²

Conclusion

In a world where the creative process does not end with the distribution of a product, we need to rethink our understanding of how people create. The maker community offers a great learning opportunity for all of us as they openly experiment with the creative process. But others do too: rap musician Kanye West, for instance, updated his album ‘The Life of Pablo’ after its official release date, describing this as a “living breathing changing creative expression.”³ Another pop cultural example is the original *Star Wars* trilogy, which was reedited several times after its initial release, but in this case much to the dismay of the movies’ fanbase. In the future, we will see an increasing digitization of creative processes as more and more products bridge the digital

and the physical world. The development in the maker movement is only one manifestation of this. Against this background, it is helpful to investigate how new technologies combined with concepts from software development and the startup world can help create a digitized creative process (an overview can be found in Figure 4). With the ‘digitization of the physical’ our creative processes become more fluid and, in turn, even physical goods become inspired by processes from the world of bits. The digitization of the creative process not only brings about frequent interaction with fellow designers but also provides users with the earlier versions of a prototype in an incremental and iterative manner, which over time leads to a better problem-solution-fit. Restructuring existing creative processes along these learnings will help us to keep them up to date and question decade old assumptions on how to shape and control creativity. 

References

- Anderson, C. *Makers. The New Industrial Revolution*. Crown Business, New York, NY, 2012.
- Beck, K. et al. Manifesto for agile software development. (2001).
- Helman, P. Kanye West’s Updated The Life Of Pablo Is Now On Apple Music And Spotify. Stereogum. (2016); <https://bit.ly/396s937>
- Ries, E. *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Books. (2011).
- Schneider, J. and Hall, J. Why most product launches fail. *Harvard Business Review* (Apr. 2011).
- Uzzi, B. et al. Atypical combinations and scientific impact. *Science* 342, 6157 (2013), 468–472.
- Von Hippel, E. and Von Krogh, G. Crossroads—Identifying viable need–solution pairs: Problem solving without problem formulation. *Organization Science* 27, 1 (2015), 207–221.

Sascha Friesike (s.friesike@udk-berlin.de) is a Professor of Designing Digital Innovation at the University of the Arts in Berlin, Germany and a Director of the Weizenbaum Institute for the Networked Society also in Berlin.

Frédéric Thiesse (frederic.thiesse@uni-wuerzburg.de) is a Professor of Information Systems Engineering at the University of Würzburg, Germany.

George Kuk (george.kuk@ntu.ac.uk) is a Professor of Innovation and Entrepreneurship at Nottingham Business School, U.K.

Copyright held by authors.

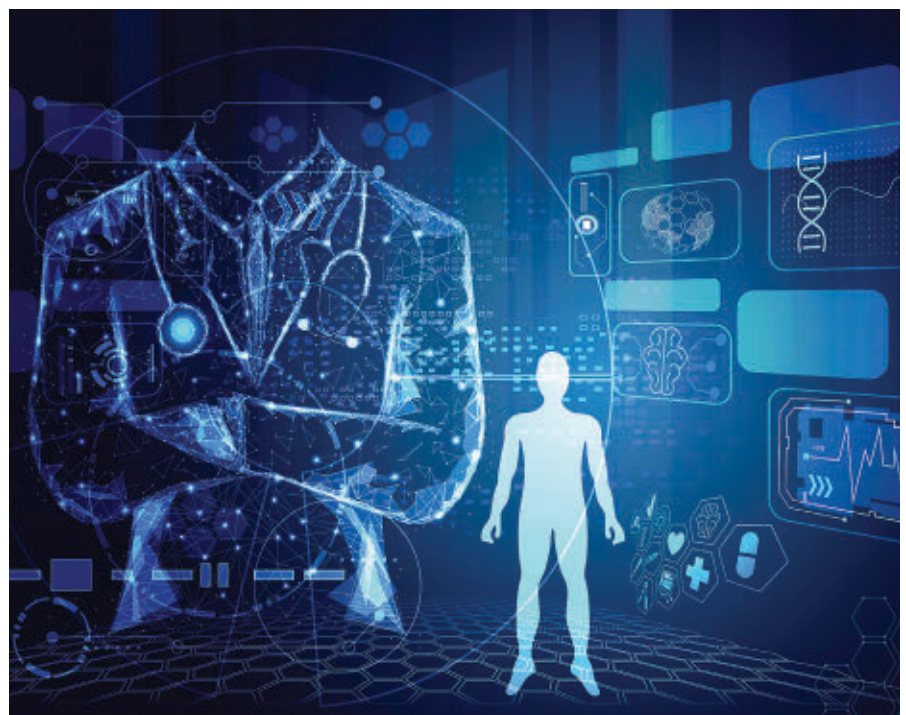
Viewpoint

The Transformation of Patient-Clinician Relationships with AI-based Medical Advice

A “bring your own algorithm” era in healthcare.

ONE OF THE dramatic trends at the intersection of computing and healthcare has been patients’ increased access to medical information, ranging from self-tracked physiological data to genetic data, tests, and scans. Increasingly however, patients and clinicians have access to advanced machine learning-based tools for diagnosis, prediction, and recommendation based on large amounts of data, some of it patient-generated. Consequently, just as organizations have had to deal with a “Bring Your Own Device” (BYOD) reality⁵ in which employees use their personal devices (phones and tablets) for some aspects of their work, a similar reality of “Bring Your Own Algorithm” (BYOA) is emerging in healthcare with its own challenges and support demands. BYOA is changing patient-clinician interactions and the technologies, skills and workflows related to them.

In this Viewpoint, we argue that BYOA is changing the patient-clinician relationship and the nature of expert work in healthcare, and better patient-clinician-information-interpretation relationships can be facilitated with solutions that integrate technological and organizational perspectives.



AI Is Changing the Patient-Provider-Information-Interpretation Relationships

Situations in which patients have direct access to algorithmic advice are becoming commonplace.⁴ However, many new tools are based on entirely new “black-box” AI-based technologies, whose inner workings are likely not fully understood by patients or

clinicians. For example, most patients with Type 1 diabetes now use continuous glucose monitors and insulin pumps to tightly manage their disease. Their clinicians carefully review the data streams from both devices to recommend dosage adjustments. Recently, however, new automated recommender systems to monitor and analyze food intake, insulin doses,

physical activity, and other factors influencing glucose levels, and provide data-intensive, AI-based recommendations on how to titrate the regimen, are in different stages of FDA approval (for example, DreaMed, Tidepool Loop), using “black box” technology—an alluring proposition for a clinical scenario that requires identification of meaningful patterns in complex and voluminous data.

But how these AI-based insights are consumed by the patient and clinician is uncharted territory, with scant population-level evidence to guide their use. Just as Bring Your Own Device can lead to incompatibility between institutional infrastructure and personal tools, with Bring Your Own Algorithm in healthcare, patients and clinicians confront cases where the AI-based advice patients obtain on their own is incompatible with best practice clinical guidelines, the clinician’s judgment, or in some cases, with prior models or algorithms used for similar medical cases.² Navigating the conflicting recommendations from population-level guidelines and individualized, algorithmic recommendations generated through a combination of advanced medical testing, patient-generated data, and AI-based systems is a challenge for which both clinicians and patients are unprepared.

The potential for unproductive contestability,⁷ where the clinician challenges the machine recommendations that are available to the patient, is concerning because the patient’s involvement may transform potentially productive differences in perspective (for example, clinicians thinking more deeply due to algorithmic advice that differs from their intuition) into personalized conflict that threatens the perceived expertise of the clinician and patient-clinician trust, and may generate uncertainty or worry for the patient. Yet contestability is likely because the machine learning models are fallible and sensitive to bias in training, and patients often lack the broader medical context within which to evaluate the algorithmic advice. As a result, the emerging BYOA reality alters clinicians’ role, emphasizing their ability to effectively interact with patients and curate, reconcile and communicate alternative interpretations of the infor-

To complement the development of patient and clinician-facing explainable systems, new occupations may be needed to serve as curators and communication bridges between patients, medical information, and clinicians.

mation and recommendation made by algorithmic advice tools.

While a wealth of information can help educate patients about their health and medical options, patients often lack the more abstract overarching background that is needed to efficiently interpret the medical information now available to them, leading to misunderstandings or errors that clinicians must correct or reconcile. Troublingly, new tools and misguided interpretation of data can erode patients’ trust in clinicians and the medical advice they provide when the AI-based tools offer alternative or conflicting diagnoses, advice, or courses of treatment.

How We Can Manage This New Reality

As BYOA profoundly alters patient-clinician-information-interpretation relationships, new thinking is required to best harness computing in a clinical interaction context. We see three complementary approaches to potential solutions, bringing together new computing-based tools and organizational practices, as described here.

The use of “black-box” tools for diagnoses and recommendation by patients and clinicians begets two unde-

sired outcomes. First, such tools are often not trusted by their clinician users because they do not understand why the tool reached certain diagnoses or recommendations. Clinician distrust may be especially likely in the BYOA situation where the algorithms patients access are unfamiliar to clinicians. Second, increasing patients’ direct access to such tools can jeopardize patients’ trust in clinicians’ judgment and advice.¹¹ One way to alleviate these concerns involves the use of explainable systems,¹ focusing on both user types (patients and clinicians). Much of the research on explainability and interpretability of black-box systems has included visualization of neural networks, analyzing machine learning systems, and training easily interpretable systems to approximate black-box systems. The intended audiences for these approaches are often computer scientists. More work is needed on how explanations should be provided to clinicians (users who do not understand the technology but are experts in the application domain) and patients (users lacking knowledge of technology and application domain).

One potential way to make explainable systems more useful is with natural language-based explanation user interfaces, via embodied and non-embodied conversational agents. In previous research,³ we found there are many complex and interacting human factors that affect non-expert user confidence in a system, including perceptions of the understandability of the explanation, its adequacy, and how intelligent and friendly the system is. The importance of these factors likely differ based on user level of domain expertise, suggesting that different explanations would be effective for patients and physicians. We need to further investigate the effects of different explanatory styles on patients and physicians in BYOA contexts in addition to improving techniques for making black-box algorithms more explainable and interpretable.

To align the information patients and clinicians are exposed to while considering the vast differences in their expertise and formal education, new tools should be developed providing patients a simplified version of the explainable systems clinicians use, as


well as tools and features that can help users determine the reliability of the algorithms used. Such new tools and features will help enhance patients' and clinicians' trust in the algorithms and understanding of their limitations, mitigate potentially unproductive contestability, and help establish a common ground for patient-clinician interaction and enhanced patient trust in clinicians.

To complement the development of patient and clinician-facing explainable systems, new occupations may be needed to serve as curators and communication bridges between patients, medical information, and clinicians. Just as new technologies in the past often led to the emergence of new occupational categories and the elimination of others,⁶ BYOA may demand new work functions whose training and day-to-day operation will integrate medical knowledge, basic understanding of machine learning, communication skills and information and curation savvy. These new healthcare team members will be trained to engage with patients around shared BYOA and explainable systems in ways that are empowering to patients without threatening clinicians. Their inclusion in a patient-focused healthcare environment will be a boon to overburdened and increasingly burned-out clinicians¹⁰ who struggle to cope with growing demands on their time.

A complementary approach treats increased patient interaction with self-diagnosis and advice tools as an opportunity to engage patients in designing future tools. BYOA systems can be a clinical healthcare goal rather than an unplanned outcome of consumer product availability, making the interaction between patients, clinicians, information, and interpretation better managed and more effective. Just as companies benefit from the insights of *lead users*⁸ who bring important user perspective and novel ideas to the design of tools companies develop, BYOA tools could benefit from patient-clinician design collaborations, in which the needs, expectations, and knowledge gaps of patients will come in close contact with the clinicians, designers, and medical informaticists who develop better—and better understood—future tools. In the spirit of user-in-the-loop patient-cen-

tered co-design,⁹ patient-clinician-designer co-design of algorithmic advice tools would focus on the design of a customizable tool whose advice content properties and presentation are adjustable to different personas and user preferences, and levels of computer and visualization literacy. Following such co-design, the adjustment of algorithmic advice tools could ultimately be made by the clinician, the patient, or in consultation between them. Such patient-in-the-loop design processes, in which patients and clinicians interact around developing BYOA prototypes, could help mitigate misguided or wrong patient self-diagnosis and data interpretation, and the stress and anxiety they can provoke.

A New Era of Computing in Healthcare

Computing has a rich history of transforming healthcare: from medical imaging to electronic health records to expert systems, computing has been facilitating major shifts in healthcare practices and tools of the trade. With data-intensive and AI-based computing tools increasingly made available directly to patients, computing is once again transforming healthcare, but this time transforming the medical expert profession and the relationship between patients and their healthcare providers. This transformation poses a number of challenges to clinicians that require new thinking about the emerging patient-clinician-information-interpretation relationships. In this Viewpoint we outline some of the key characteristics of this transformation, and possible ways to address the challenges. We acknowledge that potential solutions may require the development of new tools and roles, which may lead to new challenges, such as the need to integrate new tools into clinicians' workflow. We therefore emphasize the need for a combination of technological and organizational perspectives in scoping and developing such tools and workflows, to ensure any solution will conform to the Hippocratic Oath principle of "first, do no harm." 

References

1. Abdul, A. et al. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. *Proceedings of the 2018 ACM CHI Conference*

2. Bansal, G. et al. Updates in human-ai teams: Understanding and addressing the performance/compatibility trade-off. In *Proceedings of the AAAI Conference on Artificial Intelligence 33* (July 2019), 2429–2437.
3. Ehsan, U. et al. Rationalization: A neural machine translation approach to generating natural language explanations. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. (2018), ACM, 81–87.
4. Fraser, H., Coiera, E., and Wong, D. Safety of patient-facing digital symptom checkers. *The Lancet 392*, 10161 (2018), 2263–2264.
5. French, A., Guo, C., and Shim, J. Current status, issues, and future of bring your own device (BYOD). *Communications of the Association for Information Systems 35*, 1 (2014), 10.
6. Frey, C. and Osborne, M. The future of employment: How susceptible are jobs to computerization? *Technological Forecasting and Social Change 114*, (2017), 254–280.
7. Hirsch, T. et al. Designing contestability: Interaction design, machine learning, and mental health. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (2017), ACM, 95–99.
8. Lilien, G.L. et al. Performance assessment of the lead user idea-generation process for new product development. *Management Science 48*, 8 (2002), 1042–1059.
9. Luo, Y., Liu, P. and Choe, E.K. Co-designing food trackers with dietitians: Identifying design opportunities for food tracker customization. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), 1–13.
10. Schwenk T.L. and Gold K.J. Physician burnout—A serious symptom; but of what? *JAMA 320*, 11 (2018), 1109–1110; doi:10.1001/jama.2018.11703.
11. Vayena, E., Blasimme, A., and Cohen, I.G. Machine learning in medicine: Addressing ethical challenges. *PLoS Med 15*, 11 (2018); e1002689.

Oded Nov (onov@nyu.edu) is a professor of human-computer interaction at the NYU Tandon School of Engineering, New York, NY, USA.

Yindalon Aphinyanaphongs (Yindalon.Aphinyanaphongs@nyulangone.org) is a physician-scientist, director of Operational Data Science and Machine Learning at NYU Langone Health, and an assistant professor of Healthcare Delivery Science at the NYU Grossman School of Medicine, New York, NY, USA.

Yvonne W. Lui (Yvonne.Lui@nyulangone.org) is a practicing neuro-radiologist and an associate professor and Associate Chair for AI at the Radiology Department, NYU Grossman School of Medicine, New York, NY, USA.

Devin Mann (Devin.Mann@nyulangone.org) is an associate professor of population health at the Grossman School of Medicine and senior director for Informatics Innovation at NYU Langone Health, as well as a practicing internal medicine physician, New York, NY, USA.

Maurizio Porfiri (mporfiri@nyu.edu) is a professor of mechanical and aerospace engineering, and biomedical engineering at the NYU Tandon School of Engineering, New York, NY, USA.

Mark Riedl (riedl@cc.gatech.edu) is an associate professor of computer science in the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA.

John-Ross Rizzo (johnross.rizzo@nyulangone.org) is a physician-scientist and the director of Innovation and Technology for the Rehabilitation Medicine Department; he is an assistant professor of rehabilitation medicine and neurology (NYU Grossman School of Medicine), biomedical, and mechanical engineering (NYU Tandon School of Engineering), New York, NY, USA.

Batia Wiesenfeld (bwiesenf@stern.nyu.edu) is the Andre Koo Professor of Management and Director of the Business & Society Program at the Stern School of Business, NYU, New York, NY, USA.

This work was supported by a U.S. National Science Foundation grants #1928614, #1928586.

Copyright held by authors.

ACM Transactions on Quantum Computing (TQC)

Open for
Submissions

Publishes high-impact, original research papers and select surveys on topics in quantum computing and quantum information science



Recent advances in quantum computing have moved this new field of study closer toward realization and provided new opportunities to apply the principles of computer science. A worldwide effort is leveraging prior art as well as new insights to address the critical science and engineering challenges that face the design, development, and demonstration of quantum computing. Alongside studies in physics and engineering, the field of quantum computer science now provides a focal point for discussing the theory and practice of quantum computing.

ACM Transactions on Quantum Computing (TQC) publishes high-impact, original research papers and select surveys on topics in quantum computing and quantum information science. The journal targets the quantum computer science community with a focus on the theory and practice of quantum computing including but not limited to: quantum algorithms and complexity, models of quantum computing, quantum computing architecture, principles and methods of fault-tolerant quantum computation, design automation for quantum computing, issues surrounding compilers for quantum hardware and NISQ implementation, quantum programming languages and systems, distributed quantum computing, quantum networking, quantum security and privacy, and applications (e.g. in machine learning and AI) of quantum computing.

For more
information
and to submit
your work,
please visit:

tqc.acm.org



Association for
Computing Machinery

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

The Amazon CTO sits with Tom Killalea to discuss designing for evolution at scale.

A Second Conversation with Werner Vogels

WHEN I JOINED Amazon in 1998, the company had a single U.S.-based website selling only books and running a monolithic C application on five servers, a handful of Berkeley DBs for key/value data, and a relational database. That database was called “ACB” which stood for “Amazon.Com Books,” a name that failed to reflect the range of our ambition. In 2006, *acmqueue* published a conversation between Jim Gray and Werner Vogels, Amazon’s CTO, in which Vogel explained that Amazon should be viewed not just as an online bookstore but as a technology company. In the intervening 14 years, Amazon’s distributed systems, and the patterns used to build and operate them, have grown in influence. In this follow-up conversation, Vogel and I pay particular attention to the lessons to be learned from the evolution of a single distributed system—Simple Storage Service (S3)—that was publicly launched close to the time of that 2006 conversation.

—Tom Killalea

TOM KILLALEA: In your keynote at the AWS re:Invent conference in December 2019, you said that in March 2006 when it launched, S3 was made up of eight services, and by 2019 it was up to 262 services. As I sat there I thought that’s a breathtaking number, and it struck me that very little has been written about how a large-scale, always-on service evolves over a very extended period of time. That is a journey that would be of great interest to our software practitioner community. This is evolution at a scale that is unseen and certainly hasn’t been broadly discussed.

WERNER VOGELS: I absolutely agree that this is unparalleled scale. Even today, even though there are Internet services these days that have reached incredible scale—I mean look at Zoom, for example [this interview took place over Zoom]—I think S3 is still two or three generations ahead of that. And why? Because we started earlier; it’s just a matter of time, and at the same time having a strict feedback loop with your customers that continuously evolves the service. Believe me, when we were designing it, when we were building it, I don’t think that anyone anticipated the complexity of it eventually. I think what we did realize is that we would not be running the same architecture six months later, or a year later.

So, I think one of the tenets up front was don’t lock yourself into your architecture, because two or three orders of magnitude of scale and you will have to rethink it. Some of the things we did early on in thinking hard about what an evolvable architecture would be—something that we could build on in the future when we would be adding functionality to S3—were revolutionary. We had never done that before.

Even with Amazon the Retailer, we had unique capabilities that we wanted to deliver, but we were always quite certain where we wanted to go. With S3, nobody had done that before, and remember when we were in the room designing it, [AWS Distinguished Engineer] Al Vermeulen put a number on the board: the number of objects that

we would be storing within six months.

KILLALEA: I remember this conversation.

VOGELS: We put two additional zeroes at the end of it, just to be safe. We blew through it in the first two months.

A few things around S3 were unique. We launched with a set of ten distributed systems tenets in the press release. (See sidebar, “Principles of Distributed System Design.”)

That was quite unique, building a service that was fundamentally sound such that you could evolve on top of it. I think we surprised ourselves a bit.

The eight services were really just the fundamental pieces to get, put, and manage incoming traffic. Most importantly, there are so many different tenets that come with S3, but durability, of course, trumps everything. The eleven 9s (99.999999999%) that we promise our customers by replicating over three availability zones was unique. Most of our customers, if they have on-premises systems—if they’re lucky—can store two objects in the same data center, which gives them four 9s. If they’re really good, they may have two data centers and actually know how to replicate over two data centers, and that gives them five 9s. But eleven 9’s, in terms of durability, is just unparalleled. And it trumps everything. The need for durability also means that for example, one of the eight microservices would be the one that continuously checks all the objects, all the CRCs (cyclic redundancy checks), and there are trillions and trillions of objects by now. There’s a worker going around continuously checking in case an object had some bit rot or something like that.

One of the biggest things that we learned early on is—and there’s this quote that I use—“Everything fails, all the time.” Really, everything fails, all the time, in unexpected ways, things that I never knew. Bit flips in memory, yes. You need to protect individual data structures with a CRC or checksum on it because you can’t trust the data in it anymore. TCP (Transmission Control Protocol) is supposed to be reliable and



not have any flips in bits, but it turns out that’s not the case.

KILLALEA: Launching with distributed-systems tenets was unique. Fourteen years later, would the tenets be different? There’s this expectation that tenets should be evergreen; would there be material changes?

VOGELS: Not these; these are truly fundamental concepts that we use in distributed systems. The ten tenets were separate from S3, stating that this is how you would want to build distributed systems at scale. We just demonstrated that S3 is a really good example of applying those skills.

Some of the other tech companies that were scaling at the same time, search engines and so on, in general had only one task, such as do search really well. In the case of Amazon the Retailer, we had to do everything: robotics, machine learning, high-volume transaction processing, rock-solid delivery of Web pages, you name it. There isn’t a technology in a computer science textbook that wasn’t pushed to the edge at Amazon.com. We were operating at unparalleled scale,

with really great engineers—but they were practical engineers—and we made a change before building S3 to go back to fundamentals, to make sure that what we were building was fundamentally sound because we had no idea what it was going to look like in a year. For that we needed to have a really solid foundation.

KILLALEA: One of the keys to the success of S3 was that, at launch, it was as simple as it possibly could be, offering little more than GetObject and PutObject. At the time that was quite controversial, as the offering seemed almost too bare bones. With the benefit of hindsight, how do you reflect on that controversy, and how has that set up S3 to evolve since then? You mentioned *evolvable architecture*.

VOGELS: List is the other one that goes with Get and Put, prefixed List.

KILLALEA: Right. Could it possibly have been simpler at launch?

VOGELS: It was slightly controversial, because most technology companies at the time were delivering everything and the kitchen sink, and it would come with a very thick book and 10 different part-

Principles of Distributed System Design

Amazon used the following principles of distributed system design to meet Amazon S3 requirements:²

- ▶ **Decentralization.** Use fully decentralized techniques to remove scaling bottlenecks and single points of failure.
- ▶ **Asynchrony.** The system makes progress under all circumstances.
- ▶ **Autonomy.** The system is designed such that individual components can make decisions based on local information.
- ▶ **Local responsibility.** Each individual component is responsible for achieving its consistency; this is never the burden of its peers.
- ▶ **Controlled concurrency.** Operations are designed such that no or limited concurrency control is required.
- ▶ **Failure tolerant.** The system considers the failure of components to be a normal mode of operation and continues operation with no or minimal interruption.
- ▶ **Controlled parallelism.** Abstractions used in the system are of such granularity that parallelism can be used to improve performance and robustness of recovery or the introduction of new nodes.
- ▶ **Decompose into small, well-understood building blocks.** Do not try to provide a single service that does everything for everyone, but instead build small components that can be used as building blocks for other services.
- ▶ **Symmetry.** Nodes in the system are identical in terms of functionality, and require no or minimal node-specific configuration to function.
- ▶ **Simplicity.** The system should be made as simple as possible, but no simpler.

ners that would tell you how to use the technology. We went down a path, one that Jeff [Bezos] described years before, as building *tools* instead of *platforms*. A platform was the old-style way that large software platform companies would use in serving their technology.

If you would go from Win32 to .NET, it was clear that the vendor would tell you exactly how to do it, and it would come with everything and the kitchen sink—not small building blocks but rather, “This is how you should build software.”

A little before we started S3, we began to realize that what we were doing might radically change the way that software was being built and services were being used. But we had no idea how that would evolve, so it was more important to build small, nimble tools that customers could build on (or we could build on ourselves) instead of having everything and the kitchen sink ready at that particular moment. It was not necessarily a timing issue; it was much more that we were convinced that whatever we would be adding to the interfaces of S3, to the functionality of S3, should be driven by our customers—and how the next gen-

eration of customers would start building their systems.

If you build everything and the kitchen sink as one big platform, you build with technology that is from five years before, because that’s how long it takes to design and build and give everything to your customers. We wanted to move much faster and have a really quick feedback cycle with our customers that asks, “How would you develop for 2025?”

Development has changed radically in the past five to ten years. We needed to build the right tools to support that rate of radical change in how you build software. And with that, you can’t predict; you have to work with your customers, to wait to see how they are using your tools—especially if these are tools that have never been built before—and see what they do. So, we sat down and asked, “What is the minimum set?”

There’s one other thing that I want to point out. One of the big differences between Amazon the Retailer and AWS in terms of technology is that in retail, you can experiment the hell out of things, and if customers don’t like it, you can turn it off. In AWS you can’t do that. Customers are going to build their busi-

nesses on top of you, and you can’t just pull the plug on something because you don’t like it anymore or think that something else is better.

You have to be really consciously careful about API design. APIs are forever. Once you put the API out there, maybe you can version it, but you can’t take it away from your customers once you’ve built it like this. Being conservative and minimalistic in your API design helps you build fundamental tools on which you may be able to add more functionality, or which partners can build layers on top of, or where you can start putting different building blocks together. That was the idea from the beginning: to be so minimalistic that we could allow our customers to drive what’s going to happen next instead of us sitting in the back room thinking, “This is what the world should look like.”

KILLALEA: The idea of being minimalistic in defining an MVP (minimum viable product) has gained broad adoption now, but S3 at launch pushed it to the extreme. In those early days there was some discussion around which persistence service the AWS team should bring to market first: an object store or a key-value store or a block store. There was a sense that eventually each would be out there, but there’s a necessary sequencing in a small team. Launching S3 first was done very intentionally, with EBS (Elastic Block Store), for example, following in August 2008. Can you share with us the rationale?

VOGELS: Quite a bit of that is learning from how we had built systems ourselves, where a key-value store was the most crucial. After our “mishap” with one of our database vendors in December 2004, we decided to take a deep look at how we were using storage, and it turned out that 70 percent of our usage of storage was key-value. Some of those values were large, and some were really small. One of those drove in the direction of Dynamo, in terms of small keys, a table interface, things like that, and the other one became S3, with S3 more as a blob and bigger value store, with some different attributes.

One of the big winners in the early days of S3 was direct HTTP access to your objects. That was such a winner for everyone because now suddenly on every Web page, app, or whatever, you could pull your object in just by us-

ing HTTP. That was unheard of. Maybe there were things at the beginning that we thought would be more popular and that didn't turn out to be the case—for example, the BitTorrent interface. Did it get used? Yes, it did get used. But did it get used massively? No. But we launched FTP access, and that was something that people really wanted to have.

So, sometimes it seems like not very sexy things make it, but that's really what our customers are used to using. Again, you build a minimalistic interface, and you can build it in a robust and solid manner, in a way that would be much harder if you started adding complexity from day one, even though you know you're adding something that customers want.

There were things that we didn't know on day one, but a better example here is when we launched DynamoDB and took a similar minimalistic approach. We knew on the day of the launch that customers already wanted secondary indices, but we decided to launch without it. It turned out that customers came back saying that they wanted IAM (Identity and Access Management)—access control on individual fields within the database—much more than they wanted secondary indices. Our approach allows us to reorient the roadmap and figure out the most important things for our customers. In the DynamoDB case it turned out to be very different from what we thought.

KILLALEA: I think that much of this conversation is going to be about evolvability. As I listened to you at re:Invent, my mind turned to Gall's Law: "A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over with a working simple system." How do you think this applies to how S3 has evolved?

VOGELS: That was the fundamental thinking behind S3. Could we have built a complex system? Probably. But if you build a complex system, it's much harder to evolve and change, because you make a lot of long-term decisions in a complex system. It doesn't hurt that much if you make long-term decisions on very simple interfaces, because you can build on top of them. Complex systems are much harder to evolve.

Let me give you an example. One of the services added to S3 was auditing capability—auditing for whether your objects are still fresh and alive and not touched, or whatever. That was the first version of auditing that we did. Then we started to develop CloudTrail (launched in November 2013), which had to be integrated into S3. If you've built a complex system with all of these things in a monolith or maybe in five monoliths, that integration would be a nightmare, and it would definitely not result in a design that you are comfortable with evolving over time.

Mai-Lan Tomsen Bukovec [vice president, AWS Storage] has talked about a culture of durability. For example, within S3, durability trumps everything, even availability. Imagine if the service were to go down: You cannot lose the objects. Your data cannot disappear; maybe it takes you five minutes to get access to it again, but your objects should always be there. Mai-Lan's team has a culture of durability, which means that they use tools such as TLA+ to evaluate their code to see whether its algorithms are doing exactly what they're supposed to be doing.

Now let's say, to make it simple, you have a 2,000-line algorithm. That's something you can evaluate with formal verification tools; with 50,000 lines, forget about it. Simple building blocks allow you to have a culture that focuses exactly on what you want to do, whether it's around auditing or whether it's around using TLA+ or durability reviews or whatever. Everything we change in S3 goes through a durability review, making sure that none of these algorithms actually does anything other than what we want them to do.

KILLALEA: With full-blown formal verification?

VOGELS: Here's a good example in the context of S3. If you look at libssl, it has a ridiculous number of lines of code, with something like 70,000 of those involved in processing TLS. If you want to create vulnerabilities, write hundreds of thousands of lines of code. It's one of the most vulnerable access points in our systems.

KILLALEA: I know that there's a plug for S2N coming.

VOGELS: Yes, so we wrote S2N, which stands for signal-to-noise, in 5,000 lines. Formal verification of these 5,000 lines can tell exactly what it does. Now

everything on S3 runs on S2N, because we have way more confidence in that library—not just because we built it ourselves but because we use all of these additional techniques to make sure we can protect our customers. There is end-to-end encryption over every transfer we do. In how you use encrypted storage, do you want us to create the keys? Do you want to bring the keys and give them to us? Do you want to bring the keys and put them in a KMS (key management service)? Or do you want to completely manage your keys? I'm pretty sure that we started off with one, and customers started saying, "But that, and that, and that." You need those too.

If you build this as an evolvable architecture with small microservices, you can still allow encryption at rest just to do its job, and then you can think about how to start adding other services that may do other things—like life-cycle management from S3 down to Glacier. If this object hasn't been touched in 30 days, move it to reduced instance storage; and if it then hasn't been touched for another two months, automatically move it to Glacier.

KILLALEA: You launched S3 Object Versioning in February 2010. How did that fit into the evolving expectations of the service, how customers wanted to use it, and the architectural demands that posed?

VOGELS: It was mostly a back-and-forth with our customer base about what would be the best interface—really listening to people's requirements. And to be honest, immutability was a much bigger requirement than having a distributed lock manager, which is notoriously hard to build and operate. It requires a lot of coordination among different partners, and failure modes are not always well understood.

So, we decided to go for a simpler solution: object versioning, officially called S3 Object Lock. There are two things that you can do to a locked object. First, once you create it you can only change it, which in the world of blockchain and things like that is a very interesting concept. You can also set two attributes on it: one is the retention period (for example, this cannot be deleted for the coming 30 days); and another is LegalHold, which is independent of retention periods and basically says that this object cannot be deleted until an authorized

user explicitly takes an action on it.

It turns out that object versioning is extremely important in the context of regulatory requirements. You may need to be able to tell your hospital or regulatory examiners that this object is being kept in live storage for the coming six months, and then it is moved to cold storage for the 30 years after. But being able to prove to the regulator that you are actually using technology that will still be alive in 30 years is quite a challenge, and as such we've built all of these additional capabilities in there.

KILLALEA: The absence of traditional explicit locking has shifted responsibility to developers to work around that in their code, or to use versioning. That was a very intentional decision.

VOGELS: It was one of these techniques that we used automatically in the 1980s and 1990s and maybe in the early 2000s—the distributed lock managers that came with databases and things like that. You might have been using a relational database because that was the only tool you had, and it came with transactions, so you used transactions, whether you needed them or not. We wanted to think differently about an object store, about its requirements; and it turns out that our approach gave customers the right tools to do what they wanted to do, unless they really wanted lock and unlock, but that's not something that can scale easily, and it's hard for our customers to understand. We went this different way, and I've not heard a lot of complaints from customers.

KILLALEA: S3 launched more than four years before the term *data lake* was first used in a late 2010 blog post by James Dixon.⁵ S3 is used by many enterprise data lakes today. If they had known what was coming, would it have been helpful or distracting for the 2006 S3 team to try to anticipate the needs of these data lakes?

VOGELS: We did a number of things in the early days of AWS in general—it has nothing to do necessarily with S3—where there are a few regrets. For example, I am never, ever going to combine account and identity at the same time again. This was something we did in the early days; we didn't really think that through with respect to how the system would evolve. It took us quite a while actually to rip out accounts. An account is something you bill to; identity is some-

thing you use in building your systems. These are two very different things, but we didn't separate them in the early days; we had one concept there. It was an obvious choice in the moment but the wrong choice.

Here is another interesting example with S3. It's probably the only time we changed our pricing strategy. When we launched S3, we were charging only for data transfer and data storage. It turned out that we had quite a few customers who were storing millions and millions of thumbnails of products they were selling on eBay. There was not much storage because these thumbnails were really small, and there wasn't much data transfer either, but there were enormous numbers of requests. It made us learn, for example, when you design interfaces, and definitely those you charge for, you want to charge for what is driving your own cost. One of the costs that we didn't anticipate was the number of requests, and request handling. We added this later to the payment model in S3, but it was clearly something we didn't anticipate. Services that came after S3 have been able to learn the lessons from S3 itself.

Going back to the concept of a data lake, I wouldn't do anything else actually, except for those two things, mostly because I think we have created the basic building blocks since S3 serves so much more than data lakes. There are a few interesting parts, in terms of the concepts in data lakes, where I think S3 works well under the covers, but you need many more components to build a data lake. In terms of building a data lake, for example, Glue is an equally important service that sits next to S3, discovers all of your data, manages your data, lets you decide who has access to which data, and whether you need to pull this from on-premises, or does it need to come out of a relational database, and all of these kinds of things.

It turns out that you need a whole lot of components if you really want to build a mature data lake. It's not just storing things in S3. That's why we built Lake Formation. One of the things you see happening both at AWS and with our partners is that now that we have this massive toolbox—175 different services—they've always been intended as small building blocks. This makes them sometimes hard to use because they're not really solutions, they're ba-

sic building blocks. So, to build a data lake, you need to put a whole bunch of these things together. What you see now is that people are building solutions to give you a data lake.

We have to remember that S3 is used for so much more than that, whether it's a content lake with massive video and audio files, or a data lake where people are storing small files, or maybe it's a data lake where people are doing genomics computation over it. One of our customers is sequencing 100 million human genomes. One human genome is 100 GB; that's just raw data—there's nothing else there, so a lot of things have to happen to it. In life sciences, files are getting huge. If I look at the past, or if I look at that set of customers that start to collect that set of data, whether it's structured or unstructured data, quite a few of them are starting to figure out how to apply machine learning to their data. They are not really there yet, but they may want to have the data stored there and start making some use of Redshift or EMR or Kinesis or some other tools to do more traditional approaches to analytics. Then they might be prepared for a year from now when they've trained their engineers and are ready to apply machine learning to these datasets.

These datasets are getting so large—and I'm talking here about petabytes or hundreds of petabytes in a single data file—and requirements are shifting over time. When we designed S3, one of its powerful concepts was separating compute and storage. You can store the hell out of everything, but your compute doesn't need to scale with it, and can be quite nimble in terms of compute. If you store more, you don't need more EC2 (Elastic Compute Cloud) instances.

With datasets becoming larger and larger, it becomes more interesting to see what can be done inside S3 by bringing compute closer to the data for relatively simple operations. For example, we saw customers retrieving tens if not hundreds of petabytes from their S3 storage, then doing a filter on it and maybe using five percent of the data in their analytics. That's becoming an important concept, so we built S3 Select, which basically does the filter for you at the lowest level and then moves only the data that you really want to operate on.

Similarly, other things happened in our environment that allowed us to ex-

tend S3. In the Lambda and Serverless components, the first thing we did was fire up a Lambda function when a file arrives in S3. The ability to do event-driven triggering and extend S3 with your own functions and capabilities without having to run EC2 instances made it even more powerful, because it's not just our code that runs there, it's your code. There's a whole range of examples where we go under the covers of S3 to run some compute for you in case you want that.

KILLALEA: This concept of extensibility is really key in terms of the lessons that our readers could take away from this journey. I know there were some examples starting with bare bones in the case of S3 and learning from the requests of a few very early and demanding adopters such as Don MacAskill at SmugMug and Adrian Cockcroft, who at the time was at Netflix. Are there other examples of situations where customer requests made you pop open your eyes and say, "That's interesting; I didn't see that coming," and it became a key part of the journey?

VOGELS: There are other examples around massive high-scale data access. To get the performance they needed out of S3, some customers realized that they had to do some randomization in the names. They would pre-partition their data to get the performance they were looking for, especially the very high-volume access people.

It's now been three years since we made significant changes in how partitioning happens in S3, so that this process is no longer needed. If customers don't do pre-partitioning themselves, we now have the opportunity to do partitioning for them through observability. We observe what's happening and may do very quick rereplication, or repartitioning, to get the right performance to our customers who in the past had to figure it out by themselves.

With our earliest customers, we looked at that particular behavior and realized we had to fix this for them. Indeed, people like Don [MacAskill] have been very vocal but also very smart technologists. Such developers knew exactly what they wanted to build for their businesses, and I think we thrived on their feedback.

Today it may be, for example, telemedicine, which needs HIPAA (Health Insurance Portability and Accountability Act) compliance and other regulatory



We wanted to think differently about an object store, about its requirements; and it turns out that our approach gave customers the right tools to do what they wanted to do.



requirements; it needs to be sure about how the data is stored and so on. We've started to build on this so that we could easily use a microservices architecture to test for auditors whether we are meeting HIPAA or PCI DSS (Payment Card Industry Data Security Standard) or another compliance specification in realtime.

Amazon Macie, for example, is one of these services. The capabilities sit in S3 to review all of your data, discover which is personally identifiable information, intellectual property, or other things, and we can use machine learning to discover this. Why? Every customer is different. It's not just discovering what they tell you they have; it's discovering the access patterns to the data, and when we see a change in the access patterns to your data that may signal a bad actor coming in. We could build all of these things because we have this microservices architecture; otherwise, it would be a nightmare.

KILLALEA: In a 2006 conversation for *acmqueue* with Jim Gray, you talked about how the team "is completely responsible for the service—from scoping out the functionality, to architecting it, to building it, and operating it. You build it, you run it. This brings developers into contact with the day-to-day operation of their software."⁶ I know that you remember that conversation fondly. That continues to be among our most widely read articles even today. There's universal relevance in so many of the concepts that came up in it.

VOGELS: That conversation with Jim was great. It wasn't so much about AWS. It was much more about retail, about experimentation, and making sure that your engineers who are building customer-facing technology are not sitting in the back room and handing it off to someone else, who is then in contact with the customers. If your number-one leadership principle is to be customer obsessed, then you need to have that come back into your operations as well. We want everybody to be customer obsessed, and for that you need to be in contact with customers. I do also think, and I always joke about it, that if your alarm goes off at 4 am, there's more motivation to fix your systems. But it allows us to be agile, and to be really fast.

I remember during an earlier period in Amazon Retail, we had a whole year where we focused on performance, es-

pecially at the 99.9 percentile, and we had a whole year where we focused on removing single points of failure, but then we had a whole year where we focused on efficiency. Well, that last one failed completely, because it's not a customer-facing opportunity. Our engineers are very well attuned to removing single points of failure because it's good for our customers, or to performance, and understanding our customers. Becoming more efficient is bottom-line driven, and all of the engineers go, "Yes, but we could be doing all of these other things that would be good for our customers."


KILLALEA: Right. That was a tough program to lead.

VOGELS: That's the engineering culture you want. Of course, you also don't want to spend too much money, but I remember that bringing product search from 32 bits to 64 bits immediately resulted in needing only a third of the capacity, but, most importantly, Amazon engineers are attuned to what's important to our customers. That comes back to our technology operational model as well; of course, DevOps didn't exist before that. All of these things came after that.


Probably one of the reasons that that *acmqueue* article is popular is because it was one of the first times we talked about this. The reaction was similar when we wrote the Dynamo paper.⁴ The motivation for writing it was not really to promote Amazon but to let engineers know what an amazing environment we had to build the world's largest scalable distributed systems, and this was even before AWS. One of my hardest challenges, yours as well in those days, was hiring. You couldn't hire engineers because, "Why, you're a [expletive] bookshop!" I know from myself as an academic, I almost wouldn't give a talk at Amazon. Why? "A database and a Web server, how hard can it be?"

It wasn't until we started talking about these kinds of things publicly that the tide started to shift in our ability not only to hire more senior engineers, but also to have people excited about it: "If you want to build really big stuff, you go to Amazon." I think now with AWS, it's much easier; everybody understands that. But in those days, it was much harder.

KILLALEA: That initial S3 team was a single agile team in the canonical sense,



With datasets becoming larger and larger, it becomes more interesting to see what can be done inside S3 by bringing compute closer to the data for relatively simple operations.



and in fact was quite a trailblazer when it came to agile adoption across Amazon.

VOGELS: Yes.

KILLALEA: And the "You build it, you run it" philosophy that you discussed with Jim applied to the developers on the S3 team in 2006. They all knew each of those initial eight services intimately and would have been in a position to take a first pass at debugging any issue. As the complexity of a system increases, it becomes harder for any individual engineer to have a full and accurate model of that system. Now that S3 is not a single team but a big organization, how can an engineer reason with and model the whole?

VOGELS: As always, there's technology, there's culture, and there's organization. The Amazon culture is well known, technology we don't need to talk that much about, so it's all about organization. Think about the kinds of things we did early on at Amazon, with the creation of the culture around principal engineers. These are people who span more than one service, people who have a bigger view on this, who are responsible for architecture coherence—not that they tell other people what to do, but at least they have the knowledge.

If you have a team in S3 that is responsible for S3 Select, that is what they do. That's what you want. They may need to have deep insight in storage technology, or in other technologies like that, or even in the evolution of storage technologies over time, because the way that we were storing objects in 2006 is not the same way we're storing objects now. But we haven't copied everything; you can't suddenly start copying exabytes of data because you feel that some new technology may be easier to use. There is some gravity with the decisions that you make as well, especially when it comes to storage.

Principal engineers, distinguished engineers—these are roles that have evolved over time. When I joined, we didn't have a distinguished engineer. Over time we started hiring more senior leaders, purely with the goal not necessarily of coding on a day-to-day basis, but sort of being the advisor to these teams, to multiple teams. You can't expect the S3 Select team to have sufficient insight into exactly what the auditing capabilities of Macie are, but you do need to have people in your organization who are allowed to roam more freely on top of this.

Our decentralized nature makes it easy to move fast within the particular area of responsibility of your team; the downside of decentralization is coordination. Now suddenly, you need to invest in coordination because these teams are small and nimble and agile and fast-moving, and they don't have additional people to help with coordination.

In the past at Amazon we had a few of these cases; when Digital (for example, Kindle or Amazon Video) wanted to add something to the order pipeline, a physical delivery address was required. There was no way around it. They would walk to the 80 different ordering teams and say, "We need to change this." The ordering teams would respond that they hadn't budgeted for it. One of the consequences was we allowed duplication to happen. We allowed the Digital team to build their own order pipeline for speed of execution. There are advantages; otherwise, we wouldn't be doing it. There are disadvantages as well.

Sharing knowledge, principal engineers and distinguished engineers help with this. But sometimes people go to the wiki and read about your old API, not knowing that it's the old API, and then start hammering your service through the old API that you thought you had deprecated.

Information sharing, coordination, oversight, knowing what else is going on and what are the best practices that some of the other teams have developed—at our scale, these things become a challenge, and as such you need to change your organization, and you need to hire into the organization people who are really good at I won't say oversight because that implies that they have decision control, but let's say they are the teachers.

KILLALEA: Approachability is a key characteristic for a principal engineer. Even if junior engineers go through a design review and learn that their design is terrible, they should still come away confident that they could go back to that same principal engineer once they believe that their reworked design is ready for another look.

VOGELS: Yes.

KILLALEA: Could you talk about partitioning of responsibility that enables evolution across team boundaries, where to draw the boundaries and to scope appropriately?

VOGELS: I still think there are two other angles to this topic of information sharing that we have always done well at Amazon. One, which predates AWS of course, is getting everybody in the room to review operational metrics, or to review the business metrics. The database teams may get together on a Tuesday morning to review everything that is going on in their services, and they will show up on Wednesday morning at the big AWS meeting where, now that we have 175 services not every service presents anymore, but a die is being rolled.

KILLALEA: Actually I believe that a wheel is being spun.

VOGELS: Yes. So, you need to be ready to talk about your operational results of the past week. An important part of that is there are senior people in the room, and there are junior folks who have just launched their first service. A lot of learning goes on in those two hours in that room that is probably the highest-value learning I've ever seen. The same goes for the business meeting, whether it's Andy [Jassy, CEO of AWS] in the room, or Jeff [Bezos, CEO of Amazon] or [Jeff] Wilke [CEO of Amazon Worldwide Consumer]; there's a lot of learning going on at a business level. Why did S3 delete this much data? Well, it turns out that this file-storage service that serves to others deletes only once a month, and prior to that they mark all of their objects (for deletion).

You need to know; you need to understand; and you need to talk to others in the room and share this knowledge. These operational review and business review meetings have become extremely valuable as an educational tool where the most senior engineers can shine in showing this is how you build and operate a scalable service.

KILLALEA: Fourteen years is such a long time in the life of a large-scale service with a continuously evolving architecture. Are there universal lessons that you could share for service owners who are much earlier in their journey? On S3's journey from 8 to 262 services over 14 years, any other learnings that would benefit our readers?

VOGELS: As always, security is number one. Otherwise, you have no business. Whatever you build, with whatever you architect, you should start with security. You cannot have a customer-facing service, or even an internally facing service,

without making that your number-one priority.

Finally, I have advice that is twofold. There's the Well-Architected Framework, where basically over five different pillars we've collected 10 or 15 years of knowledge from our customers as to what are the best practices.³ The Well-Architected Framework deals with operations, security, reliability, performance, and cost. For each of those pillars, you get 100 questions that you should be able to answer yourself for whatever you're building. For example, "Are you planning to do key rotation?" Originally our solutions architects would do this review for you, but with millions of customers, that doesn't really scale. We've built a tool for our customers so that they can do it in the console. Not only that, but they can also see across multiple of their projects what may be common deficiencies in each and every one of those projects. If you're not doing key rotation in any of them, maybe you need to put a policy in place or educate your engineers.

Second, and this is much more developer-oriented, is the Amazon Builders' Library.¹ That's a set of documents about how we built stuff at Amazon. One of the most important ones is cell-based architecture: How do you partition your service so that the blast radius is as minimal as possible? How do you do load shedding? All of these things that we struggled with at Amazon the Retailer—and came up with good solutions for—we now make available for everybody to look at.

KILLALEA: Thank you Werner; it's been wonderful to catch up with you.

VOGELS: Tom, it's been a pleasure talking to you. □

References

1. Amazon Builder's Library; <https://aws.amazon.com/builders-library/>.
2. Amazon Press Center. Amazon Web Services launches, 2006; <https://press.aboutamazon.com/news-releases/news-release-details/amazon-web-services-launches-amazon-s3-simple-storage-service>.
3. AWS Well-Architected; <https://aws.amazon.com/architecture/well-architected/>.
4. DeCandia, G., et al. Dynamo: Amazon's highly available key-value store. In *Proceedings of the 21st Annual Symp. Operating Systems Principles*. (Oct. 2007) 205–220; <https://dl.acm.org/doi/10.1145/1294261.1294281>.
5. Dixon, J. Pentaho, Hadoop, and data lakes, 2010; <https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/>.
6. Gray, J. A conversation with Werner Vogels. *acmqueue* 4, 4 (2006); <https://queue.acm.org/detail.cfm?id=1142065>.

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

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

From thingamabobs to rockets, 3D printing takes many forms.

BY JESSIE FRAZELLE

Out-of-This-World Additive Manufacturing

more online

A version of this article with embedded informational links is available at <https://queue.acm.org/detail.cfm?id=3430113>

POPULAR CULTURE USES the term *3D printing* as a synonym for *additive manufacturing* processes. In 2010, the American Society for Testing and Materials group ASTM F42—Additive Manufacturing—came up with a set of standards to classify additive

manufacturing processes into seven categories.

Each process uses different materials and machine technology, which affects the use cases and applications, as well as the economics. I went down a rabbit hole researching the various processes in my hunt to buy the best 3D printer. You can read my reviews on my blog. In this article, I will share a bit of what I learned about each process, as well as some of the more interesting use cases I found along the way.

Additive manufacturing has a variety of use cases ranging from thingamabobs to jewelry to metal parts for complex systems to even immense, exciting things like building a boat⁷ or rockets to go into space. Yes, you read that right, both Relativity Space

and Launcher Space are using additive manufacturing to build rockets that launch satellites (or other cargo) into space. Relativity Space even built its own additive manufacturing machine, named Stargate, for this purpose, while Launcher partnered with Additive Manufacturing Customized Machines on its engine, E-2.

It's amazing to see the scale and variety of different use cases for additive manufacturing processes. By using additive manufacturing, companies and individuals can go from idea to creation faster than having to involve a third-party manufacturing partner. The differentiation in all the various processes enables choosing a process that works best with what you intend to create. Let's continue to dive into each process!

Material Extrusion

Material extrusion is the most commonly available and least expensive type of 3D printing technology. It represents the largest installed base of 3D printers globally. In material extrusion an object is built by melting and extruding a thermoplastic polymer filament in a predetermined path, layer by layer. Imagine building an object using only a tube of toothpaste. You would slowly build the walls of the object by putting layers of toothpaste on top of each other. Material extrusion works in a similar way.

The most common applications for material extrusion are electrical housings, form and fit testings, jigs and fixtures, and investment casting patterns. The technology commonly used for material extrusion is known as fused deposition modeling, or FDM.

Fused deposition modeling. FDM, also known as FFF (fused filament fabrication), works with a range of standard thermoplastic filaments, such as ABS (acrylonitrile butadiene styrene), PLA (polylactic acid), PET (polyethylene terephthalate), TPU (thermoplastic polyurethane), nylon, and their various blends.

FDM works in the following way:

1. First, a spool of thermoplastic filament is loaded into the printer.



Once the nozzle has heated to the correct temperature, the filament is fed to the extrusion head and into the nozzle, where it melts.

2. Then the extrusion head is connected to a three-axis system that allows it to move in the *X*, *Y*, and *Z* dimensions. The melted material is extruded in thin strands and deposited layer by layer in predetermined locations, where it cools and solidifies. The cooling process can be accelerated by using fans attached to the extrusion head if the device supports it.

3. Filling an area requires multiple passes, similar to coloring with a marker. When a layer is complete, the build platform moves down or the extrusion head moves up, depending on the device, and a new layer is deposited. This process is repeated until the object is completed.

This process tends to result in FDM objects having visible layer lines, unless smoothed, possibly showing inaccuracies around complex features. The toothpaste analogy holds true here, as it would have visible layer lines as well.

While FDM is traditionally used for plastics, Markforged uses a combination of FDM and MIM (metal injection molding) for its Metal X printers. These machines can print metal and carbon-fiber parts. They use two filament materials—a bound metal powder filament and a ceramic release material—to create the part in a material extrusion process. The part then gets washed to break down the polymer binding material. Finally, the part is transformed from a lightly bound metal powder part to a full metal part via *sintering*. Each step uses a different machine, and the parts require supports, structures that

hold the parts and provide strength and resistance against forces like gravity for the object being printed. This demonstrates that you are not tied to a given process but can use aspects of each of these processes to fulfill a given use case. Markforged claims its process is safer and more cost efficient than using a loose metal powder.

Vat Photopolymerization

Photopolymerization is a common approach used by additive technologies to build an object one layer at a time. It occurs when a photopolymer resin is exposed to light of a specific wavelength, causing a chemical reaction that makes it turn solid. *Vat* refers to inducing this chemical reaction repeatedly in a vat to create a solid object.

Vat polymerization processes are excellent at producing objects with fine

details and smooth surface finishes. This makes them ideal for jewelry, low-run injection molding, dental applications, and medical applications such as hearing aids. The main limitation of vat polymerization is the brittleness of the produced objects. For this reason it is not suitable for mechanical parts.

Stereolithography. SLA was one of the world's first 3D printing technologies, invented by Charles Hull in 1984.³ SLA 3D printers use a laser to cure liquid resin into hardened plastic.


SLA machines have two main setups: top-down and bottom-up. These refer to the orientation of the laser and the part as it is being printed. Each approach has pros and cons, depending on the use case.

In a top-down setup, the laser source is above the tank and the part is built facing up. The build platform begins at the very top of the resin vat and moves downward after every layer. A top-down machine can handle very large build sizes and is faster than bottom-up machines, but it costs more and requires a specialist to operate. Also keep in mind that changing material in a top-down orientation printer requires emptying the whole tank, which can be time consuming and inefficient.


In a bottom-up machine, the light source comes from beneath the resin tank and the part is built facing upside down. The tank has a transparent bottom with a silicone coating that allows the light of the laser to pass through but prevents the cured resin from sticking to it. After every layer, the cured resin is detached from the bottom of the tank, as the build platform moves upward. This is known as the peeling step. A bottom-up machine is lower cost and more widely available but has a smaller build size and material range than a top-down setup. Bottom-up also requires more post-processing, a result of the extensive use of supports.⁸

The SLA process follows these steps:

1. First, a liquid photopolymer is filled into a vat or tank.
2. A concentrated beam of ultraviolet light or a laser is focused onto the surface of the vat or tank. The beam or laser creates each layer of the desired 3D object using cross-linking or by degrading the polymer at a specific location. This step is repeated layer by layer until the 3D object is built to completion.



It's amazing to see the scale and variety and different use cases for additive manufacturing processes.



An SLA object has high resolution and accuracy, clear details, and a smooth surface finish. SLA is quite versatile and can be applied to many different use cases since photopolymer resin formulations have a wide range of optical, mechanical, and thermal properties to match those of standard, engineering, and industrial thermoplastics.

Formlabs uses a bottom-up orientation for its 3D printers. It is common for desktop 3D printers to take this approach.

Direct light processing. DLP is nearly identical to SLA, except it uses a digital light projector screen to flash a single image of each layer all at once. Each layer is composed of square pixels, called *voxels*, since the projector is a digital screen. In a way, it is similar to an eight-bit ancestor of SLA in the same way that eight-bit drawings have more defined individual square pixels. Since each layer is exposed all at once, DLP can have faster print times compared with SLA, which solidifies a layer in cross sections.⁴

Continuous direct light processing. CDLP, also known as CLIP (continuous liquid interface production), produces objects in the same way as DLP but relies on the *continuous* motion of the build plate on the *Z*-axis. This results in faster build times because the printer is not required to stop and separate the part from the build plate after each layer is produced.

Powder Bed Fusion

PBF technologies produce a solid part using a thermal source that induces fusion, sintering, or melting between the particles of a plastic or metal powder one layer at a time. Most PBF technologies have mechanisms for spreading and smoothing thin layers of powder as a part is constructed, resulting in the final component being encapsulated in powder after the build is completed. The most common applications are functional objects, complex ducting (hollow designs), and low-run parts production.

The main variations in PBF technologies come from different energy sources, such as lasers or electron beams, and the powders used in the process, such as plastics or metals. Polymer-based PBF technologies allow for

innovation in that there is no need for support structures. This makes creating objects with complex geometries easier.

Both metal and plastic PBF objects typically are strong and stiff, with mechanical properties that are comparable to, or sometimes even better than, the bulk material. There is a range of post-processing methods available that can give objects a very smooth finish. For this reason, PBF is often used to manufacture functional metal parts for applications in the aerospace, automotive, medical, and dental industries.

The limitations of PBF tend to be surface roughness and shrinkage or distortion during processing, as well as the challenges that arise from powder handling and disposal.

Selective laser sintering. SLS is the most common additive manufacturing technology for industrial applications. The technology originated in the late 1980s at the University of Texas at Austin.⁶ An SLS 3D printer uses a high-powered CO₂ laser to fuse small particles of polymer powder.

The SLS process follows these steps:

1. First, a bed is filled with powder.
2. The inside of the printer is then heated to near the powder's melting point. This allows the laser to effectively finish what the heat started and sinter, or coalesce, the powdered material to create a solid structure. This step is repeated, layer by layer, until the object is completed.
3. Finally, the object, still encased in loose powder, is cleaned with brushes and pressurized air.

In contrast to SLA and FDM, SLS does not require an object to have support structures. This is because the unfused powder supports the part during printing. This makes SLS ideal for objects with complex geometries, including interior features, undercuts, and negative features. Parts produced with SLS printing typically have excellent mechanical characteristics, meaning they are very strong. Objects with thin walls cannot be printed because there is a minimum 1mm limitation, and thin walls in large models may warp after cooling down.

The most common material for selective laser sintering is polyamide (nylon), a popular engineering thermoplastic with great mechanical properties. Nylon is lightweight, strong, and

flexible, as well as stable against impact, chemicals, heat, UV light, water, and dirt. Alumide, a blend of gray aluminum powder and polyamide, and rubberlike materials can also be used.

The combination of low cost per part, high productivity, and established materials make SLS a popular choice among engineers for functional prototyping and a cost-effective alternative to injection molding for limited-run or bridge manufacturing.

Selective laser melting and direct metal laser sintering. Both SLM and DMLS produce objects via a method similar to SLS. Unlike SLS, however, SLM and DMLS are used in the production of metal parts. SLM fully melts the powder, while DMLS heats the powder to near melting temperatures until it chemically fuses. In practice, SLM and DMLS are functionally the same.¹

Unlike SLS, SLM and DMLS require support structures to compensate for the high residual stresses generated during the build process. Support structures help to limit the possibility of warping and distortion. DMLS is the most well-established metal additive manufacturing process and has the largest installed base.

Electron beam melting. EBM uses a high-energy beam rather than a laser to induce fusion between particles of metal powder. A focused electron beam scans across a thin layer of powder, which causes localized melting and solidification over a specific cross-section. An advantage of electron beam systems is that they produce less residual stress in objects, meaning there is less need for support structures. EBM also uses less energy and can produce layers quicker than SLM and DMLS. The minimum feature size, powder particle size, layer thickness, and surface finish, however, are typically lower quality than SLM and DMLS. EBM requires the objects to be produced in a vacuum, and the process can be used only with conductive material.²

Multijet fusion. MJF is essentially a combination of the SLS and material-jetting technologies. A carriage with nozzles, similar to those used in inkjet printers, passes over the print area, depositing a fusing agent on a thin layer of plastic powder. Simultaneously, a detailing agent that inhibits sintering is printed near the edge

of the part. A high-power infrared radiation energy source then passes over the build bed and sinters the areas where the fusing agent was dispensed, while leaving the rest of the powder untouched. The process repeats until the object is completed.

Material Jetting

Of all the additive manufacturing processes, material jetting is most comparable to the inkjet printing process. In the same way that an inkjet printer places ink layer by layer onto a piece of paper, material jetting deposits material onto the build surface. The layer is then cured or hardened using ultraviolet light. This is repeated layer by layer until the object is completed. Since the material is deposited in drops, the materials are limited to photopolymers, metals, or waxes that cure or harden when exposed to UV light or elevated temperatures.

Material jetting is ideal for realistic prototypes, providing excellent detail, high accuracy, and smooth surface finish. Material jetting allows a designer to use multiple colors and multiple materials in a single run. This makes the process great for low-run injection molds and medical models. It also allows support structures to be printed from a dissolvable material that is easily removed after building. The main drawbacks of material-jetting technologies are the high cost and brittle mechanical properties of the UV-activated photopolymers.

Nanoparticle jetting. NPJ is a process in which a liquid containing metal nanoparticles or support nanoparticles is loaded into the printer via a cartridge. The liquid is then jetted, similar to an inkjet printer, onto a build tray through thousands of nozzles in extremely thin layers of droplets. High temperatures inside the building chamber cause the liquid to evaporate, leaving behind a metal object.

Drop-on-demand. DOD material jetting printers have two print jets: one to deposit the build material, typically a wax-like liquid, and another for a dissolvable support material. Similar to material extrusion, a DOD printer follows a predetermined path and deposits material in a pointwise fashion to build layers of an object. This machine also employs a fly-cutter, a single-point

more online

A version of this article with embedded informational links is available at <https://queue.acm.org/detail.cfm?id=3430113>

cutting tool, that skims the build area after each layer to ensure a perfectly flat surface before printing the next layer. DOD technology is typically used to produce waxlike patterns for lost-wax casting (used to duplicate a metal sculpture cast from an original sculpture) and mold-making applications.⁹

Binder Jetting

A binder jetting process, also referred to as 3DP, uses two materials: a powder and a binder. The binder, which is typically a liquid, acts as the adhesive for the powder. A print head, much like that in an inkjet printer, moves horizontally across the *x* and *y* axes to deposit alternating layers of the powder and the binder. The platform holding the bed of powder, which the object is printed on, lowers as each layer is printed. This is repeated until the object is completed. Like SLS, the object does not need support structures since the powder bed acts as support. The powder materials can be either ceramic-based such as glass or gypsum, or metal such as stainless steel.

Ceramic-based binder jetting, which uses a ceramic powder as the material, is best for aesthetic applications that need intricate designs such as architectural models, packaging, molds for sand casting, and ergonomic verification. It is not intended for functional prototypes, as the objects created are quite brittle.

Metal binder jetting, which uses a metal powder as the material, is well suited for functional components and more cost effective than SLM or DMLS metal parts. The downside, however, is the metal parts have poorer mechanical properties.

The same people who created binder jetting also created Desktop Metal, a 3D printer system using this technology.

Direct Energy Deposition

DED creates objects by melting powder material as it is deposited, similar to material extrusion. It is predominantly used with metal powders or wire and is often referred to as metal deposition since it is exclusive to metals. DED relies on dense support structures, which are not ideal for creating parts from

scratch. This makes it best suited for repairing or adding material to existing objects such as turbine blades.

Metal direct energy is what Relativity Space uses to print its rocket parts. Because of the size of the parts it needs to build, it uses a custom machine.

Laser powder forming. Laser powder forming is also known by its proprietary name, LENS (Laser Engineered Net Shaping), developed at Sandia National Labs. The process uses a deposition head that consists of a laser head, powder-dispensing nozzles, and inert gas tubing. The deposition head melts the powder as it is ejected from the nozzles to build an object layer by layer. The laser creates a melt pool on the build area, and powder is sprayed into the pool, where it is melted and then solidified.

Electron beam additive manufacturing. EBAM uses an electron beam to create metal objects by welding together metal powder or wire. In contrast to laser powder forming, which uses a laser, electron beams are more efficient and operate under a vacuum that was originally designed for use in space.⁵

Sheet Lamination

Sheet lamination processes include LOM (laminated object manufacturing) and UAM (ultrasonic additive manufacturing). You might be familiar with LOM—it is basically the same technology used by the laminator you may have used as a child. To laminate a piece of paper, you place the paper in a laminator pouch composed of two types of plastic: PET (polyethylene terephthalate) on the outer layer and EVA (ethylene-vinyl acetate) on the inner layer. A heated roller then adheres the two sides of the pouch together so the paper is fully encased in plastic. The same basic process is used to build objects.


UAM, on the other hand, builds metal objects by fusing and stacking metal strips, sheets, or ribbons. The layers are bound together using ultrasonic welding. The process is done on a machine able to CNC (computer numerical control) mill the workpiece as the layers are built. The process requires removal of the unbound metal, often during the welding process. UAM uses metals such as aluminum, copper, stainless steel, and titanium. The process can

bond different materials, build at a fast rate, and make large objects practically, while requiring relatively little energy since the metal is not melted.

The Future

The ability to go from a digital file to a physical object rapidly with many different materials allows you to create something you could only imagine in your wildest dreams. Additive manufacturing enables individuals and companies to create without having to involve third-party manufacturers, enabling them to go from idea to prototype much faster. Additive manufacturing is even being used to go to space. We have just begun to imagine all the applications for additive manufacturing. Stay tuned to see what people build in the future.

Acknowledgments

Thank you to Jordan Noone (@thejordannoone) and Michael Fogleman (@FogleBird) for reading an initial draft of this article. 

References

1. Jones, G. Direct metal laser sintering (DMLS)—simply explained. All3DP, 2019; <https://all3dp.com/2/direct-metal-laser-sintering-dmls-simply-explained/>.
2. Murr, L.E., Gaytan, S.M. Advances in additive manufacturing and tooling. *Comprehensive Materials Processing*. S. Hashmi, C.J. Van Tyne, G.F. Batalha, and B. Yilbas, Eds. Elsevier, 2014, 135–161; <https://www.sciencedirect.com/topics/chemistry/electron-beam-melting>.
3. Norman, J. Chuck Hull invents stereolithography or 3D printing and produces the first commercial 3D printer. History of Information; <https://www.historyofinformation.com/detail.php?id=3864>.
4. Redwood, B. Additive manufacturing technologies: an overview. 3D Hubs; <https://www.3dhubs.com/knowledge-base/additive-manufacturing-technologies-overview/>.
5. Shuhe, C., Gach, S., Senger, A., Haoyu, Z. A new 3D printing method based on non-vacuum electron beam technology. *J. Physics: Conference Series* 1074:012017, 2018; https://www.researchgate.net/publication/328169730_A_new_3D_printing_method_based_on_non-vacuum_electron_beam_technology.
6. University of Texas at Austin. Selective laser sintering, birth of an industry, 2012; <https://www.me.utexas.edu/news/news/selective-laser-sintering-birth-of-an-industry>.
7. V., C. University of Maine creates the world's largest 3D printed boat. 3D Natives, 2019; <https://www.3dnatives.com/en/3d-printed-boat-university-of-maine-161020195/#!>.
8. Varotsis, A.B. Introduction to SLA 3D printing. 3D Hubs; <https://www.3dhubs.com/knowledge-base/introduction-sla-3d-printing/>.
9. Zhang, L. Characteristics of drop-on-demand droplet jetting with effect of altered geometry of printhead nozzle. *Sensors and Actuators A: Physical*, 2019, 298; <https://www.sciencedirect.com/science/article/abs/pii/S0924424719312701>.

Jessie Frazelle is the cofounder and chief product officer of the Oxide Computer Company. Before that, she worked on various parts of Linux, including containers, and the Go programming language.

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

Providing Sound Foundations for Cryptography

On the work of Shafi Goldwasser and Silvio Micali

Cryptography is concerned with the construction of schemes that withstand any abuse. A cryptographic scheme is constructed so as to maintain a desired functionality, even under malicious attempts aimed at making it deviate from its prescribed behavior. The design of cryptographic systems must be based on firm foundations, whereas ad hoc approaches and heuristics are a very dangerous way to go. These foundations were developed mostly in the 1980s, in works that are all co-authored by Shafi Goldwasser and/or Silvio Micali. These works have transformed cryptography from an engineering discipline, lacking sound theoretical foundations, into a scientific field possessing a well-founded theory, which influences practice as well as contributes to other areas of theoretical computer science.

This book celebrates these works, which were the basis for bestowing the 2012 A.M. Turing Award upon Shafi Goldwasser and Silvio Micali. A significant portion of this book reproduces some of these works, and another portion consists of scientific perspectives by some of their former students. The highlight of the book is provided by a few chapters that allow the readers to meet Shafi and Silvio in person. These include interviews with them, their biographies and their Turing Award lectures.

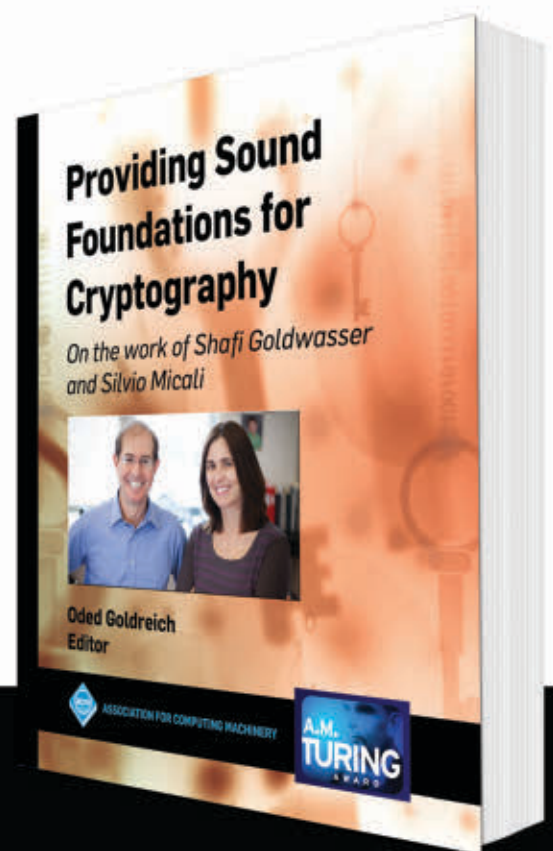
Oded Goldreich, Editor

ISBN: 978-1-4503-7267-1

DOI: 10.1145/3335741

<http://books.acm.org>

<http://store.morganclaypool.com/acm>



ACM BOOKS
Collection II

DOI:10.1145/3430936

Technological and economic forces are now pushing computing away from being general purpose and toward specialization.

BY NEIL C. THOMPSON AND SVENJA SPANUTH

The Decline of Computers as a General Purpose Technology

PERHAPS IN NO other technology has there been so many decades of large year-over-year improvements as in computing. It is estimated that a third of all productivity increases in the U.S. since 1974 have come from information technology,^{a,4} making it one of the largest contributors to national prosperity.

The rise of computers is due to technical successes, but also to the economics forces that financed them. Bresnahan and Trajtenberg³ coined the term *general purpose technology* (GPT) for products, like computers, that have broad technical applicability *and* where product improvement and market growth

^a Their analysis excludes the farming sector.

could fuel each other for many decades. But, they also predicted that GPTs could run into challenges at the end of their life cycle: as progress slows, other technologies can displace the GPT in particular niches and undermine this economically reinforcing cycle. We are observing such a transition today as improvements in central processing units (CPUs) slow, and so applications move to *specialized processors*, for example, graphics processing units (GPUs), which can do fewer things than traditional universal processors, but perform those functions better. Many high profile applications are already following this trend, including deep learning (a form of machine learning) and Bitcoin mining.

With this background, we can now be more precise about our thesis: “The Decline of Computers as a General Purpose Technology.” We do *not* mean that computers, taken together, will lose technical abilities and thus ‘forget’ how to do some calculations. We *do* mean that the economic cycle that has led to the usage of a common computing platform, underpinned by rapidly improving universal processors, is giving way to a fragmentary cycle, where economics push users toward divergent computing platforms driven by special purpose processors.

» key insights

- **Moore’s Law was driven by technical achievements and a “general purpose technology” (GPT) economic cycle where market growth and investments in technical progress reinforced each other. These created strong economic incentives for users to standardize to fast-improving CPUs, rather than designing their own specialized processors.**
- **Today, the GPT cycle is unwinding, resulting in less market growth and slower technical progress.**
- **As CPU improvement slows, economic incentives will push users toward specialized processors, which threatens to fragment computing. In such a computing landscape, some users will be in the ‘fast lane,’ benefiting from customized hardware, and others will be left in the ‘slow lane,’ stuck on CPUs whose progress fades.**

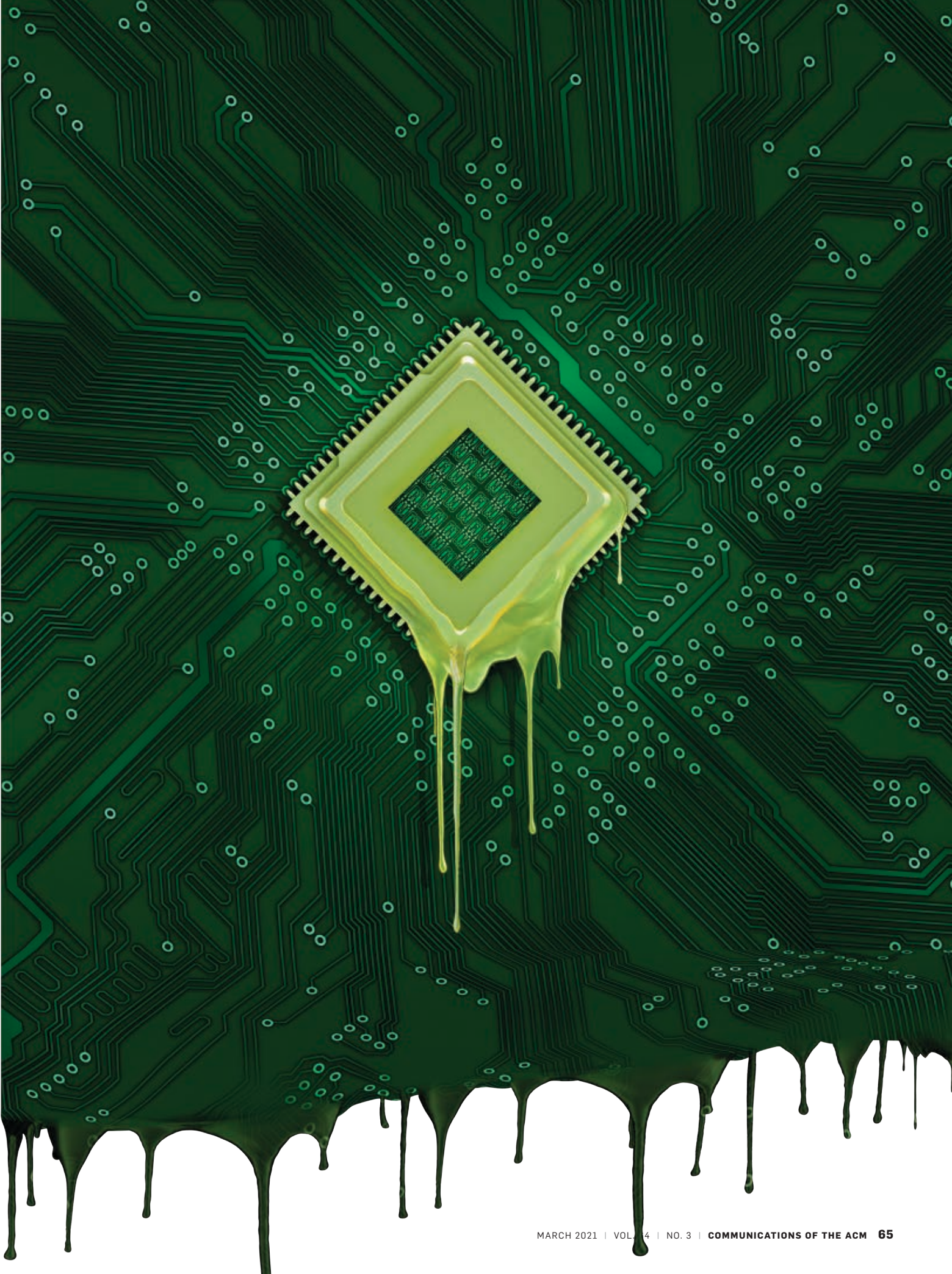
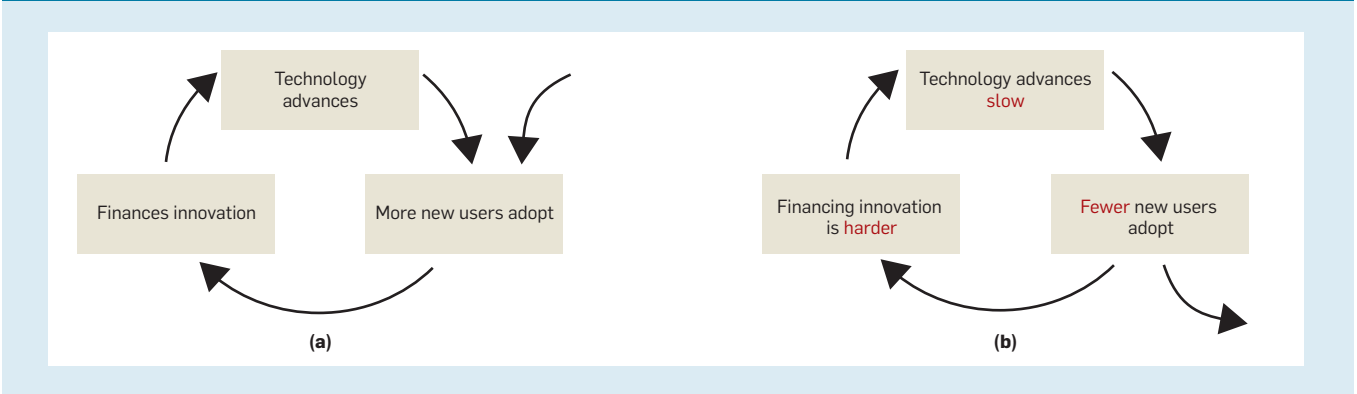


Figure 1. The historical virtuous cycle of universal processors (a) is turning into a fragmentation cycle (b).



This fragmentation means that parts of computing will progress at different rates. This will be fine for applications that move in the ‘fast lane,’ where improvements continue to be rapid, but bad for applications that no longer get to benefit from field-leaders pushing computing forward, and are thus consigned to a ‘slow lane’ of computing improvements. This transition may also slow the overall pace of computer improvement, jeopardizing this important source of economic prosperity.

Universal and Specialized Computing

Early days—from specialized to universal. Early electronics were not universal computers that could perform many different calculations, but dedicated pieces of equipment, such as radios or televisions, designed to do one task, and only one task. This specialized approach has advantages: design complexity is manageable and the processor is efficient, working faster and using less power. But specialized processors are also ‘narrower,’ in that they can be used by fewer applications.

Early electronic computers,^b even

b In this article, the term “computer” describes both, devices with solely a universal processor and those that also contain specialized functionality.

those designed to be ‘universal,’ were in practice tailored for specific algorithms and were difficult to adapt for others. For example, although the 1946 ENIAC was a theoretically universal computer, it was primarily used to compute artillery range tables. If even a slightly different calculation was needed, the computer would have to be manually re-wired to implement a new hardware design. The key to resolving this problem was a new computer architecture that could store instructions.¹⁰ This architecture made the computer more flexible, making it possible to execute many different algorithms on universal hardware, rather than on specialized hardware. This ‘von Neumann architecture’ has been so successful that it continues to be the basis of virtually all universal processors today.

The ascent of universal processors. Many technologies, when they are introduced into the market, experience a virtuous reinforcing cycle that helps them develop (Figure 1a). Early adopters buy the product, which finances investment to make the product better. As the product improves, more consumers buy it, which finances the next round of progress, and so on. For many products, this cycle winds down in the short-to-medium term as

product improvement becomes too difficult or market growth stagnates.

GPTs are defined by the ability to continue benefiting from this virtuous economic cycle as they grow—as universal processors have for decades. The market has grown from a few high-value applications in the military, space, and so on, to more than two billion PCs in use worldwide.³⁸ This market growth has fueled ever-greater investments to improve processors. For example, Intel has spent \$183 billion on R&D and new fabrication facilities over the last decade.^c This has paid enormous dividends: by one estimate processor performance has improved about 400,000x since 1971.⁸

The alternative: Specialized processors. A universal processor must be able to do many different calculations well. This leads to design compromises that make many calculations fast, but none optimal. The performance penalty from this compromise is high for applications well suited to specialization, that is those where:

- ▶ substantial numbers of calculations can be parallelized
- ▶ the computations to be done are stable and arrive at regular intervals (‘regularity’)
- ▶ relatively few memory accesses are needed for a given amount of computation (‘locality’)
- ▶ calculations can be done with fewer significant digits of precision.¹⁵

In each of these cases, specialized processors (for example, Application-specific Integrated Circuits (ASICs)) or specialized parts of heterogeneous chips (for example, I.P. blocks) can

c Calculated as 2008–2017 R&D and additions to PPE spending.

Technical specifications of a CPU compared to a GPU.

Processor	Model	Calculations in parallel ⁱ	Speed	Memory Bandwidth	Access to Level 1 Cache
CPU	Intel Xeon E5-2690v4	28	2.6–3.5 GHz	76.8 GB/s	5–12 clock cycles ⁱ
GPU	NVIDIA P100	3,584	1.1 GHz	732 GB/s	80 clock cycles

ⁱ Data from Intel and NVIDIA data sheets, ‘Access to L1 Cache’ from Giles^{59ppx}
ⁱⁱ Approximated by number of threads for CPU and number of CUDA cores for GPU.

perform better because custom hardware can be tailored to the calculation.²⁴

The extent to which specialization leads to changes in processor design can be seen in the comparison of a typical CPU—the dominant universal processor—and a typical GPU—the most-common type of specialized processor (see the accompanying table).

The GPU runs slower, at about a third of the CPU’s frequency, but in each clock cycle it can perform ~100x more calculations in parallel than the CPU. This makes it much quicker than a CPU for tasks with lots of parallelism, but slower for those with little parallelism.^d

GPUs often have 5x–10x more memory bandwidth (determining how much data can be moved at once), but with much longer lags in accessing that data (at least 6x as many clock cycles from the closest memory). This makes GPUs better at predictable calculations (where the data needed from memory can be anticipated and brought to the processor at the right time) and worse at unpredictable ones.

For applications that are well-matched to specialized hardware (and where programming models, for example CUDA, are available to harness that hardware), the gains in performance can be substantial. For example, in 2017, NVIDIA, the leading manufacturer of GPUs, estimated that Deep Learning (AlexNet with Caffe) got a speed-up of 35x+ from being run on a GPU instead of a CPU.²⁷ Today, this speed-up is even greater.²⁶

Another important benefit of specialized processors^e is that they use less power to do the same calculation. This is particularly valuable for applications limited by battery life (cell phones, Internet-of-things devices), and those that do computation at enormous scales (cloud computing/datacenters, supercomputing).

As of 2019, 9 out of the top 10 most power efficient supercomputers were using NVIDIA GPUs.³⁷

d Of course, many tasks will have multiple parts, some parallelizable and some not. In which case, speed-ups will be constrained by Amdahl’s Law.

e For brevity we will use the term “specialized processors” throughout this article to refer both to stand-alone processors as well as specialized functionality on heterogeneous chips (for example, I.P. blocks)

Specialized processors also have important drawbacks: they can only run a limited range of programs, are hard to program, and often require a universal processor running an operating system to control (one or more of) them. Designing and creating specialized hardware can also be expensive. For universal processors, their fixed costs (also called non-recurring engineering costs (NRE)) are distributed over a large number of chips. In contrast, specialized processors often have much smaller markets, and thus higher per-chip fixed costs. To make this more concrete, the overall cost to manufacture a chip with specialized processors using leading-edge technology is about \$80 million^f (as of 2018). Using an older generation of technology can bring this cost down to about \$30 million.²³

Despite the advantages that specialized processors have, their disadvantages were important enough that there was little adoption (except for GPUs) in the past decades. The adoption that did happen was in areas where the performance improvement was inordinately valuable, including military applications, gaming and cryptocurrency mining. But this is starting to change.

The state of specialized processors today. All the major computing platforms, PCs, mobile, Internet-of-things

f This true for the 16/14nm node size. Lithography cost are by far the biggest cost component of the manufacturing NRE.²¹ Other costs include labor and design tools, as well as IP licensing.

(IoT), and cloud/supercomputing, are becoming more specialized. Of these, PCs remain the most universal. In contrast, energy efficiency is more important in mobile and IoT because of battery life, and thus, much of the circuitry on a smartphone chip,³⁴ and sensors, such as RFID-tags, use specialized processors.^{5,7}

Cloud/supercomputing has also become more specialized. For example, 2018 was the first time that new additions to the biggest 500 supercomputers derived more performance from specialized processors than from universal processors.¹¹

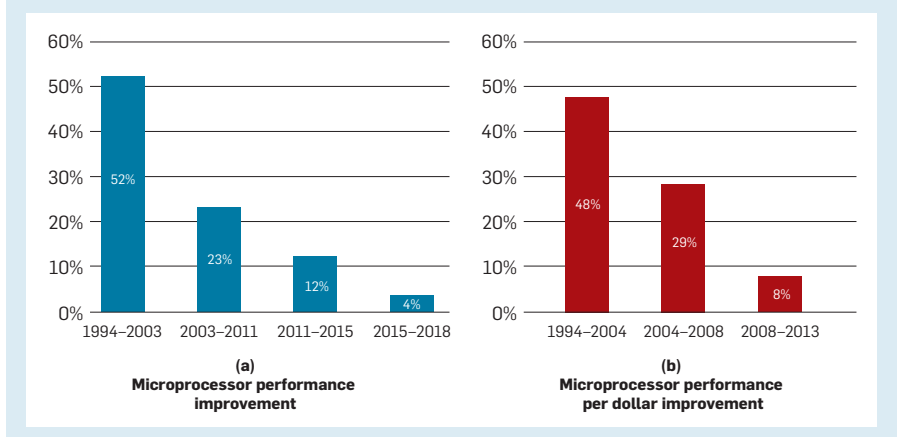
Industry experts at the International Technology Roadmap for Semiconductors (ITRS), the group which coordinated the technology improvements needed to keep Moore’s Law going, implicitly endorsed this shift toward specialization in their final report. They acknowledged the traditional one-solution-fits-all approach of shrinking transistors should no longer determine design requirements and instead these should be tailored to specific applications.¹⁶

The next section explores the effect that the movement of all of the major computing platforms toward specialized processors will have on the economics of producing universal processors.

The Fragmentation of a General Purpose Technology

The virtuous cycle that underpins GPTs comes from a mutually reinforcing set of technical and economic forces. Unfortunately, this mutual reinforcement also applies in the reverse direc-

Figure 2. Rate of improvement in microprocessors, as measured by (a) Annual performance improvement on the SPECint benchmark,^{7appx} and (b) Annual quality-adjusted price decline.^{1appx}



tion: if improvements slow in one part of the cycle, so will improvements in other parts of the cycle. We call this counterpoint a ‘fragmenting cycle’ because it has the potential to fragment computing into a set of loosely-related siloes that advance at different rates.

As Figure 1(b) shows, the fragmenting cycle has three parts:

- ▶ Technology advances slow
- ▶ Fewer new users adopt
- ▶ Financing innovation is more difficult

The intuition behind this cycle is straightforward: if technology advances slow, then fewer new users adopt. But, without the market growth provided by those users, the rising costs needed to improve the technology can become prohibitive, slowing advances. And thus each part of this synergistic reaction further reinforces the fragmentation.

Here, we describe the state of each of these three parts of the cycle for computing and show that fragmentation has already begun.

Technology advancements slow. To measure the rate of improvement of processors we consider two key metrics: *performance*^g and *performance-per-dollar*. Historically, both of these metrics improved rapidly, largely because miniaturizing transistors led to greater density of transistors per chip (Moore’s Law) and to faster transistor switching speeds (via Dennard Scaling).²⁴ Unfortunately, Dennard Scaling ended in 2004/2005 because of technical challenges and Moore’s Law is coming to an end as manufacturers hit the physical limits of what existing materials and designs can do,³³ and these limits take ever more effort to overcome.² The loss of the benefits of miniaturization can be seen vividly in the slowdown of improvements to performance and performance-per-dollar.

Figure 2(a), based Hennessy and Patterson’s characterization of progress in SPECInt, as well as Figure 2(b) based on the U.S. Bureau of Labor Statistics’ producer-price index, show how dramatic the slowdown in performance improve-



GPTs are defined by the ability to continue benefiting from this virtuous economic cycle as they grow—as universal processors have for decades.



ment in universal computers has been. To put these rates into perspective, if performance per dollar improves at 48% per year, then in 10 years it improves 50x. In contrast, if it only improves at 8% per year, then in 10 years it is only 2x better.

Fewer new users adopt. As the pace of improvement in universal processors slows, fewer programs with new functionality will be created, and thus customers will have less incentive to replace their computing devices. Intel CEO Krzanich confirmed this in 2016, saying that the replacement rate of PCs had risen from every four years to every 5–6 years.²² Sometimes, customers even skip multiple generations of processor improvement before it is worth updating.²⁸ This is also true on other platforms, for example U.S. smartphones were upgraded on average every 23 months in 2014, but by 2018 this had lengthened to 31 months.²⁵

The movement of users from universal to specialized processors is central to our argument about the fragmentation of computing, and hence we discuss it in detail. Consider a user that could use either a universal processor or a specialized one, but who wants the one that will provide the best performance at the lowest cost.^h Figures 3(a) and 3(b) present the intuition for our analysis. Each panel shows the performance over time of universal and specialized processors, but with different rates at which the universal processor improves. In all cases, we assume that the time, T , is chosen so the higher price of a specialized processor is exactly balanced out by the costs of a series of (improving) universal processors. This means that both curves are cost equivalent, and thus superior performance also implies superior performance-per-dollar. This is also why we depict the specialized processor as having constant performance over this period. (At the point where the specialized processor would be upgraded, it too would get the benefit of whatever process improvement had benefited the universal processor and the user would again repeat this same decision process.)

A specialized processor is more at-

^g While we have in mind a measure of performance based on computational power/speed, this model is actually more general and could refer to other characteristics (for example, energy efficiency).

^h Computing at larger scales (including the massive parallelism of current deep learning models) are scaled-up versions of this same problem, and the logic of our analysis (and thus our results) also carry over to them.

tractive if it provides a larger initial gain in performance. But, it also becomes more attractive if universal processor improvements go from a rapid rate, as in panel (a), to a slower one, as in panel (b). We model this formally by considering which of two time paths provides more benefit. That is, a specialized processor is more attractive if

$$\int_0^T P_s dt \geq \int_0^T P_{u,t_0} e^{rt} dt$$

Where universal and specialized processors deliver performanceⁱ P_u , and P_s , over time T , while the universal processor improves at r .^j We present our full derivation of this model in the online appendix (<https://doi.org/10.1145/3430936>). That derivation allows us to numerically estimate the volume needed for the advantages of specialization to outweigh the higher costs (shown in Figure 3(c) for a slowdown from 48% to 8% in the per-year improvement rate of CPUs).

Not surprisingly, specialized processors are more attractive when they provide larger speedups or when their costs can be amortized over larger volumes. These cutoffs for when specialization becomes attractive change, however, based on the pace of improvement of the universal processors. Importantly, this effect does not arise because we are assuming different rates of progress between specialized and universal processors overall—all processors are assumed to be able to use whatever is the cutting-edge fabrication technology of the moment. Instead, it arises because the higher per-unit NRE of specialized processors must be amortized and how well this compares to upgrading universal processors over that period.

A numerical example makes clear the

i Here we assume that the cost of the CPU running the OS and controlling the specialized processor(s) does not materially affect this calculation. Relaxing that assumption would not change our model but would require incorporating of these costs into the specialized processor parameter estimates.

j In practice, manufacturers do not update continuously, but in large steps when they release new designs. Users, however, may experience these jumps more continuously since they tend to constantly refresh some fraction of their computers. The continuous form is also more mathematically tractable.

importance of this change. At the peak of Moore’s Law, when improvements were 48% per year, even if specialized processors were 100x faster than universal ones, that is, $\frac{P_s}{P_u} = 100$ (a huge difference), then ~83,000 would need to be built for the investment to pay off. At the other extreme, if the performance benefit were only 2x, ~1,000,000 would need to be built to make specialization attractive. These results make it clear why, during the heyday of Moore’s Law, it was so difficult for specialized processors to break into the market.

However, if we repeat our processor choice calculations using an improvement rate of 8%, the rate from 2008–2013, these results change markedly: for applications with 100x speed-up, the number of processors needed falls from 83,000 to 15,000, and for those with 2x speed-up it drops from 1,000,000 to 81,000. Thus, after universal processor progress slows, many more applications became viable for

specialization.^k

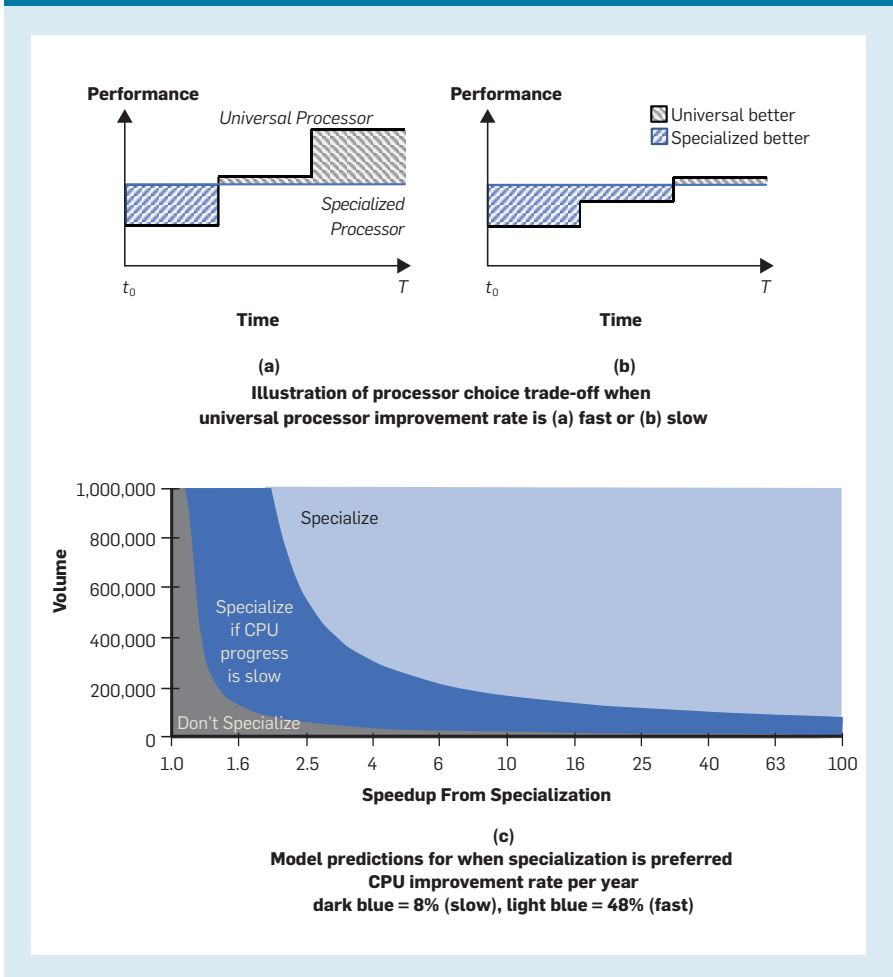
Financing innovation is harder. In 2017, the Semiconductor Industry Association estimated that the cost to build and equip a fabrication facility (‘fab’) for the next-generation of chips was roughly \$7 billion.³⁵ By “next-generation,” we mean the next miniaturization of chip components (or process ‘node’).

The costs invested in chip manufacturing facilities must be justified by the revenues that they produce. Perhaps as much as 30%^l of the industry’s \$343 billion annual revenue (2016) comes from cutting-edge chips. So

k In the online appendix (<https://doi.org/10.1145/3430936>) we also consider how these values change with code development costs.

l \$23 billion of Foundry revenue (TSMC and GlobalFoundries) can be attributed to leading-edge nodes.³⁶ Assuming the majority (90%) of Intel’s (\$54 billion) and Samsung’s (\$40 billion) total revenues¹² derives from leading-edge nodes, yields an upper bound of \$108 billion/\$343 billion ≈ 30%.

Figure 3. Optimal processor choice depends on the performance speed-up that the specialized processor provides, as well as the rate of improvement of the universal technology.



revenues are substantial, but costs are growing. In the past 25 years, the investment to build leading-edge fab (as shown in Figure 4a) rose 11% per year (!), driven overwhelmingly by lithography costs. Including process-development costs into this estimate further accelerates cost increases to 13% per year (as measured for 2001 to 2014 by Santhanam et al.³²). This is well known by chipmakers who quip about Moore’s “second law”: the cost of a chip fab doubles every four years.⁹

Historically, the implications of such a rapid increase in fixed cost on unit costs was only partially offset by strong overall semiconductor market growth (CAGR of 5% from 1996–

2016^{m,35}), which allowed semiconductor manufacturers to amortize fixed costs across greater volumes. The remainder of the large gap between fixed costs rising 13% annually and the market growing 5% annually, would be expected to lead to less-competitive players leaving the market and remaining players amortizing their fixed costs over a larger number of chips.

As Figure 4(b) shows, there has indeed been enormous consolidation in the industry, with fewer and fewer com-

panies producing leading-edge chips. From 2002/2003 to 2014/2015/2016, the number of semiconductor manufacturers with a leading-edge fab has fallen from 25 to just 4: Intel, Taiwan Semiconductor Manufacturing Company (TSMC), Samsung and GlobalFoundries). And GlobalFoundries recently announced that they would not pursue development of the next node.⁶

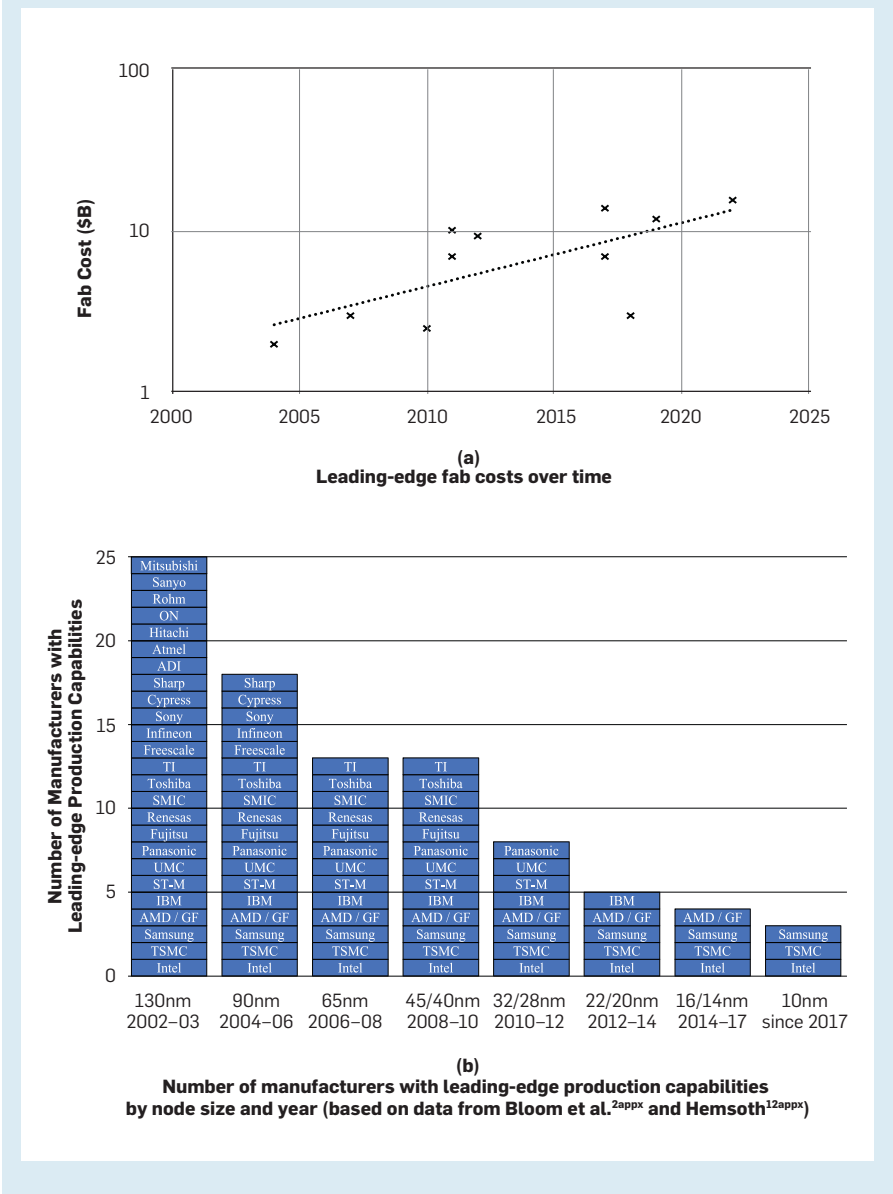
We find it very plausible this consolidation is caused by the worsening economics of rapidly rising fixed costs and only moderate market size growth. The extent to which market consolidation improves these economics can be seen through some back-of-the-envelope calculations. If the market were evenly partitioned amongst different companies, it would imply a growth in average market share from 4% = $\frac{100\%}{25}$ in 2002/2003 to 25% = $\frac{100\%}{4}$ in 2014/2015/2016. Expressed as a compound annual growth rate, this would be 14%. This means that producers could offset the worsening economics of fab construction through market growth and taking the market share of those exiting (13% < 5% + 14%).

In practice, the market was not evenly divided, Intel had dominant share. As a result, Intel would not have been able to offset fixed cost growth this way.ⁿ And indeed, over the past decade, the ratio of Intel’s fixed costs to its variable costs has risen from 60% to over 100%.^o This is particularly striking because in recent years Intel has slowed the pace of their release of new node sizes, which would be expected to *decrease* the pace at which they would need to make fixed costs investments.

The ability for market consolidation to offset fixed cost increases can only proceed for so long. If we project forward current trends, then by 2026 to 2032 (depending on market growth rates) leading-edge semiconductor manufacturing will only be able to support a single monopolist manufacturer, and yearly fixed costs to build a single new facility for each node size will be equal to yearly industry

m We implicitly assume that this is also the rate of growth for the leading-edge nodes. In practice it may be somewhat lower, which would only accentuate our point.

Figure 4. Deteriorating economics of chip manufacturing.



n Despite their rate of change being less favorable, Intel’s large market share meant that they started from a lower base, so they remained highly competitive.
o Calculated from Intel financial statements, with fixed costs as R&D + Property, Plant and Equipment, and variable costs as cost of goods sold.

revenues (see endnote for details^p). We make this point not to argue that in late 2020s this will be the reality, but precisely to argue that current trends *cannot* continue and that within only about 10 years(!) manufacturers will be forced to dramatically slow down the release of new technology nodes and find other ways to control costs, both of which will further slow progress on universal processors.


The fragmentation cycle. With each of the three parts of the fragmentation cycle already reinforcing each other, we expect to see more and more users facing meager improvements to universal processors and thus becoming interested in switching to specialized ones. For those with sufficient demand and computations well-suited to specialization (for example, deep learning), this will mean orders of magnitude improvement. For others, specialization will not be an option and they will remain on universal processors improving ever-more slowly.

Implications


Who will specialize. As shown in Figure 3(c), specialized processors will be adopted by those that get a large speedup from switching, and where enough processors would be demanded to justify fixed costs. Based on these criteria, it is perhaps not surprising that big tech companies have been amongst the first to invest in specialized processors, for example, Google,¹⁹ Microsoft,³¹ Baidu,¹⁴ and Alibaba.²⁹ Unlike the specialization with GPUs, which still benefited a broad range of applications, or those in cryptographic circuits, which are valuable to most users, we expect narrower specialization going forward because only small numbers of processors will be needed to make the economics attractive.

We also expect significant usage from those who were not the original designer of the specialized processor, but who re-design their algorithm to take advantage of new hardware, as deep learning users did with GPUs.

^p Assumes new facilities are needed every two years; 30% of market sales go to leading edge chips; and 13% annual increase in fixed costs. 2026: 0% market growth / 2032: 5% market growth. We (conservatively) assume all market demand can be met with a single facility. If more than that is needed, the date moves earlier.



It is expected the final benefits from miniaturization will come at a price premium, and are only likely to be paid for by important commercial applications.



Who gets left behind. Applications that do not move to specialized processors will likely fail to do so because they:

- ▶ Get little performance benefit,
- ▶ Are not a sufficiently large market to justify the upfront fixed costs, or
- ▶ Cannot coordinate their demand.

Earlier, we described four characteristics that make calculations amenable to speed-up using specialized processors. Absent these characteristics, there are only minimal performance gains, if any, to be had from specialization. An important example of this is databases. As one expert we interviewed told us: over the past decades, it has been clear that a specialized processor for databases could be very useful, but the calculations needed for databases are poorly-suited to being on a specialized processor.

The second group that will not get specialized processors are those where there is insufficient demand to justify the upfront fixed costs. As we derived with our model, a market of many thousands of processors are needed to justify specialization. This could impact those doing intensive computing on a small scale (for example, research scientists doing rare calculations) or those whose calculations change rapidly over time and thus whose demand disappears quickly.

A third group that is likely to get left behind are those where no individual user represents sufficient demand, and where coordination is difficult. For example, even if thousands of small users would collectively have enough demand, getting them to collectively contribute to producing a specialized processor would be prohibitively difficult. Cloud computing companies can play an important role in mitigating this effect by financing the creation of specialized processors and then renting these out.^q

Will technological improvement bail us out? To return us to a convergent cycle, where users switch back to universal processors, would require rapid improvement in performance and/or performance-per-dollar. But technological trends point in the opposite direction. For example on performance, it is

^q Already, Google provides TPUs on its cloud,¹³ and Amazon Web Services (and others) provide GPUs.¹

expected that the final benefits from miniaturization will come at a price premium, and are only likely to be paid for by important commercial applications. There is even a question whether all of the remaining, technically-feasible, miniaturization will be done. Gartner predicts that more will be done, with 5nm node sizes being produced at scale by 2026,¹⁸ and TSMC recently announced plans for a \$19.5B 3nm plant for 2022.¹⁷ But many of the interviewees that we contacted for this study doubt were skeptical about whether it would be worthwhile miniaturizing for much longer.

Might another technological improvement restore the pace of universal processor improvements? Certainly, there is a lot of discussion of such technologies: quantum computing, carbon nanotubes, optical computing. Unfortunately, experts expect that it will be at least another decade before industry could engineer a quantum computer that is broader and thus could potentially substitute for classical universal computers.³⁰ Other technologies that might hold broader promise will likely still need significantly more funding to develop and come to market.²⁰

Conclusion

Traditionally, the economics of computing were driven by the general purpose technology model where universal processors grew ever-better and market growth fuels rising investments to refine and improve them. For decades, this virtuous GPT cycle made computing one of the most important drivers of economic growth.

This article provides evidence that this GPT cycle is being replaced by a fragmenting cycle where these forces work to slow computing and divide users. We show each of the three parts of the fragmentation cycle are already underway: there has been a dramatic and ever-growing slowdown in the improvement rate of universal processors; the economic trade-off between buying universal and specialized processors has shifted dramatically toward specialized processors; and the rising fixed costs of building ever-better processors can no longer be covered by market growth rates.

Collectively, these findings make it clear that the economics of processors has changed dramatically, pushing computing into specialized domains

that are largely distinct and will provide fewer benefits to each other. Moreover, because this cycle is self-reinforcing, it will perpetuate itself, further fragmenting general purpose computing. As a result, more applications will split off and the rate of improvement of universal processors will further slow.

Our article thus highlights a crucial shift in the direction that economics is pushing computing, and poses a challenge to those who want to resist the fragmentation of computing. **C**

References

1. Amazon Web Services: Elastic GPUs, 2017; <https://aws.amazon.com/de/ec2/elastic-gpus/>
2. Bloom, N., Jones, C., Van Reenen, J. and Webb, M. Are Ideas Getting Harder to Find? Cambridge, MA, 2017; <https://doi.org/10.3386/w23782>
3. Bresnahan, T.F. and Trajtenberg, M. General purpose technologies 'Engines of growth'? *J. Econom.* 65, 1 (Jan. 1995), 83–108; [https://doi.org/10.1016/0304-4076\(94\)01598-T](https://doi.org/10.1016/0304-4076(94)01598-T)
4. Byrne, D.M., Oliner, S.D. and Sichel, D.E. Is the information technology revolution Over? *SSRN Electron. J.* (2013), 20–36; <https://doi.org/10.2139/ssrn.2303780>
5. Cavin, R.K., Lugli, P. and Zhirnov, V. V. Science and engineering beyond Moore's Law. In *Proceedings of the IEEE 100, Special Centennial Issue* (May 2012), 1720–1749; <https://doi.org/10.1109/JPROC.2012.2190155>
6. Dent, S. Major AMD chip supplier will no longer make next-gen chips, 2018; <https://www.engadget.com/2018/08/28/global-foundries-stops-7-nanometer-chip-production/>
7. Eastwood, G. How chip design is evolving in response to IoT development. *Network World* (2017); <https://www.networkworld.com/article/3227786/internet-of-things/how-chip-design-is-evolving-in-response-to-iiot-development.html>
8. *Economist*. The future of computing—After Moore's Law (2016); <https://www.economist.com/news/leaders/21694528-era-predictable-improvement-computer-hardware-ending-what-comes-next-future>
9. *Economist*. The chips are down: The semiconductor industry and the power of globalization (2018); <https://www.economist.com/briefing/2018/12/01/the-semiconductor-industry-and-the-power-of-globalisation>
10. ENIAC Report. Moore School of Electrical Engineering, 1946.
11. Feldmann, M. New GPU-accelerated supercomputers change the balance of power on the TOP500, 2018; <https://www.top500.org/news/new-gpu-accelerated-supercomputers-change-the-balance-of-power-on-the-top500/>
12. Gartner Group. Gartner Says Worldwide Semiconductor Revenue Grew 22.2 Percent in 2017. Samsung Takes Over No. 1 Position. Gartner, 2018; <https://www.gartner.com/newsroom/id/3842666>
13. Google Cloud. Google: Release Notes, 2018; <https://cloud.google.com/tpu/docs/release-notes>
14. Hemsoth, N. An Early Look at Baidu's Custom AI and Analytics Processor. The Next Platform; <https://www.nextplatform.com/2017/08/22/first-look-baidus-custom-ai-analytics-processor/>
15. Hennessy, J. and Patterson, D. *Computer Architecture: A Quantitative Approach* (6th ed.). Morgan Kaufmann Publishers, Cambridge, MA, 2019..
16. International Technology Roadmap for Semiconductors 2.0: Executive Report. International technology roadmap for semiconductors, 79, 2015; http://www.semiconductors.org/main/2015_international_technology_roadmap_for_semiconductors_itrs/
17. Jao, N. Taiwanese chip maker TSMC to build the world's first 3nm chip factory. *Technode*, 2018; <https://technode.com/2018/12/20/taiwanese-chip-maker-tsmc-to-build-the-worlds-first-3nm-chip-factory/>
18. Johnson, B., Tuan, S., Brady, W., Jim, W. and Jim, B. *Gartner Predicts 2017: Semiconductor Technology in 2026*.
19. Jouppi, N.P. et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the*

- 44th Annual Int. Symp. Comput. Archit. (2017), 1–12; <https://doi.org/10.1145/3079856.3080246>
20. Khan, H.N., Hounshell, D.A. and Fuchs, E.R.H. Science and research policy at the end of Moore's Law. *Nat. Electron.* 1, 1 (2018), 14–21; <https://doi.org/10.1038/s41928-017-0005-9>
21. Khazraee, M., Zhang, L., Vega, L. and Taylor, M.B. Moonwalk? NRE optimization in ASIC clouds or accelerators will use old silicon. In *Proceedings of ACM ASPLOS 2017*, 1–16; <https://doi.org/http://dx.doi.org/10.1145/3037697.3037749>
22. Krzanich, B. Intel Corporation Presentation at Sanford C Bernstein Strategic Decisions Conference, 2016.
23. Lapedus, M. Foundry Challenges in 2018. *Semiconductor Engineering*, 2017; <https://semiengineering.com/foundry-challenges-in-2018/>
24. Leiserson, C.E., Thompson, N., Emer, J., Kuszmaul, B.C., Lampon, B.W., Sanchez, D. and Scharld, T.B. There's plenty of room at the top: What will drive growth in computer performance after Moore's Law ends? *Science* (2020).
25. Martin, T.W. and Fitzgerald, D. Your love of your old smartphone is a problem for Apple and Samsung. *WSJ* (2018); <https://www.wsj.com/articles/your-love-of-your-old-smartphone-is-a-problem-for-apple-and-samsung-1519822801>
26. Mims, C. Huang's Law is the new Moore's Law, and explains why Nvidia wants arm. *WSJ* (2020); <https://www.wsj.com/articles/huangs-law-is-the-new-moores-law-and-explains-why-nvidia-wants-arm-11600488001>
27. NVIDIA Corporation. Tesla P100 Performance Guide. HPC and Deep Learning Applications, 2017.
28. Patton, G. Forging Intelligent Systems in the Digital Era. MTL Seminar, 2017; <https://www-mtl.mit.edu/mtlseminar/garypatton.html#simple3>
29. Pham, S. 2018. Who needs the US? Alibaba will make its own computer chips. *CNN Business*, 2018; <https://edition.cnn.com/2018/10/01/tech/alibaba-chip-company/index.html>
30. Prickett Morgan, T. Intel Takes First Steps To Universal Quantum Computing. *Next Platform*, 2017; <https://www.nextplatform.com/2017/10/11/intel-takes-first-steps-universal-quantum-computing/>
31. Putnam, A. et al. A reconfigurable fabric for accelerating large-scale datacenter services. *Commun. ACM* 59, 11 (Oct. 2016), 114–122; <https://doi.org/10.1145/2996868>
32. Santhanam, N., Wiseman, B., Campbell, H., Gold, A. and Javetski, B. McKinsey on Semiconductors, 2015.
33. Shalf, J.M. and Leland, R. Computing beyond Moore's Law. *Computer* 48, 12 (Dec. 2015), 14–23; <https://doi.org/10.1109/MC.2015.374>
34. Shao, Y.S., Reagen, B., Wei, G.Y., and Brooks, D. Aladdin: A pre-RTL, power-performance accelerator simulator enabling large design space exploration of customized architectures. In *Proceedings of the Int. Symp. Comput. Archit.* (2014), 97–108; DOI:<https://doi.org/10.1109/ISCA.2014.6853196>
35. Semiconductor Industry Association: 2017 Factbook; <http://go.semiconductors.org/2017-sia-factbook-0-0-0>
36. Smith, S.J. Intel: Strategy Overview, 2017; <https://doi.org/10.1016/B978-0-240-52168-8.10001-X>
37. Top500. The Green500 List (June 2019); <https://www.top500.org/green500/lists/2019/06/>
38. Worldometers. Computers sold in the world this year, 2018; <http://www.worldometers.info/computers/>

Neil C. Thompson (neil_t@mit.edu) is an innovation scholar in the Computer Science and Artificial Intelligence Lab and the Initiative on the Digital Economy at the Massachusetts Institute of Technology, Cambridge, MA, USA.

Svenja Spanuth (sspanuth@ethz.ch) is a Ph.D. candidate in the Department of Management, Technology, and Economics at ETH Zurich, Switzerland.

The authors contributed equally to the work.

Copyright held by authors.



Watch the authors discuss this work in the exclusive *Communications* video. <https://cacm.acm.org/videos/the-decline-of-computers>

A study of female students enrolled in IT degrees in Australia traces how programs influenced decision making.

**BY ANDREEA MOLNAR, THERESE KEANE,
AND ROSEMARY STOCKDALE**

Educational Interventions and Female Enrollment in IT Degrees

DESPITE INCREASING AWARENESS and efforts made to attract women to computing, they are still poorly represented in information technology (IT) careers.¹⁶ The number of females graduating with an IT degree has consistently declined since 1984 when women were 34% of computer science graduates and they

currently account for less than 20% of IT graduates in many countries.^{8,14,15} These figures are replicated in the IT industry where women currently constitute a small part of the workforce—24% in the U.S., 18% in the U.K., and 28% in Australia.^{5,6} This lack of diversity in IT has repercussions for organizations and for society. There is increasing evidence that diversity in the workplace has a positive influence on productivity.^{2,6} For example, Herring¹³ reports that organizations with high levels of gender diversity in teams have higher sales revenues, more customers and great-

er profitability than companies with predominantly male teams.

Finding capable and confident people with IT skills is very difficult, and many employers struggle to recruit employees, particularly as the talent pool is restricted by the lack of qualified women. Encouraging more women into IT university courses would lead to an increase in both the size and diversity of a country's skilled workforce and address the growing shortfall of IT professionals in the industry. Women's participation in the field could also lead to monetary benefits. For example, in the European

Event attendance and influence on enrolling in a computing-like degree.

Initiative	Description	Geographic Coverage^a
Bebras	An international competition for students in Year 3–12 to promote students' computational thinking.	National/International
Big Day In	A conference organized by university students for Year 9–12 students interested in careers in ICT and technology.	NSW, Vic, QLD, WA
Career Fairs	Career fairs to display different career opportunities to senior secondary students.	National
Club Kidpreneur/name change to Entropolis HQ	A program aimed to encourage Years 5–10 students in entrepreneurial thinking and improve knowledge about finances, business acumen and other skills.	National
Code Club	Coding club for children aged 9–13 years run by volunteers.	National/International
Code Like a Girl	A social enterprise to develop girls aged 8–12 programming skills through a three-day camp.	National
CoderDojo	Computer programming clubs for students aged between 7 and 17.	QLD, NSW, VIC, WA/International
Computer Games Boot Camp	An industry engagement event for students in Years 9–12 to learn about how games are designed and developed with insights into IT career paths.	VIC
ECOMAN	Aims to familiarize students about business concepts by using a business simulation program.	QLD
Endeavour	Workshop and a design expo aiming to familiarize students with engineering.	VIC
E.X.I.T.E. - Exploring Interests in Technology and Engineering	Camps for girls in Years 8–10 aimed to increase students' interest in STEM. They also explore the opportunities of contributing to the community and being creative in technology and engineering careers.	QLD, VIC, NSW/International
Females in Technology and Telecommunications	A network aiming to inspire and to support women in technology and communications.	National
FIRST LEGO League	Children in Years 4–9 solve real-world projects. In the process they have to build a robot and program it using LEGO® Mindstorms to solve an annual challenge.	National/International
FIRST Robotics Competition	Students in Years 9–12 design, build, program an industrial robot and compete against other teams to solve an annual challenge.	National/International
Girl GeekAcademy - #MissMakesCode	#MissMakesCode is an initiative to build confidence and self-efficacy in the areas of algorithmic thinking, programming and coding for young girls aged 5–8 years.	VIC/NSW
Girl Power in engineering & IT program	This program is specifically for girls and targets Year 9 students to attend a 3-day camp. In the following year, the students undertake work experience and in Year 11 and 12 the participants are given university student mentors.	VIC
Girls Programming Network	A program run by girls for girls as a one-day workshop. As part of the workshop, they develop games, mobile apps and learn about digital media and encourage high school girls to attend.	NSW
Hour of code	Events during which the students code.	National/International

ACT = Australian Capital Territory; NSW = New South Wales; NT = Northern Territory; QLD = Queensland; VIC = Victoria; SA = South Australia; TAS = Tasmania; WA = Western Australia.

^a Where the initiatives were online and there were no restrictions to where the participants need to be in order to enroll they are marked as national in the table.

Union, it is estimated that existing initiatives⁸ will improve women's participation in STEM-related fields and this will result in a 610–820 billion euros increase in the GDP.⁷ Except for monetary benefits, narrowing the gender gap in STEM could also improve science and society.⁸

While many approaches have been widely implemented, many others have failed to systematically improve

female participation at the university level¹¹ (and few have been successful). In the U.S., the Computer Science degree at Carnegie Mellon University reached almost 50% in 2016, 2017, and 2018⁹ and Harvey Mudd College increased its enrollment of women pursuing a Computer Science degree from 10% in 2006 to 40% in 2012.¹ Frieze and Quesenberry⁹ attributed the success to not changing the curriculum to be fe-

male-friendly but rather doing changes that improve the curriculum for everyone, changing the culture and institutional support for different programs. IT classes provided by the University of Cincinnati in the U.S. have also influenced some of the female students to register for the university's IT courses.¹⁰ Others^{3,4} show that integrating creative expression in computer science units can help attract and

Event attendance and influence on enrolling in a computing-like degree. (cont'd.)

Initiative	Description	Geographic Coverage^a
Indigenous Girls STEM Academy	Coding clubs, competitions and scholarship for high achieving indigenous girls in Years 8–11 to succeed in STEM careers.	NSW/WA & QLD
Computational and Algorithmic Thinking (CAT) AMT	This is a one-hour problem-solving competition aimed at students in Years 5–12 is designed to promote different ways of thinking including computational and algorithmic skills.	National
Informatics Olympiad	The Australian Informatics Olympiad (AIO) is an open national computer programming competition held annually for students up to year 10 and senior students up to Year 12. The top four students in Australia will be asked to represent Australia at the International Olympiad	National/International
(BrainSTEM) Innovation Challenge	Students in Years 9 or 10 are paired with a mentor and they work together for 12 weeks in a STEM related research environment.	VIC
Minecraft Competition	Competition aiming to engage youth with social “hot trends.” It aims to sparks creativity and collaboration by embracing new technology.	National
NCSS Challenge	Aimed at students in Years 5–12, to undertake this competition with training running for five weeks to learn or further develop their programming experience.	National
NCSS Summer School	A 10-day summer school holiday intensive program aimed at Year 11 students going into Year 12 to develop their skills in programming, robotics and Web design.	National
RACQ Technology Challenge Maryborough	Students compete and learn the use of technology and teamwork by building a vehicle.	VIC
Robocup Junior	Aims to introduce RoboCup Junior to primary and secondary school children to encompass engineering and IT skills.	QLD, NSW, VIC, WA, SA, TAS, ACT/ International
Robogals	Program aimed at primary and secondary school girls with the aim to improve their participation and confidence to work in engineering, science and technology,	QLD, NSW, VIC, WA, SA, ACT/ International
RoboGirls ^b	Female robotics competition aimed at girls.	-
(Australian) STEM Video Game Challenge	Aimed at students in Years 5–12 this challenge aims to address students' perceptions of STEM subjects.	National
SuperDaughter Day	Girls aged 5–12 years of age participate in hands on activities, including virtual and augmenting reality, app design, technology and wearables along with their parents. They also meet role models from industry.	National/International
Tech Girls are Superheroes	Girls aged between 7–17 years of age compete by solving a problem through the development of a mobile app and a business plan.	National & New Zealand
Tech School experience/workshop	Hands-on workshops on engineering and IT for school students.	VIC
Women in Technology	Supports women on all career stages from female students to women already working in the field.	QLD
Young ICT Explorers	A competition which students from Years 3–12 work on an IT project of their choice. The students showcase their projects which are judged according to the following criteria: Creativity and Innovation, Quality and Completeness, Level of Difficulty and Documentation.	National

ACT = Australian Capital Territory; NSW = New South Wales; NT = Northern Territory; QLD = Queensland; VIC = Victoria; SA = South Australia; TAS = Tasmania; WA = Western Australia.

a Where the initiatives were online and there were no restrictions to where the participants need to be in order to enroll they are marked as national in the table.

b Possibly the same initiative as Robogals.

retain more students. For example, the Technology, Arts, & Media degree at the University of Colorado, Boulder achieved a 44.8% female enrollment³ and Computing in Arts at the College of Charleston attracted and retained 46% female students.⁴

These initiatives are a positive step toward understanding how to address some of the issues that cause the low enrollment of women in IT units. In

contrast to these studies, this article focuses on existing programs and initiatives in Australia. We examine how female students enrolled in IT-related degrees perceived these initiatives. The need for further research in this area is highlighted also by Tims,¹⁹ who mentions that despite the multitude of initiatives and sustained efforts from many organizations, progress is slow.

Methodology

The authors researched existing programs and initiatives aimed at promoting IT in Australia. A total of 36 initiatives were identified, some aimed only at girls while others were aimed at both male and female. A brief description of each initiative and geographic coverage is presented in the accompanying table. This is not an exhaustive list of initiatives. To avoid some influential pro-

grams being missed during our research, the participants were also asked to provide further details about other programs they were involved in.

Data is collected through the use of a questionnaire. These were distributed to first-year female students enrolled in computing-related degrees in Australia. The questionnaire uses a combination of closed and open-ended questions. The participants were asked demographic questions (for example, in which state they lived during high school) and were asked to state what initiatives they participated in (if any) during high school. Furthermore, they were asked to expand whether they

found that any of these initiatives motivated them to enroll in a computing degree. The open-ended questions were analyzed using thematic analysis.¹²

Participants

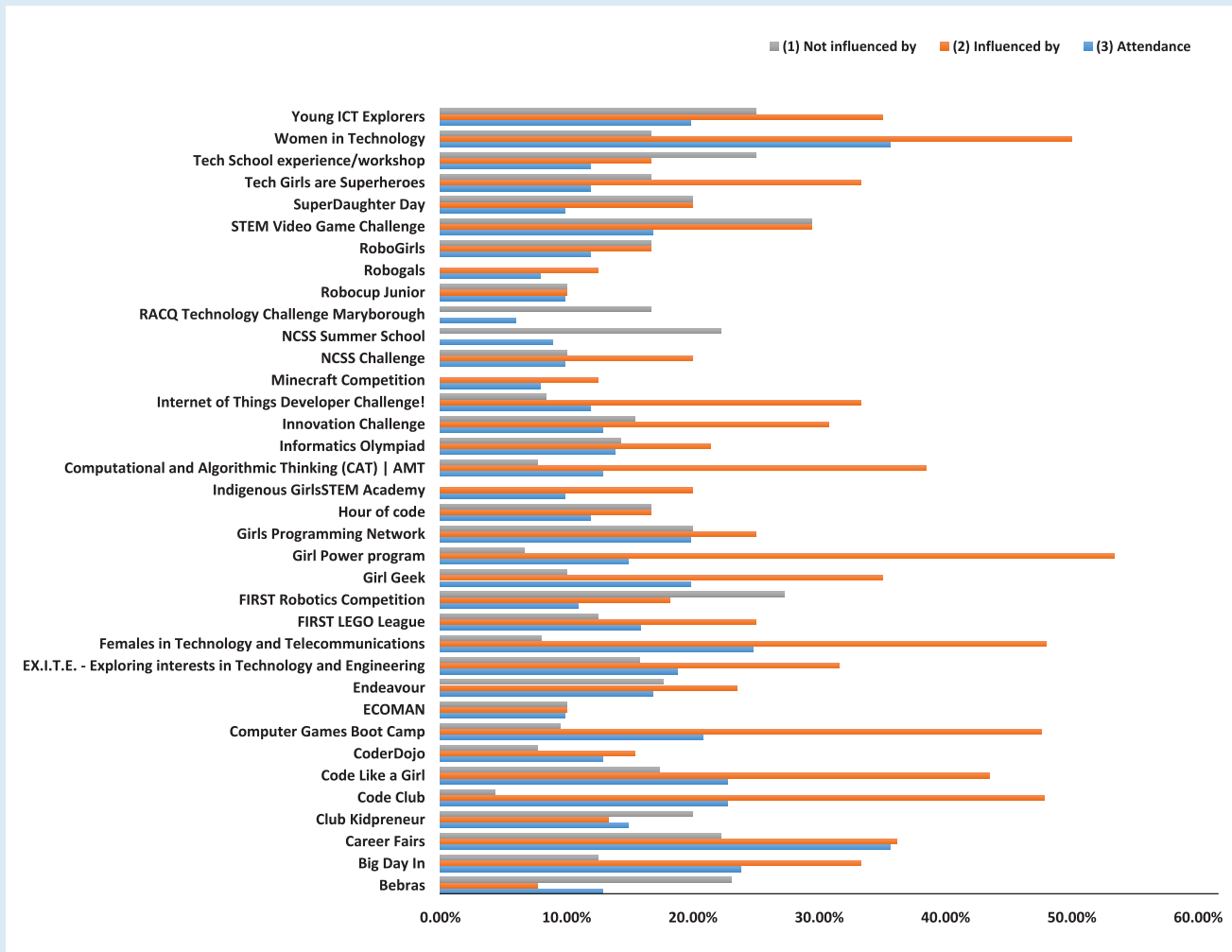
A total of 119 participants who identified themselves as females completed the survey. Among these, we eliminated the ones who did not fully complete the questionnaires, and the ones who lived in Australia for less than one year and did not complete their secondary education in Australia. They were eliminated as they were very likely not exposed to the programs. This led to 108 participants included in the analysis. Our final

sample size included 8% of students that have completed their secondary education abroad but have lived in Australia for more than one year.

Most of the participants lived in Australia all their life (72%) and all of them were first-year undergraduate students enrolled in an IT degree (for example, Computer Science, Business Information Systems). Although we did not plan for it, our sample size included participants completing their education across eight Australian states and Territories. We had more participants from Victoria and New South Wales than other states, but these are also the most populated states in Australia.

Event attendance and influence on enrolling in a computing-like degree.

- (1) Percentage of participants that attended the event;
- (2) Among those who participated in the event, percentage that felt influenced or potentially influenced by the event to enroll in a computing like degree;
- (3) Among those that participated in the event, percentage who felt that the event did not have a role in their enrollment in a computing related degree.



Findings

Exposures to programs. Our participants were exposed to a variety of programs (on average being exposed to 5.4 programs). The accompanying figure presents on the left-hand side the programs the participants had engaged with. Also, the participants mentioned two other programs not included in our questionnaire. The most exposed program (on the same level to Career Fairs) was Women in Technology to which 36% of the participants took part in. This was followed by Big Day In and Females in Technology both with 24%, Code like a Girl (23%) and Code Club (23%).

Influential programs. The second aim of our study was to determine whether the participants perceived these programs as being useful in influencing them to enroll in computing/IT type degrees. The participants had mixed feelings about the programs but overall, they have influenced many of the participants. Some of the participants felt strongly about the influence the programs had while others felt that the programs probably had some influence or at least motivated them to find more about the field. Others felt that there were other factors (for example, current workplace) that influenced them and not necessarily the programs. We grouped all the replies into positive and negative ones. As the participation in the programs was not uniform, the figure presents the replies relative to the number of participants in the program.

The most influential program for our participants was Women in Tech. A total of 50% of the participants in the study and program mentioned that they were influenced or probably influenced by this program. This was followed closely by Females in Technology and Telecommunications (48%). Although programs dedicated to females only have been perceived as the most influential, other programs such as Code Club and Computer Games Boot Camp also ranked highly.

Moving Forward

This article adds further insights into how programs and interventions influence female students' decision to enroll in IT-related degrees. A survey was targeted at first-year female students to report on which, if any, outreach pro-

grams influenced their decision to enroll in an IT degree. We found there was no single model of intervention and that different programs influenced different people. This variety of offerings contributed to attracting a diverse pool of students. Notably, those programs aimed specifically at females were perceived to potentially have more influence on the decision to enroll in an IT degree. This does not negate the influence of mixed gender programs but does highlight the importance of having programs aimed only at female students alongside more generalized outreach programs. [Additionally, students had often participated in more than one program and further research in determining how they personally rank the effectiveness of different programs and why they attended them is required.

Further studies are also needed to confirm whether these findings can be replicated more broadly and with an exhaustive list of existing initiatives. For example, this study only addresses the perceptions of current female IT students and does not consider how other students, who may not have been successful in enrolling in an IT degree, perceived the value of the programs. A further factor of interest is whether positive participation in these programs is instrumental in the decision to study IT or whether participation affirms an existing intention. Finally, this study included initiatives at both national and state level. The most popular initiatives were those delivered at a national level which are more widely offered, although we have no evidence that availability affects the findings. Further research is needed to determine the common and unique characteristics of each of these initiatives, how they were advertised and what support was made available to schools to encourage attendance. This would enable more effective targeting of initiatives to overcome the perennial problem of recruiting women into IT programs.

Acknowledgments

This research has been partially funded by the Australian Council of Deans of Information and Communications Technology. We want to thank Thomas Stockdale who has helped with the participants' recruitment for the study

and the Australian Academy of Science which has provided information about some of the different initiatives running in Australia. □

References

- Alvarado, C., Dodds, Z., and Libeskind-Hadas, R. Increasing women's participation in computing at Harvey Mudd College. *ACM Inroads* 3, 4 (2012), 55–64.
- Australian Computer Society. The Promise of Diversity: Gender Equality in the ICT Profession, 2015; http://acs.org.au/_data/assets/pdf_file/0003/87681/ACS-Gender-Equality-FINAL.pdf
- Barker, L. J., Garvin-Doxas, K., and Roberts, E. (2005). What can computer science learn from a fine arts approach to teaching? *ACM SIGCSE Bulletin* 37, 1 (2015), 421–425.
- Bares, W. H., Manaris, B., McCauley, R., and Moore, C. Achieving gender balance through creative expression. In *Proceedings of the 50th ACM Technical Symp. Computer Science Education* (Feb. 2019). ACM, 293–299.
- Deloitte. Technology, Media & Telecommunications Predictions, 2016; <http://www2.deloitte.com/content/dam/Deloitte/global/Documents/Technology-Media-Telecommunications/gx-tmt-prediction-2016-full-report.pdf>
- Deloitte Access Economics. Australia's Digital Pulse: Key Challenges for Our Nation: Digital Skills, Jobs and Education, 2015; <http://www2.deloitte.com/content/dam/Deloitte/au/Documents/Economics/deloitte-au-economics-australias-digital-pulse-240614.pdf>
- EIGE. Economic case for gender equality in the EU, 2018; <https://eige.europa.eu/gender-mainstreaming/policy-areas/economic-and-financial-affairs/economic-benefits-gender-equality>
- Fatourou, P., Papageorgiou, Y., and Petousi, V. Women are needed in STEM: European policies and incentives. *Commun ACM* 62, 4 (2019), 52–52.
- Frieze, C. and Quesenberry, J.L. How computer science at CMU is attracting and retaining women. *Commun. ACM* 62, 2 (2019), 23–26.
- Fritz, J., Wulf, T., Matthews, M., and Scott, J. University of Cincinnati and Saint Ursula Academy partnership: Introducing female high school students to the field of information technology. In *Proceedings of the 16th ACM Annual Conference on Information Technology Education* (Sept. 2015), 9–14.
- Gorbacheva, E., Beekhuizen, J., vom Brocke, J., and Becker, J. Directions for research on gender imbalance in the IT profession. *European J. Information Systems* 28, 1 (2019), 43–67.
- Guest, G., MacQueen, K.M., and Namey, E.E. Applied thematic analysis. Sage Publications, 2011.
- Herring, C. Does diversity pay?: Race, gender, and the business case for diversity. *American Sociological Review* 74, 2 (2009), 208–224.
- Hutchinson, J. and Tadros, E. Shortage of IT graduates a critical threat, 2014; http://www.afr.com/p/technology/shortage_of_it_graduates_critical_tOuFEdBporFCdLionKFJfJ
- NCWIT (Producer). Women and Information Technology By the Numbers, 2014; http://www.ncwit.org/sites/default/files/resources/btn_02282014web.pdf
- National Science Foundation. Science and Engineering Indicators. 2018; <https://nsf.gov/statistics/2018/nsb20181/report>
- OECD. Education at a Glance 2014: OECD Indicators; <http://dx.doi.org/10.1787/eag-2014-en>
- OECD. Students, Computers and Learning: Making the Connection, PISA, 2015; <http://dx.doi.org/10.1787/9789264239555-en>
- Tims, J.L. Achieving gender equity: ACM-W can't do it alone. *Commun. ACM* 61, 2 (Feb. 2018); 5.

Andreea Molnar (amolnar@swin.edu.au) is a senior lecturer at Swinburne University of Technology, Melbourne, Australia.

Therese Keane (tkeane@swin.edu.au) is an associate professor and Deputy Chair of Department, Strategic Initiatives and Partnerships, Education at Swinburne University of Technology, Melbourne, Australia.

Rosemary Stockdale (r.stockdale@griffith.edu.au) is Head of the Department of Business Strategy and Innovation at Griffith University, Melbourne, Australia.

DOI:10.1145/3430803

Under optimistic projection models, gender parity is forecast to be reached after 2100.

BY LUCY LU WANG, GABRIEL STANOVSKY,
LUCA WEIHS, AND OREN ETZIONI

Gender Trends in Computer Science Authorship

THIS ARTICLE PRESENTS a large-scale automated analysis of gender trends in the authorship of Computer Science literature. Specifically, we aim to address the following questions:

- ▶ How is gender balance among authors changing over time?
- ▶ When might gender parity be reached among authors?
- ▶ How is gender associated with co-authorship?
- ▶ And how does Computer Science compare against other fields of study?

We answer these questions by performing an automated study of literature metadata from scientific conferences and journals, using data from

the Semantic Scholar academic search engine.^a Our study incorporates metadata from 11.8M Computer Science publications. To provide a basis for comparison, we also analyze more than 140M articles from other fields of study. Our results demonstrate that although progress has been made, there is still a significant gap in gender representation among Computer Science authors. Continued delay in addressing the gender gap may perpetuate imbalances for generations to come.

Data

Our analysis was performed over the Semantic Scholar literature corpus.² The corpus contains publications between 1940 and the end of November 2019, and associated metadata such as title, abstract, authors, publication venue, and year of publication. Metadata in Semantic Scholar are derived from academic publishers, as well as scientific repositories such as arXiv, DBLP, and PubMed. We use the 19 fields of study defined by Microsoft Academic,²⁵ which are integrated with Semantic Scholar data. Table 1 shows the distribution of articles used in our analysis by field of study.

The author list is extracted from all publications and compiled into a list of first names. We use Gender API^b to perform gender lookup for each name. Gender API is a large online database of name-gender relationships derived by linking publicly available governmental data with social media profiles in

a <https://www.semanticscholar.org/>

b <https://gender-api.com/>

» key insights

- If current trends hold, gender parity among Computer Science authors will not be reached in a century.
- Computer Science lags behind other fields of study in equal gender representation among authors.
- Given the magnitude and trends associated with this gender gap, policy changes may be necessary to address these disparities in the short term.



various countries. For each name, Gender API outputs the predicted binary gender (*female* or *male*), along with the accuracy associated with the prediction and the number of samples used to arrive at that determination. We exclude authors for whom first names are missing, and for whom only first initials are available. We also filter out first names that occur less than 10 times in our overall corpus, to reduce the number of API calls to manageable numbers.

Because many names are ambiguous with respect to gender, we use the accuracy returned by Gender API to represent the gender of each author as a distribution over male and female probabilities. For example, Gender API estimates the first name Matthew to be male with an accuracy score of 100, the maximum. The name Taylor, however, is estimated to be female but

only receives an accuracy score of 55. These accuracies are used to generate two probabilities for each name, (m, f) , where m is the probability of the associated author being perceived as male, and f is the probability of the associated author being perceived as female, where $m + f = 1$. In this example, each author named Matthew will be represented with the probability tuple $(1.0, 0.0)$, and each author named Taylor will be represented as $(0.45, 0.55)$.

We acknowledge that gender identity is fluid and nonbinary. However, for the sake of this large-scale study, we adopt a simplified view of gender as a probability distribution over two genders, relying on first names as a proxy for the author's perceived gender (as opposed to self-reported gender). We use Gender API's results as an estimation of authors' perceived binary gen-

der, and use these estimates to generalize over our corpus. We are not making claims about any author's true self-reported gender.

Analyses

We perform two types of analysis on this data. First, we analyze publication trends, examining the number and proportion of female authors over time. To identify when gender parity may be reached, we project the proportion of female authors based on trends from the last 50 years (since 1970). In this article, we define parity as the proportion of female authors falling within 10% of 0.5, within the range of 0.45–0.55. We also study trends in co-authorship behavior as reflected in our data.

Authorship analysis. Most articles are authored by more than one individual. For the purposes of our analysis, each

Table 1. Corpus statistics for different fields of study.

Field of study	Total articles	Total author-article units	Average authors per article
Art	5.3M	7.4M	1.4
Biology	15.1M	55.2M	3.7
Business	3.7M	5.8M	1.6
Chemistry	14.7M	48.6M	3.3
Computer Science	11.8M	27.3M	2.3
Economics	3.8M	6.4M	1.7
Engineering	10.1M	20.9M	2.1
Environmental Science	2.0M	4.6M	2.3
Geography	4.0M	7.3M	1.8
Geology	3.2M	8.4M	2.6
History	6.0M	8.2M	1.4
Materials Science	7.4M	21.7M	2.9
Mathematics	5.5M	10.9M	2.0
Medicine	32.4M	111.9M	3.4
Philosophy	2.8M	3.9M	1.4
Physics	7.8M	31.0M	4.0
Political Science	4.9M	6.8M	1.4
Psychology	7.0M	14.7M	2.1
Sociology	4.6M	6.3M	1.4
Total	152.1M	407.2M	2.7

author-article pair is treated as one unit. An article with a single author yields one author-article unit; an article with three authors yields three author-article units, etc. In Computer Science, the average number of authors is approximately 2.3 per article. However, average authors per article have increased from approximately 1.5 per article in 1970 to approximately 3.0 in the past several years, which reflects patterns observed by other researchers.¹¹ Appendix B (available online at <https://doi.org/10.1145/3430803>) provides further discussion of this shift in relation to concurrent increases in author count in other fields.

The proportion of female authors over time is used to project the trend toward gender parity. The number of female authors in a given year is computed as the sum of probabilities f over the author-article units of that year, and the number of male authors is correspondingly generated as the sum of probabilities m . The proportion of female authors for each year F_t is computed as the num-

ber of female author-article units divided by the total number of author-article units for the corresponding year. We compute projections by performing an autoregressive integrated moving average (ARIMA) analysis, a widely used and established method for creating time series forecasting models.⁴ ARIMA is an autoregressive forecasting technique, which means it uses historical values in a time series to predict current and future values. We use the auto ARIMA function in the R “forecast” package,¹⁴ which automates the selection of ARIMA model order, with a preference for simple models with lower order.

We assume that the growth in female author proportion observes logistic behavior. The proportion of female authors is necessarily constrained between 0 and 1, and logistic growth assumes that a stable equilibrium will eventually be reached. We tested other fit functions (linear and exponential; see Appendix C at <https://doi.org/10.1145/3430803> for details), but found them to be less suitable; the root-mean-squared-error (RMSE) of the logistic fit is lower than that of these other curve types when fitting to the growth curves of each field of study.

To perform the fit, we first apply σ_α^{-1} , the inverse of the α -scaled sigmoid (or logit) function $\sigma_\alpha(x) = \alpha/(1+\exp(-x))$, to map the gender proportion into the real number line so that the data is more amenable to linear approximation. We call α the expected equilibrium proportion parameter. This transform generates $y_t = \sigma_\alpha^{-1}(F_t)$, where F_t is the proportion of female authors per year. We then fit a nonseasonal ARIMA model with parameters p , d , and q for the transformed process y_t represented by the following equation:

$$\phi_p(B)(1 - B^d)y_t = c + \theta_q(B)\varepsilon_t \quad (1)$$

where B is the backshift operator, which shifts by one to the previous time point, and ε_t is zero-centered, normally distributed noise.¹⁴

Finally, we obtain the forecast in the original domain using a sigmoid transform over the projected values, applying σ_α to y_t for $t > 2019$. We sample α from the range $[0.3, 1.0]$ so that σ_α has minimum and maximum values of 0 and α , respectively. This constrains the projected values to be between 0 and some expected

equilibrium proportion defined by α . The 80% and 95% confidence intervals of the prediction are computed from averaging the projection confidence over 10000 iterations of model fitting.

The range for α is defined based on the space of likely equilibrium proportions, as estimated based on trends observed in various fields of study (see Figure 4). Note that α represents the proportion of female authors we expect in the long run. An equilibrium proportion of 0.5 indicates that we expect the authorship makeup to eventually stabilize at around 50% men and 50% women. An equilibrium proportion of 0.9 indicates that we expect the authorship makeup to eventually stabilize at around 10% men and 90% women. As we will elaborate later, we perform a sensitivity analysis to determine the effect of the selected α parameter on the year in which parity is expected to be reached.

Co-authorship analysis. Co-authorship is computed for each unique pair of author-article pairs for each article. If an article has n authors, $\binom{n}{2}$ co-author pairs are generated. Given one co-author pair (n_1, n_2) and associated gender probabilities $n_1 \rightarrow (m_1, f_1)$ and $n_2 \rightarrow (m_2, f_2)$, we compute three probabilities, p_{mm} , p_{mf} , and p_{ff} , corresponding to the gender combinations, that is., between two male authors, a male and a female author, and two female authors, respectively:

$$\begin{aligned} p_{mm} &= m_1 m_2 \\ p_{mf} &= m_1 f_2 + f_1 m_2 \\ p_{ff} &= f_1 f_2 \end{aligned} \quad (2)$$

where $p_{mm} + p_{mf} + p_{ff} = 1$. The numbers of each type of co-author pair for each year are computed by summing over the above probabilities over all co-authorship pairs of that year.

We then assess the number of same-gender and different-gender collaborations over time. The results are measured as a deviation from the expected, where the expected co-authorships are determined by sampling from the numbers of female and male authors active in a given year, assuming the same number of collaborations per year as observed in our data. The total number of extra or missing collaborations is computed as the difference between the observed counts of each type of collaboration and the expected value. To show

rates of change, we also compute the ratio between observed and expected collaborations (O/E) of each type.

Results

Here, we discuss the main findings of our study.

Gender API results. The 152.1M articles in our corpus resulted in 407.2M author-article units. Of these author units, 14.5M lack first names, 110.0M have only a first initial, and 5.7M have a first name that occurs less than 10 times in the corpus. These author units are removed from further analysis. The remaining 277.0M author units are associated with 521K unique first names. We query these 521K names in Gender API, and acquire gender information for 351K; 170K names have insufficient information and are excluded from analysis. Of the 11.8M articles in Computer Science and the 27.3M author-article units therein, 24.1M authors have valid first names, and 16.9M author-article units (61.8%) resulted in associated gender information, which is higher coverage compared to authors in other fields (we acquire gender information for approximately 50.4% of authors across all fields).

Gender trends among authors. Figure 1 shows that the overall author count in Computer Science has increased substantially over the last several decades, as the field has experienced significant growth. The total number of author-article units in 2018 is above 1.2M. The proportion of female authors has also increased during this time.

Figure 2 shows the projected proportion of female authors in Computer Science. Residuals of the ARIMA fit line over the logit-transformed data appear normally distributed and are not significant under the Shapiro-Wilk Normality Test.²⁴ The proportion of female authors in Computer Science is predicted to reach 0.45 around 2124, more than 100 years from now. The upper bound of the 95% CI reaches 0.45 in 2065, and the lower bound of the 95% CI reaches 0.45 beyond the range of our projection. Appendix A (available online at <https://doi.org/10.1145/3430803>) provides further discussion on model choice and the sensitivity of ARIMA projections to the choice of the equilibrium parameter.

We also make the somewhat concerning observation that the rate of growth in female author proportion has slowed in recent years, visible in Figures 2 and 4. Our projection makes the optimistic assumption that the proportion will continue to grow towards or beyond parity, but the data may suggest otherwise. It remains to be seen whether a new trend is emerging that exhibits not an increase, but rather a leveling off or decrease in the proportion of female authors.

Association of gender and co-authorship. The numbers of same- (*male-male* or *female-female*) and cross-gender (*male-female*) co-authorships in Computer Science are computed for each year. Figure 3 shows the difference between the number of observed

and expected collaborations of each type since 1990.^c In this time period, there are more same-gender co-authorships than would be expected, and fewer cross-gender co-authorships than would be expected. In recent years, around 50000 cross-gender co-authorships per year were missing when compared to expected numbers.

The observed to expected ratio shows both optimistic and pessimistic collaboration trends. Although both men and women are more likely to co-author with authors of their own gender (positive O/E), the degree of same-gender bias is declining

^c We show collaboration counts after 1990 because there is higher data volume in this period of time.

Figure 1. Gender of Computer Science authors over time, computed by averaging across gender probabilities in our dataset.

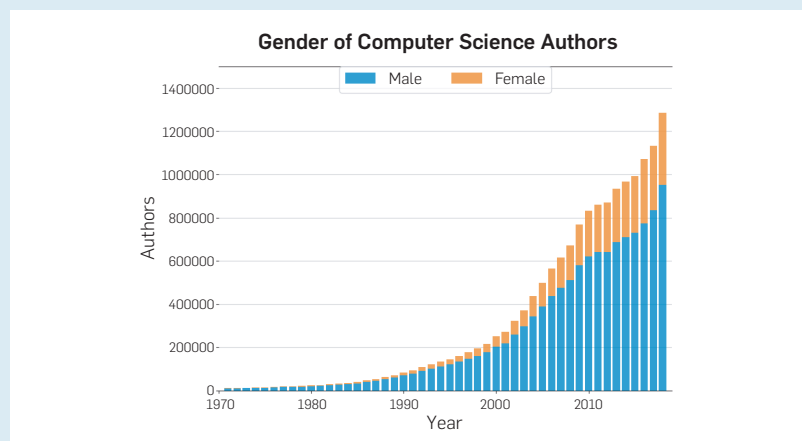


Figure 2. The proportion of female authors is projected using an ARIMA model assuming logistic growth toward equilibrium proportions in the range [0.3, 1.0]. Confidence intervals at 80% and 95% are shown.

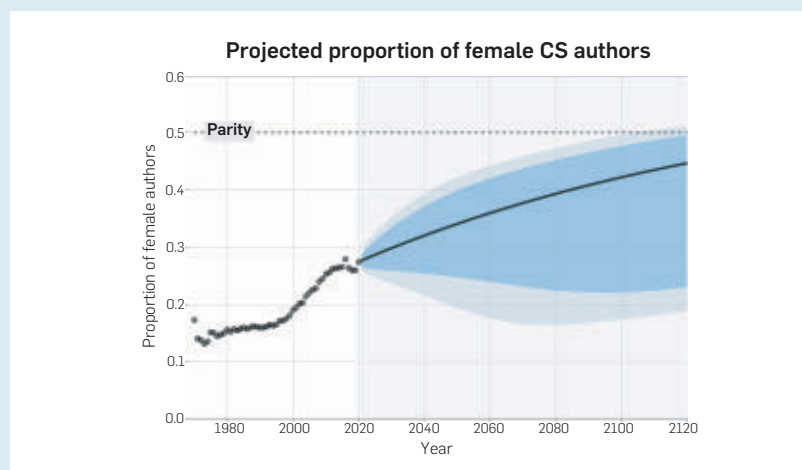
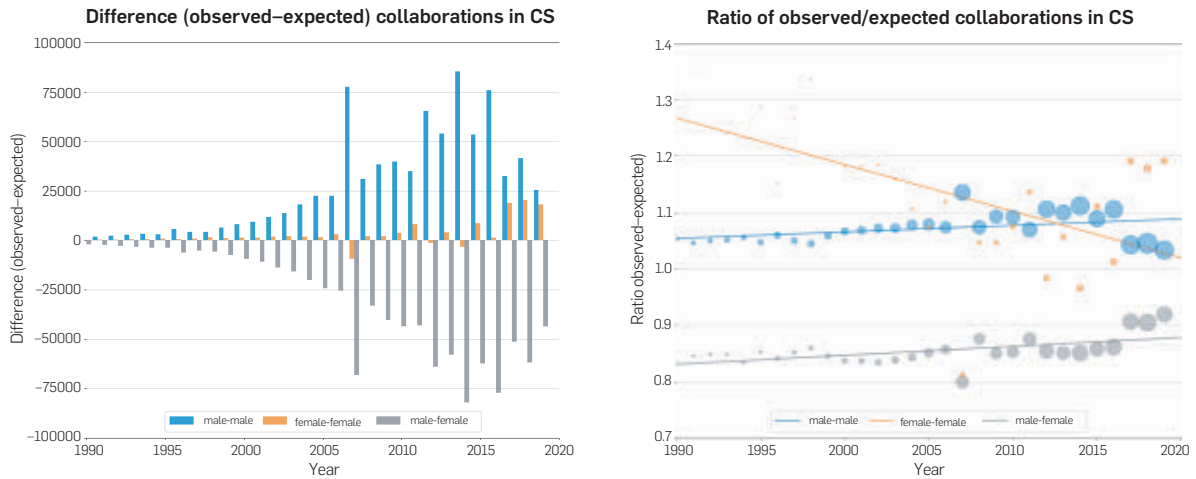


Figure 3. The difference (left) and ratio (right) between observed and expected same- and cross-gender co-authorships in Computer Science since 1990. Marker size for the O/E ratio is proportional to the number of expected collaborations of that type in each year.



among female authors but potentially increasing among male authors. At the same time, the cross-gender collaboration gap ($O/E < 1.0$) is still rather large, such that in recent years, only around 90% of expected cross-gender collaborations are observed. In other words, although there are more opportunities for cross-gender collaboration in recent years (due to an increase in the number of female scientists working in the field), the observed number of cross-gender collaborations is still below what would be expected. Optimistically, these trends may be shifting in the recent past, with numbers from the last three years showing a shift toward more cross-gender co-authorship; although it is too early to say whether this tendency will preserve itself in the future.

Comparison of CS with other fields of study. Figure 4 shows the proportion of female authors in 19 fields of study over the last 80 years. Computer Science is among the fields with the lowest female representation in recent years despite having relatively higher female representation in the middle of the 20th century.

Discussion

Our analysis of the Computer Science literature reveals the persistent patterns of inequality in gender and academic authorship. Although gender

balance among authors is improving, progress is slower than we had hoped.

Limitations. Inferring gender from first names is imperfect, and all gender-inference tools are subject to biases. Several studies have described and measured the differences between these services.^{15,22} Based on results in Santamaría and Mihaljević,²² Gender API has the lowest overall error rate but was slightly biased toward underrepresentation of females in their evaluation; in other words, the number of women estimated may be slightly lower than in reality. However, this bias may be offset by our sampling bias, because the population of CS authors is unlikely to be an unbiased sample of the general population, or the population whose names were used to construct the database behind Gender API. We attempt to mitigate some of these biases by treating the perceived gender as a probability distribution. One way to compute a more precise estimate is to weight the probabilities assigned by Gender API to each name using the prior probabilities of being a female or male CS author; this would likely produce a more pessimistic projection.

The proportion of authors in our corpus with high uncertainty in Gender API results has also grown over time. The average confidence of our gender predictions decreased from around 95% in 1970–2000 to 90% since 2005. We show and discuss this

change in confidence in Appendix D (available online at <https://doi.org/10.1145/3430803>). Although Gender API's average prediction confidence in our corpus is still high, this trend may pose a challenge for similar analysis in the future. Upon inspection of the data, we attribute this to the growing number of East Asian authors publishing in recent years. East Asian first names, when romanized, are more gender ambiguous. Gender API outperforms other gender lookup services, but still has lower overall confidence on names of East Asian origin.²² In Mattauch et al.,¹⁸ the authors explicitly exclude all authors with East Asian names from their name list during analysis, yet this accounts for the removal of more than 35% of their dataset. Rather than removing an entire group of authors from our data, we believe that representing each author name as a distribution of gender probabilities offsets some of the issues of increasing gender ambiguity in our corpus over time.

We also recognize the limitations of using author-article pairs as our units of measure. We do not distinguish between a person who is a single author on an article, and a person who co-authors with many others. This biases our data by overweighting articles with more authors. Similarly, in our analysis of collaboration, we take each combination of authors for an article as a collaborating pair, which again overweights

articles with more authors. In the Computer Science corpus, we observe an increase in the average authors per article over time, growing to approximately 3.0 authors per article in the last two years. However, Computer Science articles are still generally authored by smaller groups of individuals in the lower single digits, and we believe the bias introduced by our usage of author-article pairs or collaborating author pairs to be minimal.

Each author on a publication is also weighted equivalently in our analysis. We acknowledge that this discounts the special recognition extended to first authors, last authors, and single authors; we point readers to previous studies that have already demonstrated the distinctions between these groups.²⁷

Lastly, our projection of female author proportion uses data from the last 50 years to project more than 100 years into the future. We understand the inaccuracies of making such an extensive forecast with limited data. The goal of our projection is not to provide a definitive answer to the question of when gender parity will be reached among Computer Science authors; rather, the projection signals that even under optimistic growth, the gender gap will likely not close in the near future without some form of

community or external intervention. Observed recent trends also suggest that the increase in female representation among Computer Science authors may be slowing in the last five years. The long range forecasts we show may not adequately capture changes on this shorter time scale. Our forecasts also do not reflect changes that would result from newly introduced or as yet unimplemented interventions.

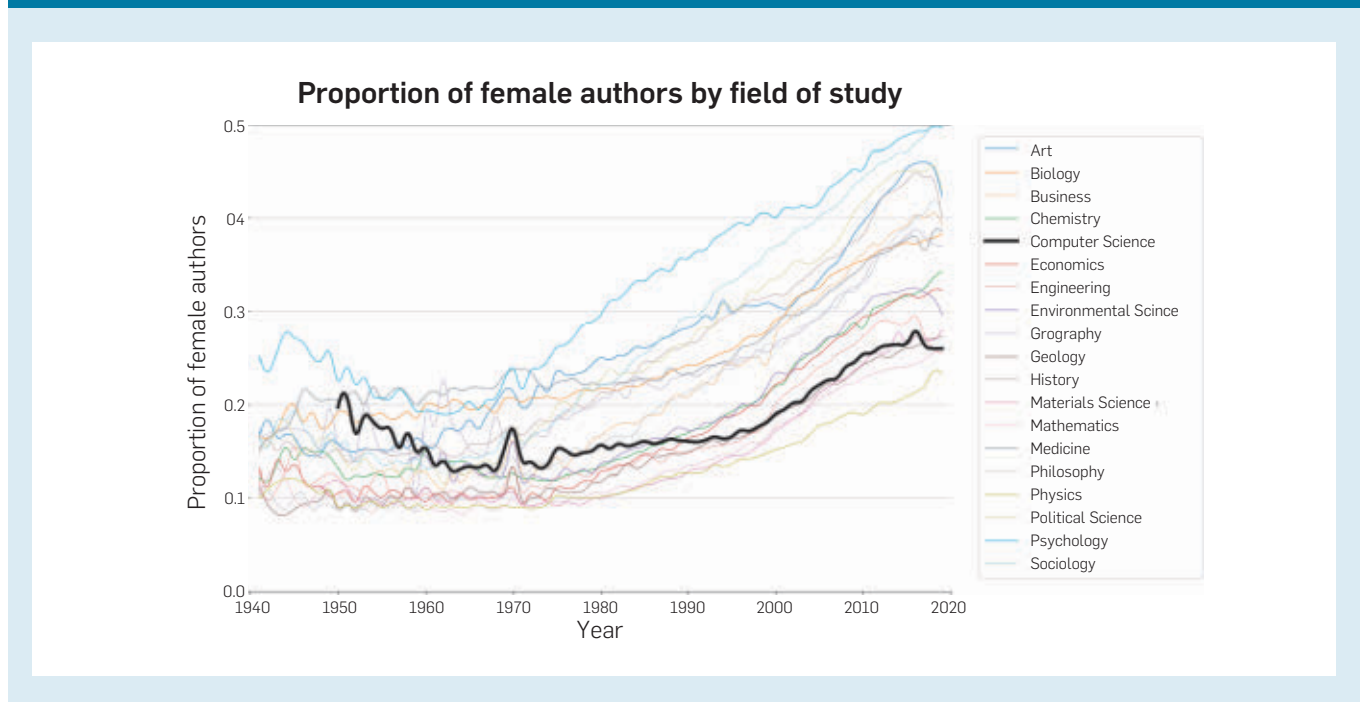
Prior work. Inequality in gender representation is a well-documented and studied issue in academia. Studies have shown that existent and perceived gender biases may affect many aspects of career and academic success, including but not limited to a woman's choice of college major,²¹ crediting in scientific publications,¹⁰ access to mentorship,^{9,20,23} rate of promotions,⁷ opportunities for collaboration,¹ as well as publishing and citation trends.^{18,19} All of these factors can lead to imbalanced representation of women in certain fields of study.

With the increasing digitization of scholarly communication and availability of publication-related metadata, scholars have been able to better quantify inequality in authorship. Cohoon et al.⁸ analyzed 86,000 ACM conference articles and showed increasing representation of women authors publishing at

Computer Science venues, which strongly correlated with increasing numbers of female Computer Science PhDs.⁸ West et al.²⁷ analyzed 1.8 million articles from JSTOR, a large multidisciplinary repository of academic literature, and revealed that although gender gaps are shrinking in academic publications, women were found to be significantly underrepresented as last and single authors. Elsevier, a large publisher of research articles, in an analysis of data from Scopus and ScienceDirect, reported the presence of gender imbalance among authors and inconsistent trends toward equal representation in different fields.¹ A study in 2018 confirmed continuing gender disparities among Nature Index journals, commonly considered some of the most reputable sources of academic literature, and in particular, limited representation of women among last authors, who are often perceived as more senior.³ Our work demonstrates that the gender gap is persistent and relatively large among Computer Science authors, which is consistent with the results of these studies.

A study of gender bias in authorship conducted by Holman et al.¹³ projected the closing of the gender gap in various fields based on recent trends. Through analyzing 9.1 million articles from PubMed, the authors projected that

Figure 4. The proportion of female authors among 19 fields of study. Proportion is plotted if there are more than 1,000 author-article units for which we could obtain gender information in a particular year.



gender parity would be reached in around 20 years in certain biomedical fields such as Molecular Biology, Medicine, or Biochemistry. Holman et al.'s analysis of a small corpus of Computer Science preprints from arXiv showed that gender parity in Computer Science will be reached in more than 100 years from the present.¹³ Also corroborating our estimate is related work from Way et al.,²⁶ which forecasts that gender parity in CS faculty hiring will be reached around 2075. Due to the long duration of faculty careers, parity in hiring would be expected to precede parity in publication and overall representation. Our results confirm and expand upon the results of this prior work. We use a significantly larger corpus of literature metadata to place the trends observed in Computer Science in the context of other fields of study. Additionally, we provide an assessment of co-authorship trends, which demonstrate a gap in cross-gender collaborations among CS authors.

Major strides have been made to reduce gender disparities. The presence of an overall structure of sexism in academia continues to be debated,^{5,16,17} but many academic institutions recognize the issue and have sought to equalize admissions and hiring procedures. Evidence of movement toward more equitable representation in hiring and publication has been observed in some controlled settings.^{6,12,28} How these observations translate into systemic change remain to be seen. Our results suggest, however, that the current pace of change in Computer Science will not result in a rapid closing of the gender gap.

Conclusion

We performed a large-scale analysis of the Computer Science literature (11.8M articles) to evaluate gender trends among authors. Based on trends over the last 50 years, the proportion of female authors in Computer Science is forecast to reach parity beyond the end of this century, and under different assumptions, it may take far longer. In this regard, Computer Science trails other fields of study, where we may want to look for inspiration. We also observed lower than expected numbers of cross-gender collaborations, with a gap of approximately 50000 cross-gender collaborations per year in the last several years.

Unless a major shift occurs that changes the gender makeup of the Computer Science community, the authorship gender gap will likely persist for a long time. Given the pervasiveness of computing technologies in our daily lives, it is of utmost importance that the researchers, designers, and builders of these technologies reflect the diversity of their users. Gender is one type of diversity among many that can be more easily assessed using the types of automated methods we employ. We hope that these findings will motivate members of the community to reflect upon the causes of these disparities, and provide evidence to back up policy decisions to change the status quo.

Acknowledgments

Thanks to Jonathan Borchardt, Matt Gardner, and Candace Ross for the initial analysis that motivated this work. Thanks to Kyle Lo for methodological discussions and Ashish Sabharwal, Maarten Sap, Noah Smith, and Mark Yatskar for helpful comments. **C**

References

1. *Gender in the Global Research Landscape*. Technical Report. Elsevier, 2017. <https://www.elsevier.com/research-intelligence/campaigns/gender-17>.
2. Ammar, W., et al. Construction of the literature graph in semantic scholar. In *NAACL-HLT* (Orleans, Louisiana, June 1–6, 2018).
3. Bendels, M.H.K., Mueller, R., Brueggmann, D., Groneberg, D.A. Gender disparities in high-quality research revealed by Nature Index journals. *PLOS One* 13, 1 (2018), e0189136. <https://doi.org/10.1371/journal.pone.0189136>.
4. Box, G.E.P., Jenkins, G.M., Reinsel, G.C. *Time Series Analysis: Forecasting and Control*, 3rd edn. Prentice Hall, Englewood Cliffs, N.J., 1994.
5. Boynton, J.R., Georgiou, K., Reid, M., Govus, A. Gender bias in publishing. *The Lancet* 392, 10157 (2018), 1514–1515. [https://doi.org/10.1016/S0140-6736\(18\)32000-2](https://doi.org/10.1016/S0140-6736(18)32000-2).
6. Ceci, S.J., Williams, W.M. Understanding current causes of women's underrepresentation in science. *Proc. National Acad. Sci.* 108, 8 (2011), 3157–3162. <https://doi.org/10.1073/pnas.1014871108>.
7. Clifton, S.M., Hill, K., Karamchandani, A.J., Autry, E.A., McMahon, P.J., Sun, G. Mathematical model of gender bias and homophily in professional hierarchies. *Chaos* 29, 2 (2019), 023135. <https://doi.org/10.1063/1.5066450>.
8. Cohoon, J.M., Nigai, S., Kaye, J. Gender and computing conference articles. *Commun. ACM* 54, (2011), 72–80. <https://doi.org/10.1145/1978542.1978561>.
9. Decastro, R., Griffith, K.A., Ubel, P.A., Stewart, A.J., Jaggi, R. Mentoring and the career satisfaction of male and female academic medical faculty. *Acad. Med.: J. Assoc. Am. Med. Colleges* 89, 2 (2014), 301–311. <https://doi.org/10.1097/ACM.000000000000109>.
10. Feldon, D.F., Peugh, J.L., Maher, M.A., Roksa, M.A., Tofel-Grehl, C. Time-to-credit gender inequities of first-year PhD students in the biological sciences. *CBE Life Sci. Educ.* 16, 1 (2017), ar4. <https://doi.org/10.1187/cbe.16-08-0237>.
11. Fernandes, J.M., Monteiro, M.P. Evolution in the number of authors of computer science publications. *Scientometrics* 110, (2016), 529–539.
12. Hengel, E. 2017. Publishing while female. Are women held to higher standards? Evidence from peer review. In *Cambridge Working Article Economics*, 2017, 1753. Faculty of Economics, University of Cambridge. <https://doi.org/10.17863/CAM.17548>.

13. Holman, L., Stuart-Fox, D., Hauser, C.E. The gender gap in science: How long until women are equally represented? *PLOS Biol.* 16, 4 (2018), e2004956. <https://doi.org/10.1371/journal.pbio.2004956>.
14. Hyndman, R.J., Khandakar, Y. Automatic time series forecasting: The forecast package for R. *J. Statistical Software* 26, 3 (2008), 1–22. <https://doi.org/10.18637/jss.v027.i03>.
15. Karimi, F., Wagner, C., Lemmerich, F., Jadidi, M., Strohmaier, M. Inferring gender from names on the web: A comparative evaluation of gender detection methods. In *WWW* (Montreal, Canada, April 11 to 15, 2016). <https://doi.org/10.1145/2872518.2889385>.
16. Lundine, J., Bourgeault, I.L., Clark, J., Heidari, S., Balabanova, D. The gendered system of academic publishing. *The Lancet* 391, 10132 (2018), 1754–1756. [https://doi.org/10.1016/S0140-6736\(18\)30950-4](https://doi.org/10.1016/S0140-6736(18)30950-4).
17. Lundine, J., Bourgeault, I.L., Clark, J., Heidari, S., Balabanova, D. Gender bias in academia. *The Lancet* 393, 10173 (2019), 741–743. [https://doi.org/10.1016/S0140-6736\(19\)30281-8](https://doi.org/10.1016/S0140-6736(19)30281-8).
18. Mattauch, S., Lohmann, K., Hannig, F., Lohmann, D., Teich, J. A bibliometric approach for detecting the gender gap in computer science. *Commun. ACM* 63, (2020), 74–80.
19. Mohammad, S.M. Gender gap in natural language processing research: Disparities in authorship and citations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020). Association for Computational Linguistics, Online, 7860–7870. <https://doi.org/10.18653/v1/2020.acl-main.702>.
20. Moss-Racusin, C.A., Dovidio, J.F., Brescoll, V.L., Graham, M.J., Handelsman, J. Science faculty's subtle gender biases favor male students. *Proc. National Acad. Sci. U.S.A.* 109, 41 (2012), 16474–16479. <https://doi.org/10.1073/pnas.1211286109>.
21. Robnett, R.D. Gender bias in STEM fields: variation in prevalence and links to STEM self-concept. *Psychol. Women Q.* 40, 1(2016), 65–79. <https://doi.org/10.1177/0361684315596162>.
22. Santamaría, L.P., Mihaljević, H. Comparison and benchmark of name-to-gender inference services. *Peer J. Comput. Sci.* 4, (2018), e156. <https://doi.org/10.7717/peerj-cs.156>.
23. Schluter, N. The glass ceiling in NLP. In *EMNLP* (Brussels, Belgium from October 31st to November 4th, 2018). <https://doi.org/10.18653/v1/D18-1301>.
24. Shapiro, S.S., Wilk, M.B. An analysis of variance test for normality (complete samples). *Biometrika* 52, 3–4 (1965), 591–611. <https://doi.org/10.1093/biomet/52.3-4.591>.
25. Shen, Z., Ma, H., Wang, K. A Web-scale system for scientific knowledge exploration. In *Proceedings of ACL 2018, System Demonstrations* (2018), Association for Computational Linguistics, Melbourne, Australia, 87–92. <https://doi.org/10.18653/v1/P18-4015>.
26. Way, S.F., Larremore, D.B., Clauset, A. Gender, productivity, and prestige in computer science faculty hiring networks. In *WWW* (Montreal, Canada, April 11–15, 2016). <https://doi.org/10.1145/2872427.2883073>.
27. West, J.D., Jacquet, J., King, M.M., Correll, S.J., Bergstrom, C.T. The role of gender in scholarly authorship. *PLOS One* 8, 7 (2013), e66212. <https://doi.org/10.1371/journal.pone.0066212>.
28. Williams, W.M., Ceci, S.J. National hiring experiments reveal 2:1 faculty preference for women on STEM tenure track. *Proceedings of the National Academy of Sciences of the United States of America* 112, 17 (2015), 5360–5365. <https://doi.org/10.1073/pnas.1418878112>.

Lucy Lu Wang (lucyw@allenai.org) is a Postdoctoral Young Investigator at the Allen Institute for Artificial Intelligence, Seattle, WA, USA.

Gabriel Stanovsky (gabriel.stanovsky@mail.huji.ac.il) is a Senior Lecturer in the School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel. This work was done while at the Allen Institute for Artificial Intelligence and the University of Washington.

Luca Weihs (lucaw@allenai.org) is a Research Scientist at the Allen Institute for Artificial Intelligence, Seattle, WA, USA.

Oren Etzioni (orene@allenai.org) is Chief Executive Officer at the Allen Institute for Artificial Intelligence, Seattle, WA, USA.

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

Digital Threats: Research and Practice (DTRAP)

Open for
Submissions

A peer-reviewed journal that targets
the prevention, identification, mitigation,
and elimination of digital threats



Digital Threats: Research and Practice (DTRAP) is a peer-reviewed journal that targets the prevention, identification, mitigation, and elimination of digital threats. DTRAP aims to bridge the gap between academic research and industry practice. Accordingly, the journal welcomes manuscripts that address extant digital threats, rather than laboratory models of potential threats, and presents reproducible results pertaining to real-world threats.

DTRAP invites researchers and practitioners to submit manuscripts that present scientific observations about the identification, prevention, mitigation, and elimination of digital threats in all areas, including computer hardware, software, networks, robots, industrial automation, firmware, digital devices, etc. For articles involving analysis, the journal requires the use of relevant data and the demonstration of the importance of the results. For articles involving the results of structured observation such as experimentation and case studies, the journal requires explicit inclusion of rigorous practices; for example, experiments should clearly describe why internal validity, external validity, containment, and transparency hold for the experiment described.

For more
information
and to submit
your work,
please visit:

dtrap.acm.org



Association for
Computing Machinery

The evolution of and countermeasures for ...

BY WOJCIECH MAZURCZYK AND LUCA CAVIGLIONE

Cyber Reconnaissance Techniques

ALMOST EVERY DAY, security firms and mass media report news about successful cyber attacks, which are growing in terms of complexity and volume. According to Industry Week, in 2018 spear-phishing and spoofing attempts of business emails increased of 70% and 250%, respectively, and ransomware campaigns targeting enterprises had an impressive 350% growth.¹⁹ In general, economic damages are relevant, as there is the need of detecting and investigating the attack as well as restoring the compromised hardware and software.¹⁵ To give an idea of the impact of the problem, the average cost of a data breach has risen from \$4.9 million in 2017 to \$7.5 million in 2018.¹⁹ To make things worse, attackers can now use a wide range of tools for compromising hosts, network appliances and Internet of Things (IoT) devices in a simple and effective manner, for example, via a Crime-as-a-Service business model.¹¹

Usually, each cyber threat has its own degree of sophistication and not every attack has the same goal, impact, or extension. However, the literature agrees

that an attack can be decomposed into some general phases as depicted in Figure 1. As shown, the Tao of Network Security Monitoring subdivides the attacks in to five stages⁶ and the Cyber Kill Chain in to seven stages,²⁶ whereas the ATT&CK framework proposes a more fine-grained partitioning.²⁷ Despite the reference model, the first step always requires gathering information on the target and it is commonly defined as “reconnaissance.” Its ultimate goals are the identification of weak points of the targeted system and the setup of an effective attack plan.

In general, reconnaissance relies upon a composite set of techniques and processes and has not to be considered limited to information characterizing the target at a technological level, such as, the used hardware or the version of software components. Attackers also aim at collecting details related to the physical location of the victim, phone numbers, names of the people working in the targeted organizations and their email addresses. In fact, any bit of knowledge may be used to develop a software exploit or to reveal weaknesses in the defensive systems.

Unfortunately, the evolution of the Internet, the diffusion of online social networks, as well as the rise of services for scanning smart appliances and IoT

» key insights

- An attack can be decomposed into some general phases. The first step always requires gathering information on the target, a.k.a. “reconnaissance.”
- There is a plethora of reconnaissance techniques available for an attacker and many of them do not even require a direct contact with the targeted victim.
- Counteracting reconnaissance attempts must be viewed within the framework of the “arms race” between attackers and defenders.
- Defenders appear to be a step back with respect to attackers. Countermeasures should aim to: strengthen training, enforce proactive approaches, explore cyber deception as a defense tool, engineer reconnaissance-proof-by design services, and rethink the privacy concept.



Figure 1. The most popular reference models used to decompose a cyber attack into phases.



nodes, lead to an explosion of sources that can make the reconnaissance phase quicker, easier, and more effective. This could also prevent contact with the victim or limit its duration, thus making it more difficult to detect early and block reconnaissance attempts. Therefore, investigating the evolution of techniques used for cyber reconnaissance is of paramount importance to deploy or engineer effective countermeasures. Even if the literature provides some surveys on some specific aspects of reconnaissance (see, for example, network scanning⁸ and techniques exploiting social engineering³¹) the knowledge is highly fragmented and a comprehensive review is missing. In this perspective, this paper provides a “horizontal” review of the existing reconnaissance techniques and countermeasures, while highlighting emerging trends.

In this article, we introduce the classification and the evolution of the most popular reconnaissance methods. Then, we discuss possible countermeasures and present some future directions.

Classification and Evolution

In order to illustrate the most important cyber reconnaissance techniques and portray their evolution, we introduce the following taxonomy composed of four classes:

- *Social Engineering*: It groups methods for collecting information to deceive a person or convincing him/her to behave in a desired manner.

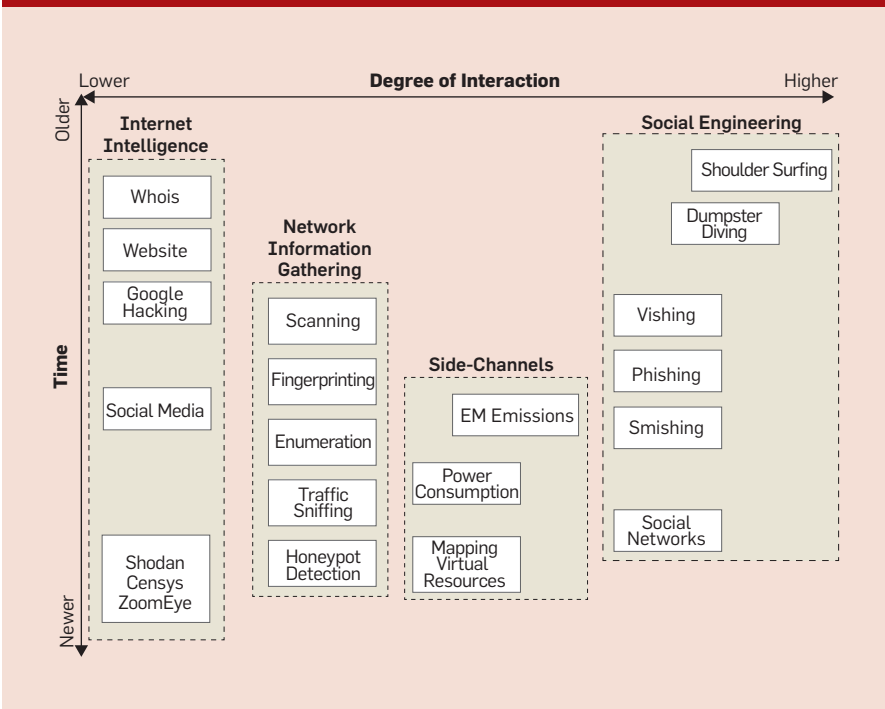
- *Internet Intelligence*: It groups methods taking advantage of information publicly available in the Internet including databases accessible via the Web.

- *Network Information Gathering*: It groups methods for mapping the network (or computing) infrastructure of the victim.

- *Side-Channels*: It groups methods exploiting unintended information leaked by the victim.

Each class accounts for a given “degree of interaction” with the victim, with the wide acceptance of how tight the coupling with the source of information should be for the purpose of the reconnaissance. For instance, reading the computer screen requires to be near the victim, thus potentially having a physical interaction, whereas scanning his/her network can be done

Figure 2. Classification of the reconnaissance techniques and their organization according to the time of appearance and the required degree of interaction with the victim.



remotely. In addition, some side-channels exploit a measurement that entails to be physically in a proximity to the target (for example, to measure the intensity of an electromagnetic field or the temperature of a heat source), while retrieving data from a social network does not require interacting with an asset run or owned by the victim itself.

Clearly, planning sophisticated attack campaigns or bypassing multiple security perimeters (for example, virtualized services deployed within a De-Militarized Zone) could require combining methodologies belonging to different classes. The longer the attacker actively interacts with the target, the higher the chance the attempt could be detected and neutralized. Unfortunately, the advent of social media, the progressive digitalization of many processes and workflows (as it happens in Industry 4.0 or in the smart-* paradigm), as well as the increasing pervasive nature of search engines, make the collection of data quicker and more effective. In this vein, Figure 2 proposes the taxonomy of reconnaissance techniques and it also emphasizes their temporal evolution and the required degree of interaction with the victim. We underline that the figure is intended to locate in time when methods firstly appeared and not how long they have been used (actually, the majority still is in the toolbox of attackers).

We now review the most important reconnaissance techniques proposed in the literature and observed in the wild, which are summarized and further commented in the sidebar “Examples of Reconnaissance Techniques and Sources.”

Social engineering is probably the oldest family of techniques used for reconnaissance and it is extraordinarily effective as it exploits the weakest link in security: humans. In essence, social engineering tries to manipulate and deceive victims by misusing their trust and convince them to share confidential information or to perform activities that can be useful to the attacker, for example, download and install a keylogger. It can also significantly decrease the time needed to gather information and often requires minimal or none technical skills.³¹ The literature

Examples of Reconnaissance Techniques and Sources

Social Engineering

Shoulder surfing: techniques where the attacker tries to determine confidential data by looking over the shoulders of the victim.

Dumpster diving: the practice of obtaining information from discarded material, such as documents, components of computing devices like hard drives and memory cards.

Phishing/Vishing/Smishing: the attacker tries to mislead the victim by impersonating a trustworthy entity by using email, VoIP, and Short Message Service.

Social Networks: the attacker utilizes social networks (for example, Facebook, LinkedIn, and Twitter) for gathering personal data or persuading the victim to reveal sensitive information or accomplish certain actions.

Internet Intelligence

whois/rwhois: databases providing information about IP address range and Autonomous Systems used by the victim.

Website: HTML pages can contain a very large and composite set of data. For the case of corporate websites, available information concerns employees, contact details, position within the organization, just to mention some. Comments left in HTML are another valuable source of information.

Google Hacking (Google Dorking): techniques utilizing advanced operators of Google to reveal potential security vulnerabilities and/or configuration errors of hardware and devices managed by the victim.

Social Media: a source of reconnaissance data where an attacker can collect personal information about the victim in order to learn, for instance, his/her habits, hobbies, likes and dislikes, with the aim of creating a more complete profile of the targeted person.

Shodan/Censys/ZoomEye: specific search engines indexing detailed technical data about different types of devices and network appliances.

Network Information Gathering

(Port) Scanning: methods for probing devices to establish whether on the targeted host there are open ports and exploitable services.

(OS/application) Fingerprinting: techniques for recognizing the operating system and/or applications utilized on the targeted device. A host can be stimulated with certain network traffic and replies are analyzed to guess the OS and/or installed applications.

(Network/Device) Enumeration: the systematic process for discovering hosts/servers/devices within the targeted network that are publicly exposed by the victim.

Traffic Sniffing: an attacker infers information about the victim network by collecting (sniffing) traffic or via monitoring tools.

Honeypot Detection: a set of techniques allowing the attacker to recognize whether the compromised machine is real or virtual. Typically, such methods rely on the detailed analysis of the behavior of the breached host (execution delays) or network configurations (MAC address, ARP and RARP entries, and so on).

Side-Channels

EM Emissions/Power Consumption: side-channels can be used to infer the signals leaked from screens, printers, or keyboards, to retrieve sensitive information. The most relevant physical quantities observed to set the side-channels are electromagnetic emissions or the power consumption of targeted devices.

Mapping Virtual Resources: side-channels are used to map a cloud infrastructure in order to establish if services are virtualized/containerized or to perform other types of attacks like co-resident threats. Typically, this class of side-channels operates in a completely remote manner.


proposes several taxonomies for social engineering attacks,³¹ but the simplest subdivision considers two main groups: *human-based* requiring a direct or in person interaction, and *technology-based* where the physical presence of the attacker is not needed. Human-based techniques are the oldest and

include methods like impersonation, dumpster diving or shoulder surfing. Even if still used, technology-based mechanisms today appear to be more popular and include methods like phishing and spam, or for tricking the user to install malware by using pop-ups and ad hoc crafted email.


Another important goal of social engineering is to get information about the victim or enrich (uncomplete) bits of information gathered with other techniques, for example, compile a custom dictionary to force the password for a username/email observed on a website. With the high exposure of people to several communication channels and the variety of social media services, an attacker has a wide array of opportunities to craft reconnaissance campaigns, as he/she can use face-to-face, telephone/VoIP, or instant messaging services, as well as online scams and fake identity attacks on online social networks. Such risks are exacerbated by the Bring Your Own Device paradigm, which makes it more difficult for an enterprise to control laptops, phones, and smart devices of their employees or to enforce access rules to shared resources like workspaces, wikis, forums, and websites.

Internet intelligence. Searching for publicly available information in the Internet is probably the first step that any attacker performs. The number of sources that can now be queried makes it possible to retrieve a huge amount of apparently insignificant fragments, which can become very informative if properly combined. In this perspective, Internet intelligence is the “offensive” subgroup of Open Source INTelligence (OSINT) and it is specialized and limited to the information available on the Internet and its services, such as the Web, public databases, specialized scanning services to map IoT nodes, and geographical or geo-referenced sources. Fortunately, the General Data Protection Regulation partially mitigated such a risk since the access to many public databases within the European Union is restricted. Internet Intelligence can be also used to perform passive *footprinting*, that is, the collection of publicly available information to identify a hardware and/or software infrastructure. In the following, we will discuss the main usage trends of Internet Intelligence.

Web sources. The increasing pervasive nature of the Web and the evolution of search engines surely added new and powerful options into the toolbox of attackers. Typically, a reconnaissance campaign starts from the website of the victim. In this way, the



The increasing pervasive nature of the Web and the evolution of search engines surely added new and powerful options into the toolbox of attackers.



attacker can gather important data like employee names, email addresses, telephone numbers or the physical address of the target, which can be used to perform social engineering or drive other threats. Personal bits of information can be also “fused” and enriched with data collected from online social networks. For instance, upon retrieving the hierarchy of the company and the list of the employees, the attacker can move to Facebook or LinkedIn to launch phishing or vishing attacks.¹²

Search engines are central for Internet intelligence, since they can limit the interaction between the attacker and the victim, thus making the data gathering phase difficult to detect. For instance, the attacker might craft some scripts to perform screen scraping directly from a website close to the victim hence leaving some traces in the log of the Web server. However, cached versions of the webpage provided by services like Google or the Internet Archive^a can be used to avoid traces of the reconnaissance attempt.

Apart from details that can be retrieved in an organic manner from indexed pages (for example, hobbies, owned books and records or visited stores), search engines can be also used to perform more fine-grained intelligence activities. Google Hacking²³ is one of the most popular techniques and it exploits advanced operators to perform narrow and precise queries mainly to reveal security breaches or configuration errors. For instance, the attacker can use operators like “*inurl*” to search within URLs. Google can then be queried with “*inurl:/hp/device/this.lcdispacher*” to discover details on a printer model to reveal potential vulnerabilities or search for a pre-cooked exploit. Another possible reconnaissance mechanism mixes the aforementioned technique for “Googling the Internet,” that is, using search engines to gather information on endpoints involved in a communication without the need of collecting or analyzing network traffic.³⁸

Public databases and sources. The variety of public records available online is another important source of information. In fact, every IP address and

a <https://archive.org/web/>

domain name should be registered in a public database, which can also contain a contact address and a telephone number. Some hints on the “layout” of the network of the victim can be inferred without needing to directly scan hosts or appliances. By querying the American Registry for Internet Numbers,^b it is possible to obtain the complete block of IP addresses assigned to the target. The Domain Name System can provide a wealth of details on the adopted addressing scheme and naming strategy. Other sources used for reconnaissance are the *whois* and *rwwhois*,^c which can provide IP address blocks and details on the autonomous system of the victim.

Public scanning services. As hinted, a large part of the success of an attack depends on identifying vulnerabilities within the targeted network/system. Until few years ago, this required performing a direct scan toward hosts, network devices, and software components or being able to collect network traffic, for example, via sniffers. To mitigate the direct exposure, a possible technique uses a botnet of zombies, that is, a network of compromised hosts under the control of the attacker. Zombies can then be used as proxies.^{13,16} Even if this approach may prevent to trace back the source of the scan/attack, still the attempt can be spotted or hindered. In this vein, a recent trend changed the situation, especially if the reconnaissance campaign targets IoT devices or smart settings like Heating, Ventilation and Air Conditioning. In fact, the availability of tools like Shodan,^d Censys,^e and ZoomEye^f imposed a paradigm shift to reconnaissance. Roughly, such services automatically scan the whole IPv4 public addressing space in a distributed and random manner and offer the obtained knowledge (for example, used hardware, open ports, or types of service delivered) via search-engine-like interfaces or ad-hoc Application Programming Interfaces. See the sidebar “Example of Shodan Query and Related Intelligence” for an example usage of Shodan for Internet intelligence.

b www.arin.net

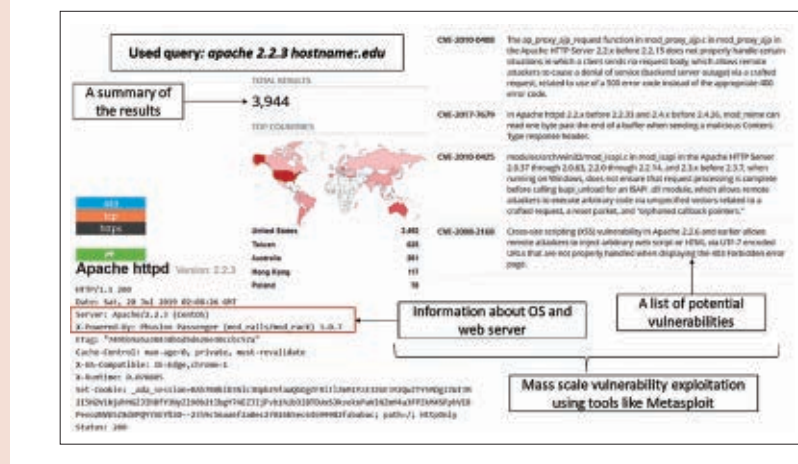
c whois.icann.org/en

d www.shodan.io

e censys.io

f www.zoomeye.org

Example of Shodan Query and Related Intelligence



Similarly to search engines used to index the Web, also in this case, attackers can gather data without directly contacting the targeted device and compile a list of potential targets/victims in a quick and easy manner: literature often defines this as “*contactless active reconnaissance*.”²⁹

A recent trend in contactless active reconnaissance combines different publicly available sources. As an example, data collected via Censys can be merged together with the National Vulnerability Database^g to improve the accuracy of discovering known vulnerabilities.²⁹

Network information gathering. When the data publicly available is not sufficient, the attacker needs to directly interact with the infrastructure of the victim. The most popular class of techniques is the one named “*network scanning*” and enables to map a remote network or identify the used operating systems and applications. Typically, network scanning techniques are divided in two main groups: passive and active.⁸ In passive scanning, the attacker infers information about the network by monitoring traffic. To this aim, *sniffers* can be deployed to capture and inspect flows, and the most popular tools are *tcpdump*^h and *Wireshark*.ⁱ This may also require

g nvd.nist.gov

h www.tcpdump.org

i www.wireshark.org

“mirroring” ports of a network appliance in order to duplicate the traffic. Instead, in active scanning, information is collected by intentionally generating and sending specific packets (also called *probes*) to the network device under investigation and by analyzing its responses. We point out that while performing scanning, attackers should stay “under the radar” to prevent detection due to anomalous traffic. For example, generating too many ICMP packets or incomplete TCP connections can be spotted with Intrusion Detection Systems (IDS) and firewalls.⁸ Scanning can be done at different levels of the protocol stack. Here, we present the most popular reconnaissance techniques for network information gathering grouped according to their scope.

Network and device enumeration. Two important parts of activities related to network information gathering, are *network enumeration* for discovering hosts and servers and *device enumeration* for identifying IoT nodes and other devices that are exposed by the victim. Despite using services like Shodan, the attacker may need to “manually” search for devices, for instance, due to the use of private IPv4 addressing schemes or to check the consistency of earlier information. The enumeration of network elements and devices is usually performed via

traffic analysis. However, the increasing diffusion of wireless technologies like WiFi, Bluetooth, and ZigBee to connect smart things like lightbulbs, various sensors, intelligent sockets and locks, makes the advent of a new form of techniques *à la* wardriving (that is, searching and marking for wireless signals for future exploitation). For instance, due to an improper configuration of wireless access points, the electromagnetic signal may be “leaking” outside the physical perimeter controlled by the victim, hence the malicious entity can expand his/her potential attack surface. For the purpose of scanning such a balkanized technological space, tools especially designed for IoT reconnaissance are becoming available. For instance,³³ proposing a passive tool for scanning multiple wireless technologies. An interesting idea is the use of the observed traffic to go beyond the enumeration of devices by classifying the type of the IoT node (for example, a camera or a smart speaker) and its state (for example, a smart switch is turned on or off). This can endow the attacker with very precise reconnaissance information.

Port scanning and fingerprinting. Methods defined as *port scanning* are designed to probe devices to determine whether there are open ports and exploitable services. Even if the literature abounds of methods, the most popular take advantage of the different behaviors of the three-way-handshake procedure of the TCP. Port scanning can then discover whether a remote TCP port is open by trying to send SYN/ACK packets, establishing a complete transport connection, or abort the process in the middle.⁸ Its main limitation is the need to maintain a large amount of TCP connections, thus causing transmission bottlenecks or exhausting the resources of the used machine. Consequently, the scanning rate is decreased and the reconnaissance attempt could be detected. A recent trend exploits distributed frameworks able to reduce both the scanning time and anomalous resource usages that could lead to identifying the attacker.²⁵

Scanning can be also used to recognize the guest OS or the applications available in the target nodes. This is

known as *fingerprinting* and may be implemented both via active and passive methods. For the case of OS fingerprinting, the main technique exploits the fact that the network stack of each OS exhibits minor differences when replying to well-crafted probe packets (for example, the initial sequence number of the TCP segments, the default TTL value for ICMP packets, among others).⁸ Such artifacts can be utilized to remotely determine the type and version of the OS of the inspected device. Application fingerprinting uses a slightly different technique. In this case, the attackers take advantage of a “banner,” which is a sort of preamble information that a server-side application sends before accepting a client. By stimulating a host with connection requests, they can harvest banners to reveal details on the active applications and services (for example, the version of the software can be used to determine the known vulnerabilities). A typical tool used for active scanning in network information gathering is *nmap*.^j

Application-level reconnaissance. The class of techniques named *application-level reconnaissance* is recently gaining attention, especially to infer some high-level features of the targeted host. To this aim, the attacker can utilize scanning tools to reveal certain weak points of the victim network. Possible examples of such tools are commercial suites like Nessus,^k Acunetix,^l and Vulners^m or opensource solutions like IVREⁿ and Vega.^o Another idea exploits probes to quantify the degree of protection of the victim. In this case, the attacker can use the timing of responses obtained to understand whether an antivirus is working on the targeted machine or if its signatures are updated.²

Honeypot detection. Honeypots are increasingly used to collect information on malware to organize suitable defense techniques or counterattacks, especially in case of botnets. From an attacker point of view, they represent a hazard since they can be

used to disseminate incorrect information, thus (partially) voiding the reconnaissance phase. Therefore, being able to detect confinement in a virtual/fictitious space is a core skill expected for the development of successful threats.²¹ To this aim, the attacker could check for the presence of TUN/TAP devices or specific entries in the ARP cache in order to have signatures to discriminate between real or virtual settings.¹⁷ Another mechanism takes advantage of the “fair” behavior of the honeypot, which impedes the node to harm a third-party entity. Thus, the attacker can try to compromise a host and launch some offensive patterns. According to the outcome, he/she can understand whether the node is real or fictitious.³⁹

Side-channels. Firstly envisaged by Lampson,²² the term *side-channel* usually defines attacks to deduce sensitive information by abusing unforeseen information leakages from computing devices. An interesting research direction started in the 1990s²⁰ with physical side-channels targeting cryptographic algorithms and their implementation. In essence, by inspecting apparently unrelated quantities, for example, the time needed to encrypt a message, the power consumed by a host or the electromagnetic field produced by the CPU of the device, attackers were able to infer information on the used algorithms and keys, thus making it feasible to exfiltrate encryption keys or conduct probabilistic guesses. Thus, in its original vision, a side-channel required a high degree of interaction with the victim.

This class of techniques is *en vogue* again especially for reconnaissance purposes. For instance, it has been proven that information or signals leaked from screens,¹⁴ printers,⁴ and keyboards⁷ can be used to retrieve login credentials or cryptographic keys. Other types of side-channels are becoming increasingly used, especially those allowing the attacker to control sensors located in close proximity of the target or to infer keyboard inputs on touchscreens,³⁴ for example, to exploit fingerprints left by user to guess the used unlock patterns or the PIN code.³ Owing to the high interconnected and virtual nature of modern hardware and software, side-channels attacks

j nmap.org

k www.tenable.com/products/nessus

l www.acunetix.com

m vulners.com

n ivre.rocks

o subgraph.com/vega

can be also operated in a completely remote manner, thus preventing the contact with the victim. For instance, they can be used to map a cloud infrastructure to understand whether services are virtualized or containerized or to perform cache-timing attacks.³⁰ To sum up, the use of a side-channel is a double-edged sword as it could require some physical proximity and this may increase the risk of exposure of the attacker, thus the value of the obtained information should be carefully evaluated.

Countermeasures

As has already happened in many other fields of cybersecurity, counteracting reconnaissance must be viewed within the framework of the “arms race” between attackers and defenders. Unfortunately, due to the availability of a composite amount of techniques, it is very difficult to completely prevent an attacker from inspecting a target. Over the years, countermeasures evolved and Figure 3 portrays a classification (also in this case, techniques have been located in the graph according to their estimated initial appearance).

As depicted, the evolution in the development of countermeasures experienced three main époques. In the earliest, the prime method aims at *training and raising awareness* of users as to reduce the effectiveness of social engineering or prevent the leakage of sensitive information. To complete this, constant auditing/monitoring campaigns of the information publicly available in the Internet should be performed on a regular basis. The paradigm shift happened when the design of countermeasures moved from considering primarily the technology rather than the human. The first wave deals with *reactive countermeasures* and aims at directly responding to a specific reconnaissance technique, for instance, scanning or sniffing. The more recent trend deals with *proactive countermeasures*: in this case, the attacker is disturbed or hindered on a constant basis, for example, by deliberately disseminating misleading data.

Human-based countermeasures. To mitigate the bulk of information that can be gathered via social engineering, including those available in

the Internet, an effective approach aims at reducing the impact of individuals by proper training and education.³⁵ Specifically, training may limit the exposure to social engineering techniques by explaining to users what kind of information can be publicly shared and how. Training can be also beneficial for technical staff that can learn the tools used by an attacker to reveal security breaches and design workarounds.

In parallel, security experts should perform public information monitoring on a continuous basis, that is, perform a sort of “protective” OSINT. Obtained data can be used again to instruct users and technicians. More importantly, public information monitoring can help assess the degree of security of the target, sanitize data leaks, as well as feed more sophisticated countermeasures.¹⁸

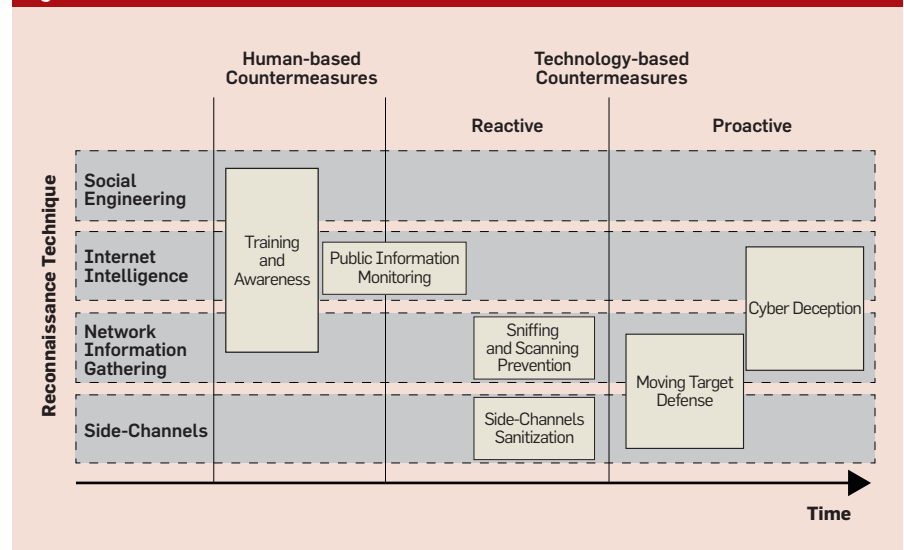
Reactive technology-based countermeasures. As hinted, reactive countermeasures are the direct response against reconnaissance attempts, including those exploiting side-channels. The main limitation of the approach is that if the threat evolves in time, the defensive mechanism has to be adjusted to stay effective. The review of the main reactive methods is as follows.

Sniffing and scanning prevention. The literature showcases several approaches to limit the ability of an attacker to sniff traffic for learning the configuration and the properties of the network.^{36,37} The common idea is to discover whether a wired/wireless

network interface card is set to promiscuous mode, that is, all the received frames are passed to the higher layers of the protocol stack despite the host is not the intended destination. To this aim, two main techniques exist: *challenge-based*³⁶ and *measurement-based*.³⁷ In challenge-based methods, the defender provokes a reply from the (supposed) sniffing machine by using ad-hoc crafted network traffic (typically, packets with a forged MAC address). In measurement-based methods, a host suspected to be controlled by the attacker is flooded with suitable traffic patterns. In both cases, the provided answer or its temporal evolution will help the defender to identify the reconnaissance attempt. Alas, the continuous development of hardware and OSs reduces the effectiveness of such techniques, mainly due to the need of having updated templates to compare the received traffic.⁹

Lastly, the advent of automatic and efficient scanning services like Shodan revamped the importance of carefully designing the addressing scheme to be used. In fact, IoT and smart devices could take advantage of IPv6 both in terms of end-to-end transparency and difficulties in performing a brute-force scan to the entire address space. However, IPv6 can directly expose portions of the network, thus the use of private IPv4 schemes jointly with Network Address Translation is a common and early front line defense technique.²⁸ Nevertheless, classical techniques (like firewalls and

Figure 3. Classification and evolution of the reconnaissance countermeasures.



IDS)⁸ should still be considered prime countermeasures against port scanning and fingerprinting attempts. Finally, recent approaches focus also on analyzing backscatter traffic, that is, network traffic generated by unallocated or unused IP addresses in a near real-time manner. Such methods can be used to identify reconnaissance campaigns in industrial control systems scenarios.¹⁰

Side-channels sanitization. To limit the exposure to side-channels, both hardware and software countermeasures have been proposed to “sanitize” the behaviors responsible for leaking data.³² Hardware mechanisms include, among others, techniques to limit the signal leakage by utilizing Faraday-cage-like packaging, minimize the number of metal parts of a component, or make the circuitry less power consuming to tame EM emissions. For the case of software countermeasures, we mention tools to randomize the sequences of operation or table lookups as well as mechanisms to avoid specific instructions patterns as to prevent the CPU/GPU radiating distinguishable EM patterns that act as a signature.

Side-channels also let attackers map virtual resources in cloud datacenters or honeypots. Since the cache architecture or the timing behaviors are often abused for this purpose, many countermeasures focus on modifying the underlying OS to introduce time-padding (to assure that execution time of a protected function is independent of any secret data the function operates on), cache cleansing (to forbid obtaining the state of the cache after running the sensitive function), and dynamic partitioning methods (to protect resources of a trusted process from being accessed by an untrusted process during its execution). Other possible countermeasures against side-channels can be deployed within the hypervisor or at the application level.⁵

Lastly, if side-channels are used to infer information through the network, a prime solution is to use some form of traffic normalization. In this case, ambiguities of the flow that can be exploited to infer data are removed by suitable manipulation of Protocol Data Units. For instance, the presence



Side-channels let attackers map virtual resources in cloud datacenters or honeypots.



of a firewall can be sensed by inspecting delay and the inter-packet time statistics, thus suitable buffering techniques could prevent to leak such an information.

Proactive technology-based countermeasures. Proactive solutions have been proposed to anticipate attackers by constantly shifting or poisoning the information that can be learned during the reconnaissance phase. The literature reports two main classes of techniques.

*Moving Target Defense (MTD)*²⁴ is a recent class of approaches aiming at recovering from the current asymmetry between attackers and defenders. In essence, MTD can limit the exposure of the victim by dynamically varying the configuration of network and nodes in order to make the leaked data unstable or outdated. The price to be paid is in terms of overheads experienced by the defender and legitimate users, for example, delays needed to change configurations and temporary device unavailability due to reassignments of addresses. As a possible example of production-quality MTD mechanisms, Dynamic Network Address Translation²⁴ allows interfering with malicious scanning phases by replacing TCP/IP header information while assuring service availability. Another method to protect cloud environments is to modify the scheduler to randomly allocate virtual machines and prevent co-residency attacks and side-channels between VMs running on the same physical machine.⁵

Cyber deception. Another emerging proactive cyber defense technology is Cyber Deception (CD). In this case, the defender provides to attackers misleading information in order to deceive them.⁴⁰ A possible approach deals with the manipulation of the network traffic to deliver the attacker a virtual, yet useless, network topology.¹ Differently from MTD, a mechanism based on CD does not continuously transform the defended deployment. Rather, it aims at distracting the attacker away from the most critical parts or to route and confine him/her within a honeypot or a honeynet. While a honeypot tries to lure the attacker into a single, deliberately vulnerable system, a honeynet

works on a larger scale by “simulating” a whole subnetwork. Thus, observing the attacker operating in such a strictly controlled environment allows to infer indicators of compromise that can be used both for anomaly detection purposes as well as to protect the real network from information gathering attempts.

Proactive countermeasures are expected to evolve into solutions able to combine CD and MTD approaches.⁴⁰ In such setups both techniques can be seen as complementary: MTD permits to adapt a system or a network to increase its diversity and complexity, while CD directs adversaries into time-consuming but pointless actions, thus draining their resources.

Conclusion and Outlook

This article has focused on the reconnaissance phase, which is the basis for the totality of cybersecurity attacks.


As a general trend, the evolution of smart devices, social media, and IoT-capable applications, boosted the amount of information that can be gathered by an attacker and also multiplied the communications paths that can be used to reach the victim. Therefore, the potential attack surface exploitable for reconnaissance techniques is expected to continue to grow, at least in the near future.

Regarding the development of countermeasures, defenders appear to be a step back with respect to attackers. To fill such gap, countermeasures should aim to:

- ▶ strengthen training and monitoring to also consider threats leveraging side-channels;
- ▶ evaluate how to incorporate results obtained via public sources into proactive countermeasures;
- ▶ expand solutions exploiting cyber deception also to counterattack social engineering (for example, when an employee detects a scam attempt, he/she intentionally mislead the attacker) and side-channels (for example, by deliberately leaking incorrect information);
- ▶ engineer a new-wave of reconnaissance-proof-by design services, for instance, by minimizing the impact of the addressing scheme, the use of IoT and the exposition to scanning services like Shodan; and,
- ▶ re-think the concept of privacy in

a more broad manner to also include protection mechanisms against advanced and malicious data gathering campaigns.

Acknowledgments

This work has been partially supported by EU Project SIMARGL, Grant Agreement No 833042 and by the Polish National Agency for Academic Exchange (Grant No PPN/BEK/2018/1/00153). 

References

1. Achleitner, S., La Porta, T., McDaniel, P., Sugrim, S., Krishnamurthy, S.V., Chadha, R. Cyber deception: Virtual networks to defend insider reconnaissance. In *Proceedings of the 8th ACM CCS Intern. Workshop on Managing Insider Security Threats*, Oct. 2016, 57–68.
2. Al-Saleh, M., Crandall, J.R. Application-level reconnaissance: Timing channel attacks against antivirus software. In *Proceedings of the 4th USENIX Conf. Large-scale Exploits and Emergent Threats*, 2011, 1–8.
3. Aviv, A., Gibson, K., Mossop, E., Blaze, M., Smith, J.M. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conf. on Offensive Technologies*, 2010, 1–7.
4. Backes, M., Dürmuth, M., Gerling, S., Pinkal, M., Spoleeder, C. Acoustic side-channel attacks on printers. In *Proceedings of the USENIX Security Symposium*, 2010, 307–322.
5. Bazm, M., M. Lacoste, M., M. Südholt, M. and J. Menaud, J. Side-channels beyond the cloud edge: New isolation threats and solutions. In *Proceedings of the 1st Cyber Security in Networking Conf.*, Oct. 2017, 1–8.
6. Bejtlich, R. The Tao of Network Security Monitoring Beyond Intrusion Detection. Pearson Education, 2004, ISBN: 0-321-24677-2.
7. Berger, Y., Wool, A., Yeredor, A. Dictionary attacks using keyboard acoustic emanations. In *Proceedings of the 13th ACM Conf. Computer and Communications Security*, 2006, 245–254.
8. Bou-Harb, E., Debbabi, M., Assi, C. Cyber scanning: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 16, 3 (3rdQ 2014), 1496–1519.
9. Cabaj, K., Gregorczyk, M., Mazurczyk, W., Nowakowski, P., Żorawski, P. Sniffing detection within the network: Revisiting existing and proposing novel approaches. In *Proceedings of the 5G Network Security Workshop to be held jointly with the 14th Intern. Conf. on Availability, Reliability and Security*, 2019.
10. Cabana, O., Youssef, A.M., Debbabi, M., Lebel, B., Kassouf, M., Agba, B.L. Detecting, fingerprinting and tracking reconnaissance campaignst industrial control systems. *Detection of Intrusions and Malware, and Vulnerability Assessment, LNCS 11543* (June 2019) . R. Perdisci, C. Maurice, G. Giacinto, M. Almgren (Eds.). Springer, 89–108.
11. Caviglione, L., Wendzel, S., Mazurczyk, W. The future of digital forensics: Challenges and the road ahead. *IEEE Security & Privacy* 15, 6, (Nov./Dec. 2017), 12–17.
12. Caviglione, L., Coccoli, M. Privacy problems with Web 2.0. *Computer Fraud & Security* 10 (2011), 16–19.
13. Collins, M., Shimeall, T., Faber, S., Janies, J., Weaver, R., Shon, M.D., Kadane, J. Using uncleanliness to predict future botnet addresses. In *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference*, 2007, 93–104.
14. Genkin, D., Pattani, M., Schuster, R., Tromer, E. Synesthesia: Detecting screen content via remote acoustic side channels. In *Proceedings of the IEEE Symp. Security & Privacy*, 2019.
15. Goodman, M. *Future Crimes*. Anchor Books, New York, 2016, ISBN 9780804171458.
16. Holz, T., Gorecki, C., Rieck, K., Freiling, F. Measuring and detecting fast-flux service networks. In *Proceedings of the 15th Network and Distributed System Security Symp.*, 2008, 257–268.
17. Holz, T., Raynal, F. Detecting Honey pots and Other Suspicious Environments. In *Proceedings of the 6th Annual IEEE SMC Information Assurance Workshop*, 2005, 29–36.
18. H2020 Project—Diversity Enhancements for Security Information and Event Management. Project Deliverable D4.1: Techniques and Tools for OSINT-based Threat Analysis; <http://disiem-project.eu/wp-content/uploads/2018/06/D4.1v2.pdf>
19. *Industry Week*. Cyberattacks skyrocketed in 2018. Are you ready for 2019?; <https://www.industryweek.com/technology-and-iiot/cyberattacks-skyrocketed-2018-are-you-ready-2019>
20. Kocher, P. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Proceedings of the Annual Intern. Cryptology Conf.* Springer, Berlin, Heidelberg, 1996, 104–113.
21. Krawetz, N. Anti-honeypot technology. *IEEE Security & Privacy* 2, 1 (Jan-Feb 2004), 76–79.
22. Lampson, B. A Note on the confinement problem. *Commun. ACM* 16, 10, (Oct. 1973), 613–615.
23. Lancor, L., Workman, R. Using Google hacking to enhance defense strategies. *ACM SIGCSE Bulletin*, 2007, 491–495.
24. Lei, C., Zhang, H.Q., Tan, J.L., Zhang, Y.C., Liu, X.H. Moving target defense techniques: A survey. *Security and Communication Networks* 2018, 1–25.
25. Li, Z., Yu, X., Wang, D., Liu, Y., Yin, H., He, S. SuperEye: A distributed port scanning system. *Artificial Intelligence and Security LNCS 11635*. X. Sun, Z. Pan, E. Bertino, (Eds). Springer, Cham, July 2019, 46–56.
26. Lockheed Martin. The Cyber Kill Chain; <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>
27. MITRE, ATT&CK Framework; <https://attack.mitre.org/>
28. Notra, S., Siddiqi, M., Gharakheili, H.H., Sivaraman, V., Boreli, R. An experimental study of security and privacy risks with emerging household appliances. In *Proceedings of the IEEE Conf. on Communications and Network Security*, 2014, 79–84.
29. O'Hare, J., Macfartane, R., Lo, O. Identifying Vulnerabilities Using Internet-Wide Scanning Data. In *Proceedings of the 12th IEEE Intern. Conference on Global Security, Safety and Sustainability*, pp. 1–10, 2019.
30. Ristenpart, T., Tromer, E., Shacham, H., Savage, S. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In *Proceedings 16th ACM Conf. Computer and Communications Security*, 2009, 199–212.
31. Salahdine, F., Kaabouch, N. Social engineering attacks: A survey. *Future Internet* 11, 4 (2019), 1–17.
32. Sayakarra, A., N.-A. L.-K., Scanton, M. A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. *Digital Investigation* 29 (2019), 43–54.
33. Siby, S., Maiti, R.R., Tippenhauer, N.O. IoTScanner: Detecting privacy threats in IoT neighborhoods. In *Proceedings of the 3rd ACM Intern. Workshop on IoT Privacy, Trust, and Security*, 2017, 23–30.
34. Simon, L., Xu, W., Anderson, R. Don't interrupt me while I type: Inferring text entered through gesture typing on Android keyboards. In *Proceedings of Privacy Enhancing Technologies* 3 (2016), 136–154.
35. Siponen, M. A Conceptual foundation for organizational information security awareness. *Information Management & Computer Security* 8, 1 (2000), 31–41.
36. Trabelsi, Z. and Rahmani, H. Detection of sniffers in an Ethernet network. *Information Security, LNCS 3225* (Sept. 2004). K. Zhang, Y. Zheng (Eds) Springer, Berlin, Heidelberg, 170–182.
37. Trabelsi, Z., Rahmani, H., Kaouech, K., Frikha, M. Malicious sniffing systems detection platform. In *Proceedings of the Intern. Symp.Applications and the Internet*, 2004, 201–207.
38. Trestian, I., Ranjan, S., Kuzmanovic, A., Nucci, A. Googling the Internet: Profiling Internet endpoints via the World Wide Web. *IEEE/ACM Trans. Networking* 18, 2 (2010), 666–679.
39. Wang, P., Wu, L., Cunningham, R., Zou, C.C. Honey pot detection in advanced botnet attacks. *Intern. J. Information and Computer Security* 4, 1 (2010), 30–51.
40. Wang, C., Lu, Z. Cyber deception: Overview and the road ahead. *IEEE Security & Privacy* 16, 2 (M-A 2018), 80–85.

Wojciech Mazurczyk is University Professor at Warsaw University of Technology, Institute of Computer Science, Warsaw, Poland.

Luca Caviglione is a senior research scientist at National Research Council of Italy, Institute for Applied Mathematics and Information Technologies, Genova, Italy.

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

Tracking the historical events that lead to the interweaving of data and knowledge.

BY CLAUDIO GUTIERREZ AND JUAN F. SEQUEDA

Knowledge Graphs

“Those who cannot remember the past are condemned to repeat it.”

—George Santayana

THE NOTION OF Knowledge Graph stems from scientific advancements in diverse research areas such as Semantic Web, databases, knowledge representation and reasoning, NLP, and machine learning, among others. The integration of ideas and techniques from such disparate disciplines presents a challenge to practitioners and researchers to know how current advances develop from, and are rooted in, early techniques.

Understanding the historical context and background of one’s research area is of utmost importance in order to understand the possible avenues of the future. Today, this is more important than ever due to the almost infinite sea of information one faces everyday. When it comes to the Knowledge Graph area, we have noticed that students and junior researchers are not completely aware of the source of the ideas, concepts, and techniques they command.

The essential elements involved in the notion of Knowledge Graphs can be traced to ancient history in the core idea of representing knowledge in a diagrammatic form. Examples include: Aristotle and visual forms of reasoning, around 350 BC; Lull and his tree of knowledge; Linnaeus and taxonomies of the natural world; and in the 19th. century, the works on formal and diagrammatic reasoning of scientists like J.J. Sylvester, Charles Peirce and Gottlob Frege. These ideas also involve several disciplines like mathematics, philosophy, linguistics, library sciences, and psychology, among others.

This article aims to provide historical context for the roots of Knowledge Graphs grounded in the advancements of the computer science disciplines of knowledge, data, and the combination thereof, and thus, focus on the developments after the advent of computing in its modern sense (1950s). To the best of our knowledge, we are not aware of an overview of the historical roots behind the notion of knowledge graphs. We hope that this article is a contribution in this direction. This is not a survey, thus, necessarily does not cover all aspects of the phenomena and does not do a systematic qualitative or quantitative analysis of papers and systems on the topic.

This article is the authors’ choice of a view of the history of the subject with

» key insights

- **Data was traditionally considered a material object, tied to bits, with no semantics per se. Knowledge was traditionally conceived as the immaterial object, living only in people’s minds and language. The destinies of data and knowledge became bound together, becoming almost inseparable, by the emergence of digital computing in the mid-20th century.**
- **Knowledge Graphs can be considered the coming of age of the integration of knowledge and data at large scale with heterogeneous formats.**
- **The next generation of researchers should become aware of these developments. Both successful and not, these ideas are the basis of current technology and contain fruitful ideas to inspire future research.**



a pedagogical emphasis directed particularly to young researchers. It presents a map and guidelines to navigate through the most relevant ideas, theories, and events that, from our perspective, have triggered current developments. The goal is to help understand what worked, what did not work, and reflect on how diverse events and results inspired future ideas.

For pedagogical considerations, we periodized the relevant ideas, techniques, and systems into five themes: Advent, Foundations, Coming-of-Age, Web Era, and Large Scale.

They follow a timeline, although with blurry boundaries. The presentation of each period is organized along two core ideas—data and knowledge—plus a discussion on data+knowledge showing their interplay. At the end of each section, we sketched a list of “realizations” (in both its senses—of becoming aware of something, as well as

achievements of something desired or anticipated), and “limitations” (or, impediments) of the period. The idea is to motivate a reflection on a balance of the period. At the end of each section we include a paragraph indicating references to historical and/or technical overviews on the topics covered.

Advent of the Digital Age

The beginnings are marked by the advent and spread of digital computers and the first programming languages (LISP, FORTRAN, COBOL, and ALGOL are among the most iconic) that gave rise to the digital processing of data at massive scale and the birth of a new area of science and technology, namely, computer science. The following are five relevant threads of this era:

1. Automation of reasoning. After the first program to process complex information, “Logic Theorist” by Newell, Shaw, and Simon in 1956,

they developed the “General Solving Program” in 1958, which illustrates well the paradigm researchers were after: “*this program is part of a research effort by the authors to understand the information processes that underlie human intellectual, adaptive, and creative abilities.*” And the goal was stated as follows: “*to construct computer programs that can solve problems requiring intelligence and adaptation, and to discover which varieties of these programs can be matched on human problem solving.*” This was continued by several other developments in the automation of reasoning, such as Robinson’s Resolution Principle³³ and Green and Raphael’s connection between theorem proving and deduction in databases by developing question-answering systems.¹⁴ At the practical level there were manifold implementations of “reasoning” features. An example is Joseph Weizenbaum’s

ELIZA, a program that could carry a dialogue in English on any topic, given it was programmed correctly.

2. Searching in spaces. Researchers recognized the process of searching in large spaces represented a form of “intelligence” or “reasoning.” Having an understanding of such space would ease searching. Sorting is a simple example. Easily 25% of computer time until the 1970s was used in sorting data to make feasible any search procedure.⁶ The very notion of search was well known to people working in data processing, even before the advent of computers. However, the idea of searching in diverse and complex spaces was new, such as search spaces arising in games (for example, chess, checkers, and Go). Dijkstra’s famous algorithm for finding shortest paths is from 1956, and its generalization A* is from 1968.¹⁹

3. Retrieving information from unstructured sources. Once having the computational capabilities, one can get data from sources beyond traditional structured data. The ideas go back to V. Bush’s report “As We May Think” but were developed systematically in the 1950s.¹¹ A milestone was Bertram Raphael’s “SIR: A Computer Program for Semantic Information Retrieval” (1964).³¹ This system demonstrated what could be called an ability to “understand” semantic information. It uses word associations and property lists for the relational information normally conveyed in conversational statements. A format-matching procedure extracts semantic content from English sentences.

4. Languages and systems to manage data. An early system to manage data was the Integrated Data Store (IDS) designed by Charles Bachman in 1963.² The IDS system maintained a collection of shared files on disk, had tools to structure and maintain them, and an application language to manipulate data. This allowed efficiency at the cost of what was later called “data independence.” IDS became the basis for the CODASYL standard, which became known as Database Management Systems (DBMS). Furthermore, the idea that there should be more dedicated languages to handle data led to the creation of COBOL (1959), which is an early example of a programming

language oriented to data handling and with a syntax resembling English.

5. Graphical representation of knowledge. Semantic networks were introduced in 1956 by Richard H. Richens, a botanist and computational linguist, as a tool in the area of machine translation of natural languages.³² The notion was developed independently by several people. Ross Quillian’s 1963 paper “A Notation for Representing Conceptual Information: An Application to Semantics and Mechanical English Paraphrasing” aimed at allowing information “to be stored and processed in a computer” following the model of human memory. The idea of searching for “design principles for a large memory that can enable it to serve as the base of knowledge underlying human-like language behavior” was further developed in his doctoral dissertation “Word concepts: A theory and simulation of some basic semantic capabilities” in 1967.²⁹

Sketch of realizations and limitations in the period. Among the realizations, the following stand out: The awareness of the importance and possibilities of automated reasoning; the problem of dealing with large search spaces; the need to understand natural language and other human representations of knowledge; the potential of semantic nets (and graphical representations in general) as abstraction layers; and the relevance of systems and high level languages to manage data. Regarding limitations, among the most salient were: the limited capabilities (physical and technical) of hardware; the availability and high cost of hardware; the gap between graphical representation and sequential implementation; and the gap between the logic of human language and the handling of data by computer systems.

Overview and secondary sources. For computing, P.E. Ceruzzi, *History of Modern Computing*; for the history of AI: N.J. Nilsson, *The Quest for Artificial Intelligence*.

Data and Knowledge Foundations

The 1970s witnessed much wider adoption of computing in industry. These are the years when companies such as Apple and Microsoft were founded. Data processing systems

such as Wordstar and VisiCalc, predecessors of current personal word processors and spreadsheets, were born. The increasing storage and processing power, as well as human expertise drove the need to improve how data should be managed for large companies.

Data. The growth in data processing needs brought a division of labor expressed in the notion of *representational independence*. Programmers and applications could now “forget” how the data was physically structured in order to access data. This idea is at the core of Edgar Codd’s paper “A Relational Model of Data for Large Shared Data Banks”⁸ that describes the use of relations as a mathematical model to provide representational independence; Codd calls this “data independence.” This theory and design philosophy fostered database management systems and modeling tools.

At the modeling level, Peter Chen introduced a graphical data model in his paper “The Entity-Relationship Model: Toward a Unified View of Data,”⁷ which advocated modeling data based on entities and relationships between them. Such ER models incorporated semantic information of the real world in the form of graphs. It is one of the early attempts to link a conceptual design with a data model—in this case the relational data model.

At the system level, software applications were developed and implemented to manage data based on the relational model, known as Relational Database Management Systems (RDBMS). Two key systems during this time were IBM’s System R, described in the paper “System R: Relational Approach to Database Management” (1976), and University of California at Berkeley’s INGRES, described in “The Design and Implementation of INGRES” (1976). These systems were the first to implement the “vision” of the relational model as described by Codd, including relational query languages such as SEQUEL and QUEL, which would lead to SQL, the most successful declarative query language in existence.

Knowledge. While the data stream was focusing on the structure of data and creating systems to best manage it, knowledge was focusing on the meaning of data. An early development in this direction was the work of


S.C. Shapiro who proposed a network data structure for organizing and retrieving semantic information.³⁴ These ideas were implemented in the semantic network and processing system (SNePS) that can be considered as one of the first stand-alone KRR systems.

In the mid-1970s, several critiques to semantic network structures emerged, focusing on their weak logical foundation. A representation of this criticism was William Woods' 1975 paper "What's in a Link: Foundations for Semantic Networks."⁴⁰


Researchers focused on extending semantic networks with formal semantics. An early approach to providing structure and extensibility to local and minute knowledge was the notion of *frames*. This was introduced by Marvin Minsky in his 1974 article "A Framework for Representing Knowledge."²⁷ A frame was defined as a network of nodes and relations. In 1976, John Sowa introduced Conceptual Graphs in his paper "Conceptual Graphs for a Data Base Interface."³⁶ Conceptual graphs serve as an intermediate language to map natural language queries and assertions to a relational database. The formalism represented a sorted logic with types for concepts and relations. In his 1977 paper "In Defense of Logic," Patrick Hayes recognized that frame networks could be formalized using first order logic.²⁰ This work would later influence Brachman and Levesque to identify a tractable subset of First-order logic, which would become the first development in Description Logics (see next section).

Data + Knowledge. In the 1970s, data and knowledge started to experience an integration. Robert Kowalski, in "Predicate Logic as Programming Language,"²³ introduced the use of logic as both a declarative and procedural representation of knowledge, a field now known as logic programming. These ideas were implemented by Alain Colmerauer in PROLOG.

Early systems that could reason based on knowledge, known as knowledge-based systems, and solve complex problems were expert systems. These systems encoded domain knowledge as if-then rules. R. Davis, B. Buchanan, and E. Shortliffe were among the first to develop a successful expert system, MYCIN, that became a classic



Conceptual graphs serve as an intermediate language to map natural language queries and assertions to a relational database.



example to select antibiotic therapy for bacteremia.¹⁰ An open problem was understanding where to obtain the data and knowledge. This area would be called knowledge acquisition.

The 1977 workshop on "Logic and Data Bases," held in Toulouse, France, and organized by Herve Gallaire, Jack Minker, and Jean-Marie Nicolas,¹³ is considered a landmark event. Important notions such as Closed World Assumption by Ray Reiter and Negation as Failure by Keith Clark were presented at this workshop, which can be considered the birth of the logical approach to data. Many researchers consider this to be the event that formalized the link between logic and databases, designating it as a field on its own.

Sketch of realizations and limitations in the period. Realizations of this period include: the need for and potential of representational independence, as shown by the case of the relational model; practical and successful implementations of the relational model; the realization that semantic networks require formal frameworks using the tools of formal logic; and the awareness of the potential of combining logic and data by means of networks. The limitations include: on the data side, the inflexibility of traditional data structures to represent new varieties of data (which gave rise to object-oriented and graph data structures); on the knowledge side, weakness of the logical formalization of common knowledge (which will be the motive of the rise of description logics).

Overview and secondary sources. On logic programming: A. Colmerauer and Ph. Roussel, *The Birth of Prolog*; R. Kowalski, *The Early Years of Logic Programming*. On knowledge representation: R.H. Brachman, H.J. Levesque, *Readings in Knowledge Representation*. On Expert Systems: F. Puppe, *Systematic introduction to Expert Systems*, Ch.1.

Coming-of-Age of Data and Knowledge


The 1980s saw the evolution of computing as it transitioned from industry to homes through the boom of personal computers. In the field of data management, the Relational Database industry was developing rapidly (Oracle, Sybase, IBM, among others). Object-oriented abstractions

were developed as a new form of representational independence. The Internet changed the way people communicated and exchanged information.


Data. Increasing computational power pushed the development of new computing fields and artifacts. These, in turn, generated complex data that needed to be managed. Furthermore, the relational revolution, which postulated the need of representational independence led to a separation of the software program from the data. This drove the need to find ways to combine object-oriented programming languages with databases. This gave rise to the development of object-oriented databases (OODB). This area of research investigated how to handle complex data by incorporating features that would become central in the future of data, such as objects, identifiers, relationships, inheritance, equality, and so on. Many systems from academia and industry flourished during this time, such as Encore-Observer (Brown University), EXODUS (University of Wisconsin–Madison), IRIS (Hewlett-Packard), ODE (Bell Labs), ORION (MCC), and Zeitgeist (Texas Instruments), which led to several commercial offerings.

Graphs started to be investigated as a representation for object-oriented data, graphical and visual interfaces, hypertext, etc. An early case was Harel's higraphs,¹⁸ which formalize relations in a visual structure, and are now widely used in UML. Alberto Mendelzon and his students developed the early graph query languages using recursion.⁹ This work would become the basis of modern graph query languages.

Knowledge. An important achievement in the 1980s was understanding the trade-off between the expressive power of a logic language and the computational complexity of reasoning tasks. Brachman and Levesque's paper "The Tractability of Subsumption in Frame-Based Description Languages" was among the first to highlight this issue.⁴ By increasing the expressive power in a logic language, the computational complexity increases. This led to research trade-offs along the expressivity continuum, giving rise to a new family of logics called *Description Logics*. Standout systems are



Increasing computational power pushed the development of new computing fields and artifacts. These, in turn, generated complex data that needed to be managed.



KL-ONE, LOOM, and CLASSIC, among others. In addition to Description Logic, another formalism was also being developed at that time: F-Logic was heavily influenced by objects and frames, allowing it to reason about schema and object structures within the same declarative language.²²

These early logic systems showed that logical reasoning could be implemented in tractable software. They would become the underpinning to OWL, the ontology language for the Semantic Web.

Additionally, the development of non-monotonic reasoning techniques occurred during this time, for example, the introduction of numerous formalisms for non-monotonic reasoning, including circumscription, default logic, autoepistemic logics and conditional logics.

Data + Knowledge. A relevant development in the 1980s was the Japanese 5th Generation Project.

Given Japan's success in the automotive and electronics industries, they were looking to succeed in software. The goal was to create artificial intelligence hardware and software that would combine logic and data and could carry on conversations, translate languages, interpret pictures, and reason like human beings. The Japanese adopted logic programming as a basis to combine logic and data.

The Japanese project sparked world wide activity leading to competing projects such as Microelectronics and Computer Technology Consortium (MCC) in the U.S., the European Computer Research Centre (ECRC) in Munich, and the Alvey Project in the U.K. MCC was an important research hub, both in hardware and software throughout the 1980s and 1990s. For example, the Cyc project, which came out of MCC, had the goal of creating the world's largest knowledge base of common sense to be used for applications performing human-like reasoning.

Expert systems proliferated in the 1980s and were at the center of the AI hype. We see the development of production rule systems such as OPS5, the Rete algorithm,¹² and Treat algorithm to efficiently implement rule-based systems. Expert systems were deployed on parallel computers, such as the DADO Parallel Computer, the Connection

Machine, and the PARKA Project, among others. Expert systems started to show business value (for example, Xcon, ACE). Venture capitalists started to invest in AI companies such as IntelliCorp, ILOG, Neuron Data, and Haley Systems, among others.

On the academic side, an initial approach of combining logic and data was to layer logic programming on top of relational databases. Given that logic programs specify functionality (“the what”) without specifying an algorithm (“the how”), optimization plays a key role and was considered much harder than the relational query optimization problem. This gave rise to deductive databases systems, which natively extended relational databases with recursive rules. Datalog, a subset of Prolog for relational data with a clean semantics, became the query language for deductive databases.⁵ One of the first deductive databases systems was the LDL system, presented in Tsur and Zaniolo’s paper “LDL: A Logic-Based Data-Language.”³⁷ Many of these ideas were manifested directly in relational databases known then as active databases.

At the beginning of the 1990s, expert systems proved expensive and difficult to update and maintain. It was hard to explain deductions, they were brittle, and limited to specific domains. Thus the IT world moved on and rolled that experience into mainstream IT tools from vendors such as IBM, SAP, and Oracle, among others. A decade after the start of the Japanese 5th Generation project, its original impressive list of goals had not been met. Funding dried out and these factors led to what has been called an AI Winter.

By the end of this decade, the first systematic study with the term “Knowledge Graph” appeared. It was the Ph.D. thesis of R.R. Bakker, “Knowledge Graphs: Representation and Structuring of Scientific Knowledge.” Many of these ideas were published later (1991) in a report authored by P. James (a name representing many researchers) and titled “Knowledge Graphs.”²¹ The term did not permeate widely until the second decade of the next century.

Sketch of realizations and limitations in the period. Among the most important realizations were the fact that the integration between logic and data must be

tightly coupled—that is, it is not enough to layer Prolog/expert systems on top of a database; and the relevance of the trade-off between expressive power of logical languages and the computational complexity of reasoning tasks. Two main limitations deserve to be highlighted: the fact that negation was a hard problem and was still not well understood at this time; and that reasoning at large scale was an insurmountable problem—in particular, hardware was not ready for the task. This would be known as the knowledge acquisition bottleneck.

Overview and secondary sources. On the golden years of graph databases, see R. Angles, C. Gutierrez, *Survey of Graph Database Models*. On O-O databases: M. Atkinson et al., *The Object-Oriented Database System Manifesto*. On the Japanese 5th Generation Project: E. Shapiro et al. *The 5th Generation Project: Personal Perspectives*.

Data, Knowledge, and the Web

The 1990s witnessed two phenomena that would change the world. First, the emergence of the World Wide Web, the global information infrastructure that revolutionized traditional data, information, and knowledge practices. The idea of a universal space of information where anybody could post and read, starting with text and images, in a distributed manner, changed completely the philosophy and practices of knowledge and data management. Second, the digitization of almost all aspects of our society. Everything started to move from paper to electronic. These phenomena paved the way to what is known today as Big Data. Both research and industry moved to these new areas of development.

Data. The database industry focused on developing and tuning RDBMS to address the demands posed by e-commerce popularized via the Web. This led to the generation of large amounts of data which were required to be integrated and analyzed. Research built on this momentum and focused on the areas of web data, data integration, data warehouse/OLAP, and data mining.

The data community moved toward the Web. Diverse efforts helped in developing an understanding of data and computations on the Web, shown

in papers such as “Formal Models of the Web” by Mendelzon and Milo²⁶ and “Queries and Computation on the Web” by Abiteboul and Vianu.¹ The Web triggered the need for distributing self-describing data. A key result of fulfilling these goals was semi-structured data models, such as Object Exchange Model (OEM), Extensible Markup Language (XML), and Resource Description Framework (RDF), among others.

During this time, organizations required integration of multiple, distributed, and heterogeneous data sources in order to make business decisions. Federated databases had started to address this problem in the 1980s (see survey³⁵). During this period, industry and academia joined forces and developed projects such as TSIMMIS and Lore from Stanford/IBM, SIMS from USC, InfoSleuth from MCC, among many others. These systems introduced the notion of mediators and wrappers.³⁹ Systems such as SIMS and InfoSleuth also introduced ontologies into the data integration mix.

In this context, due to the amount of data being generated and integrated, there was a need to drive business decision reporting. This gave rise to data warehouse systems with data modeled in star and snowflake schemas. These systems could support analytics on multi-dimensional data cubes, known as on-line analytical processing (OLAP). Much of the research focused on coming up with heuristics to implement query optimizations for data cubes. Business needs drove the development of data mining techniques to discover patterns in data.

Knowledge. Researchers realized that knowledge acquisition was the bottleneck to implement knowledge-based and expert systems. The Knowledge Acquisition Workshops (KAW in Canada and EKAW in Europe) were a series of events where researchers discussed the knowledge acquisition bottleneck problem. The topic evolved and grew into the fields of knowledge engineering and ontology engineering.

The Web was a realization that knowledge, not just data, should also be shared and reused. The need to elevate from administrative metadata to formal semantic descriptions gave rise to the spread of languages

to describe and reason over taxonomies and ontologies.

The notion of ontology was defined as a “shared and formal specification of a conceptualization” by Gruber.¹⁵

Among the first scientists arguing the relevance of ontologies were N. Guarino,¹⁶ M. Uschold, and M. Gruninger.³⁸ Research focused on methodologies to design and maintain ontologies, such as METHONLOGY, Knowledge Acquisition and Documentation Structuring (KADS) methodology, CommonKADS, and specialized methods such as OntoClean. We observe the emergence of the first ontology engineering tools (for example, Ontolingua, WebODE, and Protege) to help users code knowledge.

Data + Knowledge. The combination of data and knowledge in database management systems was manifested through *Deductive Databases*. Specialized workshops on Deductive Databases (1990–1999) and Knowledge Representation meets Databases (1994–2003) were a center for the activity of the field.³⁰ These developments led to refined versions of Datalog, such as probabilistic, disjunctive, and Datalog +/-.

An important challenge driving research was how to cope with formal reasoning at Web scale. In fact, viewing the Web as a universal space of data and knowledge, drove the need to develop languages for describing, querying and reasoning over this vast universe. The Semantic Web project is an endeavor to combine knowledge and data on the Web. The following developments influenced and framed the Semantic Web project: Simple HTML Ontology Extensions (SHOE), Ontobroker, Ontology Inference Layer (OIL) and DARPA Agent Markup Language (DAML), Knowledge Query and Manipulation Language (KQML), and the EU-funded Thematic Network OntoWeb (ontology-based information exchange for knowledge management and e-commerce) among others. The goal was to converge technologies such as knowledge representation, ontologies, logic, databases, and information retrieval on the Web. These developments gave rise to a new field of research and practice centered around the Web and its possibilities.

Sketch of realizations and limitations in the period. The main realization was that the Web was rapidly starting to change the ways the world of data, information and knowledge was traditionally conceived; new types of data were proliferating, particularly media data like images, video, and voice; and finally, the awareness that data must be—and now can be—connected to get value. Among the limitations is worth mentioning that the computational power was not enough to handle the new levels of data produced by the Web; and that the pure logical techniques have complexity bounds that made their scalability to certain growing areas like searching and pattern matching very difficult and at times infeasible.

Overview and secondary sources. About the Web: T. Berners-Lee, *Weaving the Web*. On data and the Web: S. Abiteboul et al., *Data on the Web: From Relations to Semistructured Data and XML*. On Ontology Engineering: R. Studer et al., *Knowledge Engineering: Principles and Methods*. On Web Ontology Languages: I. Horrocks et al., *From SHIQ and RDF to OWL: The Making of a Web Ontology Language*.

Data and Knowledge at Large Scale

The 2000s saw the explosion of e-commerce and online social networks (Facebook, Twitter, and so on). Advances in hardware and new systems made it possible to generate, store, process, manage, and analyze data at a much larger scale. We entered the Big Data revolution. During this era, we see the rise of statistical methods by the introduction of deep learning into AI.

Data. Web companies such as Google and Amazon pushed the barrier on data management.

Google introduced an infrastructure to process large amounts of data with MapReduce. The emergence of non-relational, distributed, data stores got a boom with systems such as CouchDB, Google Bigtable and Amazon Dynamo. This gave rise to “NoSQL” databases that (re-)popularized database management systems for Column, Document, Key-Value and Graph data models.

Many of the developments were triggered by the feasibility to handle and process formats like text, sound, imag-

es, and video. Speech and image recognition, image social networks like Flickr, advances in NLP, and so on consolidated the notion that “data” is well beyond tables of values.

The data management research community continued its research on data integration problems such as schema matching, entity linking, and XML processing. Database theory researchers studied data integration and data exchange from a foundational point of view.²⁵

Knowledge. The Description Logic research community continued to study trade-offs and define new profiles of logic for knowledge representation. Reasoning algorithms were implemented in software systems (for example, FACT, Hermit, Pellet). The results materialized as the European Ontology Inference Layer (OIL) DARPA Agent Markup Language (DAML) infrastructure. Both efforts joined forces and generated DAML+OIL, a thin ontology layer built on RDF with formal semantics based on description logics. This influenced the standardization of the Web Ontology Language (OWL) in 2004, which is a basis for the Semantic Web.

Big Data drove statistical applications to knowledge via machine learning and neural networks. Statistical techniques advanced applications that deduced new facts from already known facts. The 2012 work on image classification with deep convolutional neural networks with GPUs²⁴ is signaled as a result that initiated a new phase in AI: deep learning.

The original attempts in the 1960s to model knowledge directly through neural networks were working in practice. These techniques and systems now would outperform many human specific tasks such as classification, and applications where large amounts of training data and powerful hardware are available.

Data + Knowledge. The connection between data and knowledge was developed in this period along two lines, namely logical and statistical.

On the logical thread, the Semantic Web project was established, built upon previous results like the graph data model, description logics, and knowledge engineering.

The paper “The Semantic Web” by Tim Berners-Lee, Jim Hendler and Ora

Lassila³ sparked an excitement from industry and academia. The technologies underpinning the Semantic Web were being developed simultaneously by academia and industry through the World Wide Web Consortium (W3C) standardization efforts. These resulted in Resource Description Framework (RDF), Web Ontology Language (OWL), and SPARQL Protocol and RDF Query Language (SPARQL), among others.

In 2006, Tim Berners-Lee coined the term “Linked Data” to design a set of best practices highlighting the network structure of data on the Web in order to enhance knowledge.

This gave rise to the Linked Open Data (LOD) project and large RDF graph-based knowledge bases such as DBpedia, and Freebase, which would eventually lead to Wikidata. The LOD project was a demonstration of how data could be integrated at Web scale. In 2011, the major search engines released schema.org, a lightweight ontology, as a way to improve the semantic annotation of Web pages. These efforts were built on the results of the Semantic Web research community.

On the statistical thread, the beginning of the 21st century witnessed advances and successes in statistical techniques for large-scale data processing such as speech recognition, NLP, and image processing. This motivated Halvey, Norvig, and Pereira to speak of the “the unreasonable effectiveness of data.”¹⁷ This is probably one of the drivers that motivated the search for new forms of storing, managing and integrating data and knowledge in the world of Big Data and the emergence of the notion of Knowledge Graph. Furthermore, researchers have been making efforts to address statistical phenomena while incorporating techniques from logic and traditional databases such as statistical relational learning since the 1990s. Finally, it is relevant to highlight a new field dealing with data and knowledge that emerged under these influences: Data science.

Sketch of realizations and limitations in the period. Among the realizations in this period, we learned to think about data and knowledge in a much bigger way, namely at Web scale; and the world of data entered the era of neural networks due to new hardware and clever learning techniques. One of the



The beginning of the 21st century witnessed advances and successes in statistical techniques for large-scale data processing such as speech recognition, NLP, and image processing.



main limitations that made advances in this area difficult, is the fact that, although people realized the need to combine logical and statistical techniques, little is yet known on how to integrate these approaches. Another important limitation is that statistical methods, particularly in neural networks, still are opaque regarding explanation of their results.

Overview and secondary sources. D. Agrawal et al., *Challenges and Opportunities with Big Data*. T. Hey et al. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. R. Fagin et al. *Reasoning About Knowledge*.

Where Are We Now?

A noticeable phenomenon in the history we have sketched is the never-ending growth of data and knowledge, in both size and diversity. At the same time, an enormous diversity of ideas, theories, and techniques were being developed to deal with it. Sometimes they reached success and sometimes ended in failure, depending on physical and social constraints whose parameters most of the time were far out of the researcher’s control.

In this framework, historical accounts can be seen as a reminder that absolute success or failure does not exist, and that each idea, theory, or technique needs the right circumstances to develop its full potential. This is the case with the notion of Knowledge Graphs. In 2012, Google announced a product called the Google Knowledge Graph. Old ideas achieved worldwide popularity as technical limitations were overcome and it was adopted by large companies. In parallel, other types of “Graph” services were developed, as witnessed by similar ideas by other giants like Microsoft, Facebook, Amazon and Ebay.²⁸ Later, myriad companies and organizations started to use the Knowledge Graph keyword to refer to the integration of data, given rise to entities and relations forming graphs. Academia began to adopt this keyword to loosely designate systems that integrate data with some structure of graphs, a reincarnation of the Semantic Web, and Linked Data. In fact, today the notion of Knowledge Graph can be considered, more than a precise notion or system, an evolving project and a vision.

The ongoing area of Knowledge Graphs represents in this sense a convergence of data and knowledge techniques around the old notion of graphs or networks. From the data tradition, database technologies, and systems began to be developed by various companies and academia; manifold graph query languages are being developed: standard languages such as SPARQL and SPARQL 1.1, new industrial languages like Cypher, GSQL, and PGQL, research languages such as G-CORE, and the upcoming ISO standard GQL. On the other hand, we see a wealth of knowledge technologies addressing the graph model: on the logical side, the materialization and implementation of old ideas like semantic networks, and frames, or more recently, the Semantic Web and Linked Data projects; on the statistical side, techniques to extract, learn, and code knowledge from data on a large scale through knowledge graph embeddings.

It is not easy to predict the future, particularly the outcome of the interplay between data and knowledge, between statistics and logic. Today we are seeing a convergence of statistical and logical methods, with the former temporarily overshadowing the latter in the public eye. It is for this reason that we consider it relevant to call attention to history and “recover” the long-term significance of the achievements in the areas of data and knowledge. As we pointed out, even though some ideas and developments of the past may not have been successful or well known (or even known at all) at the time, they surely contain fruitful ideas to inspire and guide future research.

If we were to summarize in one paragraph the essence of the developments of the half century we have presented, it would be the following: Data was traditionally considered a commodity, moreover, a material commodity—something given, with no semantics per se, tied to formats, bits, matter. Knowledge traditionally was conceived as the paradigmatic “immaterial” object, living only in people’s minds and language. We have tried to show that since the second half of the 20th century, the destinies of data and knowledge became bound together by computing.

We have attempted to document how generations of computing scientists have developed ideas, techniques, and systems to provide material support for knowledge and to elevate data to the conceptual place it deserves.

Acknowledgments

This work was funded by ANID – Millennium Science Initiative Program – Code ICN17_002.

We reached out to many colleagues asking for their input on this article. We are extremely thankful for their helpful feedback: Harold Boley, Isabel Cruz, Jerome Euzenat, Dieter Fensel, Tim Finin, Enrico Franconi, Yolanda Gil, Joe Hellerstein, Jim Hendler, Jan Hidders, Ian Horrocks, Bob Kowalski, Georg Lausen, Leonid Libkin, Enrico Motta, Misty Nodine, Natasha Noy, Amit Sheth, Steffen Staab, Rudi Studer, Michael Uschold, Frank van Harmelen, Victor Vianu, Darrell Woelk, and Peter Wood. Juan thanks Daniel Miranker for inspiration on the topic of this article. We also thank Schloss Dagstuhl for hosting us in 2017 and 2018 to do this research and copyeditor Melinda O’Connell. **C**

References

- Abiteboul, S. and Vianu, V. Queries and computation on the Web. In *Proceedings of the 6th Intern. Conf. Database Theory*, 1997.
- Bachman, C.W. The origin of the integrated data store (IDS): The first direct-access DBMS. *IEEE Ann. Hist. Comput.* 31, 4 (Oct. 2009), 42–54.
- Berners-Lee, T., James Hendler, J. and Ora Lassila, O. The Semantic Web. *Sci. Amer.* 5 (May 2001), 34–43.
- Brachman, R.J. and Levesque, H.J. The tractability of subsumption in frame-based description languages. In *Proceedings of the Nat. Conf. Artificial Intelligence*. (Austin, TX, USA, Aug. 6–10, 1984), 34–37.
- Ceri, S., Gottlob, G., and Tanca, L. What you always wanted to know about datalog (and never dared to ask). *IEEE Trans. Knowl. Data Eng.* 1, 1 (1989), 146–166.
- Ceruzzi, P.E. *A History of Modern Computing* (2 ed.). MIT Press, Cambridge, MA, USA, 2003.
- Chen, P.P. The entity-relationship model—Toward a unified view of data. *ACM Trans. Database Syst.* 1, 1 (1976), 9–36.
- Codd, E.F. A relational model of data for large shared data banks. *Commun. ACM* 13, 6 (1970), 377–387.
- Cruz, I.F., Mendelzon, A.O. and Wood, P.T. A graphical query language supporting recursion. *SIGMOD*, 1987, 323–330.
- Davis, R., Buchanan, B., and Shortliffe, E. Production rules as a representation for a knowledge-based consultation program. *Artif. Intell.* 8, 1 (Feb. 1977), 15–45.
- Fairthorne, R.A. Automatic retrieval of recorded information. *Comput. J.* 1, 1 (Jan. 1958), 36–41.
- Forgy, C. Rete: A fast algorithm for the many patterns/many objects match problem. *Artif. Intell.* 19, 1 (1982), 17–37.
- Gallaire, H. and Minker, J. (Eds.). *Proceedings of the Symposium on Logic and Data Bases*, Centre d’études et de recherches de Toulouse, France, 1977.
- Green, C.C. and Raphael, B. The use of theorem-proving techniques in question-answering systems. In *Proceedings of the 1968 23rd ACM National Conf.*, 169–181.

- Gruber, T.R. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.* 43, 5-6 (Dec. 1995), 907–928.
- Guarino, N. Formal ontology, conceptual analysis and knowledge representation. *Int. J. Hum.-Comput. Stud.* 43, 5-6 (Dec. 1995), 625–640.
- Halevy, A.Y., Norvig, P. and Pereira, F. The unreasonable effectiveness of data. *IEEE Intell. Syst.* 24, 2 (2009), 8–12.
- Harel, D. On visual formalisms. *Commun. ACM* 31, 5 (1988), 514–530.
- Hart, P.E., Nilsson, N.J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Systems Science and Cybernetics* 4, 2 (1968), 100–107.
- Patrick J. Hayes. 1977. In *Defense of Logic*. In *IJCAI*. 559–565.
- James, P. Knowledge graphs. Number 945 in Memorandum Faculty of Applied Mathematics. University of Twente, Faculty of Applied Mathematics, 1991.
- Kifer, M., Lausen, G., and Wu, J. Logical foundations of object-oriented and frame-based languages. *J. ACM* 42, 4 (1995), 741–843.
- Kowalski, R.A. Predicate logic as programming language. In *Proceedings of the 6th IFIP Congress on Information Processing*, 1974, 569–574.
- Krizhevsky, A., Sutskever, I. and Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Proceedings of NIPS*.
- Lenzerini, M. Data integration: A theoretical perspective. In *Proceedings of PODS ’02*, 233–246.
- Mendelzon, A.O. and Milo, T. Formal models of Web queries. In *Proceedings of PODS ’97*, 134–143.
- Minsky, M. A Framework for Representing Knowledge. Technical Report, 1974, Cambridge, MA, USA.
- Noy, N.F., Gao, Y., Jain, A., Narayanan, A., Patterson, A., and Taylor, J. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* 62, 8 (Aug. 2019), 36–43.
- Quillian, R.M. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science* 12 (1967), 410–430.
- Ramakrishnan, R. and Ullman, J.D. A survey of deductive database systems. *J. Log. Program.* 23, 2 (1995), 125–149.
- B. Raphael, B. SIR: A Computer Program for Semantic Information Retrieval. Technical Report, 1964, Cambridge, MA, USA.
- Richens, R.H. Preprogramming for mechanical translation. *Mechanical Translation* 3, 1 (1956), 20–25.
- Robinson, J.A. A machine-oriented logic based on the resolution principle. *J. ACM* 12, 1 (1965), 23–41.
- Shapiro, S.C. 1971. A net structure for semantic information storage, deduction and retrieval. In *Proceedings of the 2nd Intern. Joint Conf. Artificial Intelligence*. (London, U.K., Sept. 1-3, 1971), 512–523.
- Sheth, A.P. and Larson, J.A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.* 22, 3 (Sept. 1990), 183–236.
- Sowa, J.F. Conceptual graphs for a data base interface. *IBM J. Research and Development* 20, 4 (1976), 336–357.
- Tsur, S. and Zaniolo, C. LDL: A logic-based data language. In *Proceedings of the 12th Intern. Conf. on Very Large Data Bases*, 1986, 33–41.
- Uschold, M. and Gruninger, M. Ontologies: Principles, methods and applications. *Knowledge Eng. Review* 11, 2 (1996), 93–136.
- Wiederhold, G. Mediation in information systems. *ACM Comput. Surv.* 27, 2 (June 1995), 265–267.
- Woods, W. What’s in a link: Foundations for semantic networks. 76 (Nov. 1975); <https://doi.org/10.1016/B978-1-4832-1446-7.50014-5>.

For an extended collection of references and resources, see the online appendix at <https://dl.acm.org/doi/10.1145/3418294>

Claudio Gutierrez (cgutierrez@dcc.uchile.cl) is a professor at the DCC, Universidad de Chile and IMFD.

Juan F. Sequeda (juan@data.world) is a principal scientist at data.world, Austin, TX, USA.

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

research highlights

P. 106

**Technical
Perspective**

Why Don't Today's Deep Nets Overfit to Their Training Data?

By Sanjeev Arora

P. 107

Understanding Deep Learning (Still) Requires Rethinking Generalization

By Chiyuan Zhang, Samy Bengio, Moritz Hardt,
Benjamin Recht, and Oriol Vinyals

P. 116

**Technical
Perspective**

Localizing Insects Outdoors

By Prabal Dutta

P. 117

3D Localization for Subcentimeter-Sized Devices

By Rajalakshmi Nandakumar, Vikram Iyer, and Shyamnath Gollakota

Technical Perspective

Why Don't Today's Deep Nets Overfit to Their Training Data?

By Sanjeev Arora

THE FOLLOWING ARTICLE by Zhang et al. is well-known for having highlighted that widespread success of deep learning in artificial intelligence brings with it a fundamental new theoretical challenge, specifically: *Why don't today's deep nets overfit to training data?* This question has come to animate the theory of deep learning.

Let's understand this question in context of *supervised learning*, where the machine's goal is to learn to provide labels to inputs (for example, learn to label cat pictures with "1" and dog pictures with "0"). Deep learning solves this task by training a net on a suitably large training set of images that have been labeled correctly by humans. The parameters of the net are randomly initialized and thereafter adjusted in many stages via the simplest algorithm imaginable: gradient descent on the current difference between desired output and actual output.

At the end of training, one usually finds that labels assigned by the net on the training images are mostly or entirely correct. Does this mean the net can be used to correctly label other pictures we will find on the Internet? Not necessarily. It is conceivable the net learned to correctly label just the training pictures, and no others. In other words, it could have *overfitted* to the training data. It is customary to check for this using a *holdout* set of training data that was left unused during training. The assumption underlying this methodology is that training data consists of independent samples from a fixed distribution, and we desire a net that gives correct labels to most images of the entire distribution. A simple probability concentration bound shows that performance on the holdout set is predictive—up to some well-defined error bars—of performance on the unseen images from the same distribution.

Received wisdom has it that overfitting happens if the net is *too expressive*, that is, has sufficient number of layers, and parameters per layer, than it is capable of expressing arbitrarily complicated mappings from inputs to 0/1 labels. To avoid overfitting, one should use a model that cannot "achieve more complicated functions than necessary." This philosophical principle is called *Occam's Razor* and related to the reasons why we prefer simpler scientific theories to complicated ones.


Decades of work in theory of machine learning and statistics has yielded measures of model complexity ranging from the old VC dimension and Rademacher complexity to more modern norm-based measures. This theory suggests that during training one must add a *regularizer* term to the training objective that penalizes models with a high measure of complexity.

Modern deep nets have turned out to confound this intuitive framework of regularizers. As the paper shows, it is possible to train nets with 50 million parameters using no regularizers on only 10,000 training examples. Surprisingly, no significant overfitting happens.

The extensive experiments detailed in the paper serve to deepen the mystery of this lack of overfitting. The experiments involve training nets on randomized/nonsensical versions of standard images datasets—the most benign being randomization of labels and more extreme being using random collections of pixels as images and random labels. Current deep nets—even with standard training and regularizer—are capable of achieving a good fit on these nonsensical datasets, which shows that these nets are capable of expressing very complicated functions. In particular, the experiment of fitting a net on images with random labels shows

that a traditional measure, Rademacher complexity, is high for the deep net architecture.

Subsequent work has explored the authors' suggestion that the training algorithm (a variant of gradient descent) plays a powerful role in how overfitting is avoided. Many new measures have been defined to measure the "effective number of parameters" of a trained net. Several of these measures were reported to correlate with good generalization. However, a recent extensive study² suggests this correlation is pretty weak and we still don't have a conclusive idea of why overfitting does not happen.

Another intriguing direction that has led to a flurry of papers is theoretical understanding of extreme over-parametrization. Since over-parametrization does not seem to hurt deep nets, it is natural to wonder if one can take it to the extreme. Recent work has analyzed the infinite limit: take a finite net and allow its width (= number of nodes for fully connected layers, and number of channels for convolutional layers) to go to infinity. This is the wonderful world of *Neural Tangent Kernels* or NTK.¹ Perhaps some of these new ideas will appear in the pages of *Communications* in future. Kudos to Zhang et al. for writing a paper that led to all this interesting follow-up work! 

References

1. Jacot, A. et al. Neural tangent kernel: Convergence and generalization in neural networks. In *Proceedings of the 32nd NeurIPS Conf.* (Montreal, Canada, 2018).
2. Neyshabur, B. et al. Towards understanding the role of over-parametrization in generalization of neural networks. In *Proceedings of ICLR*, 2019.

Sanjeev Arora is the Charles C. Fitzmorris Professor of Computer Science at Princeton University, Princeton, NJ, USA.

Copyright held by author.

Understanding Deep Learning (Still) Requires Rethinking Generalization

By Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals

Abstract

Despite their massive size, successful deep artificial neural networks can exhibit a remarkably small gap between training and test performance. Conventional wisdom attributes small generalization error either to properties of the model family or to the regularization techniques used during training.

Through extensive systematic experiments, we show how these traditional approaches fail to explain why large neural networks generalize well in practice. Specifically, our experiments establish that state-of-the-art convolutional networks for image classification trained with stochastic gradient methods easily fit a random labeling of the training data. This phenomenon is qualitatively unaffected by explicit regularization and occurs even if we replace the true images by completely unstructured random noise. We corroborate these experimental findings with a theoretical construction showing that simple depth two neural networks already have perfect finite sample expressivity as soon as the number of parameters exceeds the number of data points as it usually does in practice.

We interpret our experimental findings by comparison with traditional models.

We supplement this republication with a new section at the end summarizing recent progresses in the field since the original version of this paper.

1. INTRODUCTION

For centuries, scientists, policy makers, actuaries, and salesmen alike have exploited the empirical fact that unknown outcomes, be they future or unobserved, often trace regularities found in past observations. We call this idea generalization: finding rules consistent with available data that apply to instances we have yet to encounter.

Supervised machine learning builds on statistical tradition in how it formalizes the idea of generalization. We assume observations come from a fixed data generating process, such as samples drawn from a fixed distribution. In a first optimization step, called *training*, we fit a model to a set of data. In a second step, called *testing*, we judge the model by how well it performs on newly generated data from the very same process.

This notion of generalization as *test-time performance* can seem mundane. After all, it simply requires the model to achieve consistent success on the *same* data generating process as was encountered during training. Yet the seemingly simple question of what theory

underwrites the generalization ability of a model has occupied the machine learning research community for decades.

There are a variety of theories proposed to explain generalization.

Uniform convergence, margin theory, and algorithmic stability are but a few of the important conceptual tools to reason about generalization. Central to much theory are different notions of *model complexity*. Corresponding generalization bounds quantify how much data is needed as a function of a particular complexity measure. Despite much significant theoretical work, the prescriptive and descriptive value of these theories remains debated.

This work takes a step back. We do not offer any new theory of generalization. Rather, we offer a few simple experiments to interrogate the empirical import of different purported theories of generalization. With these experiments at hand, we broadly investigate what practices do and do not promote generalization, what does and does not measure generalization?

1.1. The randomization test

In our primary experiment, we create a copy of the training data where we replace each label independently by a random label chosen from the set of valid labels. A dog picture labeled “dog” might thus become a dog picture labeled “airplane”. The randomization breaks any relationship between the instance, for example, the image, and the label. We then run the learning algorithm both on the natural data and on the randomized data with identical settings and model choice. By design, no generalization is possible on the randomized data. After all, we fit the model against random labels!

For any purported measure of generalization, we can now compare how it fares on the natural data versus the randomized data. If it turns out to be the same in both cases, it could not possibly be a good measure of generalization for it cannot even distinguish learning from natural data (where generalization is possible) from learning on randomized data (where no generalization is possible). Our primary observation is:

Deep neural networks easily fit random labels.

The original version of this paper was published in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

More precisely, when trained on a completely random labeling of the true data, neural networks achieve 0 training error. The test error, of course, is no better than random chance as there is no correlation between the training labels and the test labels. In other words, by randomizing labels alone we can force the generalization error of a model to jump up considerably without changing the model, its size, hyperparameters, or the optimizer. We establish this fact for several different standard architectures trained on the CIFAR10 and ImageNet classification benchmarks. While simple to state, this observation has profound implications from a statistical learning perspective:

1. The effective capacity of neural networks is sufficient for memorizing the entire data set.
2. Even optimization on random labels remains easy. In fact, training time increases only by a small constant factor compared with training on the true labels.
3. Randomizing labels is solely a data transformation, leaving all other properties of the learning problem unchanged.

In particular, we find that many of the more popular explanations of generalization fail to capture what's happening in state-of-the-art deep learning models.

Extending on this first set of experiments, we also replace the true images by completely random pixels (e.g., Gaussian noise) and observe that convolutional neural networks continue to fit the data with zero training error. This shows that despite their structure, convolutional neural nets can fit random noise. We furthermore vary the amount of randomization, interpolating smoothly between the case of no noise and complete noise. This leads to a range of intermediate learning problems where there remains some level of signal in the labels. We observe a steady deterioration of the generalization error as we increase the noise level. This shows that neural networks are able to capture the remaining signal in the data while at the same time fit the noisy part using brute-force.

We discuss in further detail below how these observations rule out several standard generalization bounds as possible explanations for the generalization performance of state-of-the-art neural networks.

1.2. The role of regularization

Regularization can be thought of as the operational counterpart of a notion of model complexity. When the complexity of a model is very high, regularization introduces algorithmic tweaks intended to reward models of lower complexity. Regularization is a popular technique to make optimization problems “well posed”: when an infinite number of solutions agree with the data, regularization breaks ties in favor of the solution with lowest complexity.

Our second set of experiments interrogates the role that regularization plays in training overparameterized neural networks. Our experiments reveal that most of the regularization techniques in deep learning are not necessary for generalization: if we turn off the regularization parameters, test-time performance remains strong. Hence, explicit

regularization alone does not suffice to explain how deep learning models generalize. To summarize our finding:

Explicit regularization may improve generalization performance, but is neither necessary nor by itself sufficient for controlling generalization error.

While explicit regularizers like “dropout” and “weight-decay” may not be essential for generalization, it is certainly the case that not all models that fit the training data well generalize well. Indeed, in neural networks, we almost always choose our model as the output of running stochastic gradient descent. Appealing to linear models, we analyze how SGD acts as an implicit regularizer. For linear models, SGD always converges to a solution with small norm. Hence, the algorithm itself is implicitly regularizing the solution. Indeed, we show on small data sets that even Gaussian kernel methods can generalize well with no regularization. Though this does not explain why certain architectures generalize better than other architectures, it does suggest that more investigation is needed to understand exactly what the properties are inherited by models that were trained using SGD.

1.3. Finite sample expressivity

We complement our empirical observations with a theoretical construction showing that generically large neural networks can express any labeling of the training data. More formally, we exhibit a very simple two-layer ReLU network with $p = 2n + d$ parameters that can express any labeling of any sample of size n in d dimensions. A previous construction due to Livni et al.²² achieved a similar result with far more parameters, namely, $O(dn)$. While our depth-2 network inevitably has large width, we can also come up with a depth k network in which each layer has only $O(n/k)$ parameters.

While prior expressivity results focused on what functions neural nets can represent over the entire domain, we focus instead on the expressivity of neural nets with regard to a finite sample. In contrast to existing depth separations^{13, 15, 40, 11} in function space, our result shows that even depth-2 networks of linear size can already represent any labeling of the training data.

1.4. Related prior work

Below we discuss some related prior work. In Section 6.1, we discuss recent work that followed the initial publication of our work.

Barlett⁴ proved bounds on the fat shattering dimension of multilayer perceptrons with sigmoid activations in terms of the ℓ_1 -norm of the weights at each node. This important result gives a generalization bound for neural nets that is independent of the network size. However, for ReLU networks, the ℓ_1 -norm is no longer informative. This leads to the question of whether there is a different form of capacity control that bounds generalization error for large neural nets. This question was raised in a thought-provoking work by Neyshabur et al.,³⁰ who argued through experiments that network size is not the main form of capacity control for

neural networks. An analogy to matrix factorization illustrated the importance of implicit regularization.

Hardt et al.¹⁸ give an upper bound on the generalization error of a model trained with stochastic gradient descent in terms of the number of steps gradient descent took. Their analysis goes through the notion of *uniform stability*.⁸ As we point out in this work, uniform stability of a learning algorithm is independent of the labeling of the training data. Hence, the concept is not strong enough to distinguish between the models trained on the true labels (small generalization error) and models trained on the random labels (large generalization error). This also highlights why the analysis of Hardt et al.¹⁸ for nonconvex optimization was rather pessimistic, allowing only a very few passes over the data. Our results show that even empirically training neural networks is not uniformly stable for many passes over the data.

There has been much work on the representational power of neural networks, starting from universal approximation theorems for multi-layer perceptrons.^{12, 25, 13, 24, 15, 40, 11} All of these results are at the *population* level characterizing which mathematical functions certain families of neural networks can express over the entire domain. We instead study the representational power of neural networks for a finite sample of size n . This leads to a very simple proof that even $O(n)$ -sized two-layer perceptrons have universal finite-sample expressivity.

2. EFFECTIVE CAPACITY OF NEURAL NETWORKS

The size of a model family is often huge as it counts all possible functions in a certain set, including those that are unlikely to be found by the learning algorithm. By *effective capacity*, we informally refer to the size of the subset of models that is effectively achievable by the learning procedure. The capacity of this subset could be much smaller as it contains only “well-behaved” functions produced by some specific optimization algorithms, with bounded computation budget, and sometimes with explicit or implicit regularizations. Our goal is to understand the effective model capacity of feed-forward neural networks. Toward this goal, we choose a methodology inspired by nonparametric randomization tests. Specifically, we take a candidate architecture and train it both on the true data and on a copy of the data in which the true labels were replaced by random labels. In the second case, there is no longer any relationship between the instances and the class labels. As a result, learning is impossible. Intuition suggests that this impossibility should manifest itself clearly during training, for example, by training not converging or slowing down substantially. To our surprise, several properties of the training process for multiple standard architectures are largely unaffected by this transformation of the labels. This poses a conceptual challenge. Whatever justification we had for expecting a small generalization error to begin with must no longer apply to the case of random labels.

To gain further insight into this phenomenon, we experiment with different levels of randomization exploring the continuum between no label noise and completely corrupted labels. We also try out different randomizations of

the inputs (rather than labels), arriving at the same general conclusion.

The experiments are run on two image classification datasets, the CIFAR10 dataset and the ImageNet ILSVRC 2012 dataset. We test the *Inception V3* architecture on ImageNet and a smaller version of Inception, Alexnet, and MLPs on CIFAR10. Please see Appendix A of Zhang et al.⁴⁴ for more details of the experimental setup.

2.1. Fitting random labels and pixels

We run our experiments with the following modifications of the labels and input images:

- **True labels:** the original dataset without modification.
- **Partially corrupted labels:** independently with probability p , the label of each image is corrupted as a uniform random class.
- **Random labels:** all the labels are replaced with random ones.
- **Shuffled pixels:** a random permutation of the pixels is chosen and then *the same* permutation is applied to all the images in both training and test set.
- **Random pixels:** a different random permutation is applied to each image independently.
- **Gaussian:** A Gaussian distribution (with matching mean and variance to the original image dataset) is used to generate random pixels for each image.

Surprisingly, stochastic gradient descent with unchanged hyperparameter settings can optimize the weights to fit to random labels perfectly, even though the random labels completely destroy the relationship between images and labels. We further break the structure of the images by shuffling the image pixels, and even completely resampling the random pixels from a Gaussian distribution. But the networks we tested are still able to fit.

Figure 1a shows the learning curves of the Inception model on the CIFAR10 dataset under various settings. We expect the objective function to take longer to start decreasing on random labels because initially the label assignments for every training sample are uncorrelated. Therefore, large prediction errors are backpropagated to make large gradients for parameter updates. However, since the random labels are fixed and consistent across epochs, the network starts fitting after going through the training set multiple times. We find the following observations for fitting random labels very interesting: (a) we do not need to change the learning rate schedule; (b) once the fitting starts, it converges quickly; and (c) it converges to (over)fit the training set perfectly. Also note that “random pixels” and “Gaussian” start converging faster than “random labels.” This might be because with random pixels, the inputs are more separated from each other than natural images that originally belong to the same category, therefore, easier to build a network for arbitrary label assignments.

On the CIFAR10 dataset, Alexnet and MLPs all converge to zero loss on the training set. The shaded rows in Table 1 show the exact numbers and experimental setup. We also tested random labels on the ImageNet dataset. As shown

Figure 1. Fitting random labels and random pixels on CIFAR10. (a) The training loss of various experiment settings decaying with the training steps. (b) The relative convergence time with different label corruption ratio. (c) The test error (also the generalization error since training error is 0) under different label corruptions.

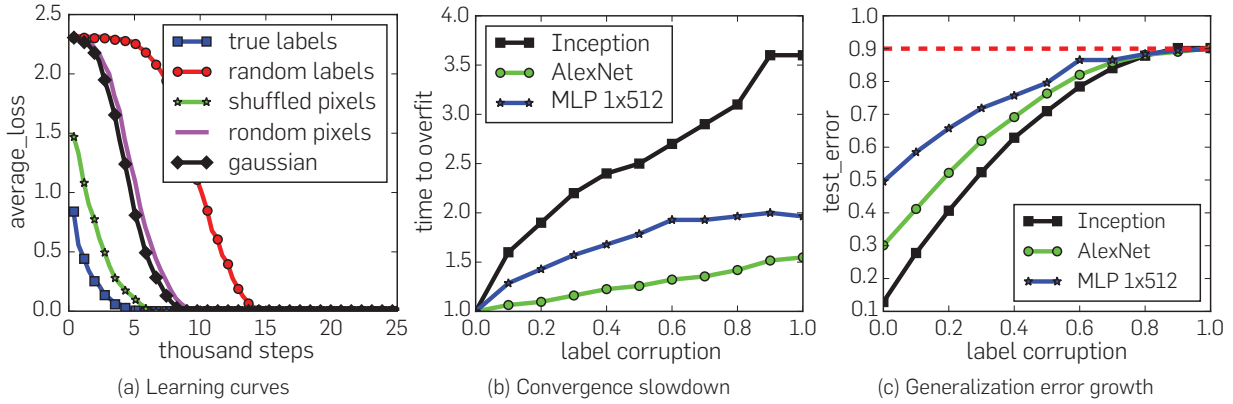


Table 1. The training and test accuracy (in %) of various models on the CIFAR10 dataset.

Model	# params	Random crop	Weight decay	Train accuracy	Test accuracy
Inception	1,649,402	Yes	Yes	100.0	89.05
		Yes	No	100.0	89.31
		No	Yes	100.0	86.03
		No	No	100.0	85.75
		(fitting random labels)	No	No	100.0
Inception w/o BatchNorm	1,649,402	No	Yes	100.0	83.00
		No	No	100.0	82.00
		(fitting random labels)	No	No	100.0
Alexnet	1,387,786	Yes	Yes	99.90	81.22
		Yes	No	99.82	79.66
		No	Yes	100.0	77.36
		No	No	100.0	76.07
		(fitting random labels)	No	No	99.82
MLP 3 × 512	1,735,178	No	Yes	100.0	53.35
		No	No	100.0	52.39
		(fitting random labels)	No	No	100.0
MLP 1 × 512	1,209,866	No	Yes	99.80	50.39
		No	No	100.0	50.51
		(fitting random labels)	No	No	99.34

Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

in the last three rows of Table 2 in Appendix of Zhang et al.,⁴⁴ although it does not reach the perfect 100% top-1 accuracy, 95.20% accuracy is still very surprising for 1.2 million random labels from 1000 categories. Note that we did not do any hyperparameter tuning when switching from the true labels to the random labels. It is likely that with some modification of the hyperparameters, perfect accuracy could be achieved on random labels. The network also manages to reach ~90% top-1 accuracy even with explicit regularizers turned on.

Partially corrupted labels. We further inspect the behavior of neural network training with a varying level of label corruptions from 0 (no corruption) to 1 (complete random labels) on the CIFAR10 dataset. The networks fit the corrupted training set perfectly for all the cases. Figure 1b shows the slowdown of the convergence time with increasing level of label noises. Figure 1c depicts the test errors after convergence. Since the training errors are always zero, the test errors are the same as generalization errors. As the noise level approaches 1, the generalization errors converge to 90%—the performance of random guessing on CIFAR10.

2.2. Implications

In light of our randomization experiments, we discuss how our findings pose a challenge for several traditional approaches for reasoning about generalization.

Rademacher complexity and VC-dimension. Rademacher complexity is commonly used and flexible complexity measure of a hypothesis class. The empirical Rademacher complexity of a function class \mathcal{F} on a dataset $\{z_1, \dots, z_n\}$ is defined as

$$\hat{\mathcal{R}}_n(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i f(z_i) \right] \quad (1)$$

where $\sigma_1, \dots, \sigma_n \in \{\pm 1\}$ are i.i.d. uniform random variables. Usually we aim to bound the Rademacher complexity of the loss function class $\mathcal{L} = \{\ell(z = (x, y)) = \ell(h(x), y) : h \in \mathcal{H}\}$, where $z_i = (x_i, y_i)$ are input-output pairs. For L -Lipschitz loss function ℓ and real valued hypothesis class \mathcal{H} , $\hat{\mathcal{R}}_n(\mathcal{L}) \leq L \hat{\mathcal{R}}_n(\mathcal{H})$ by contraction lemma. The Rademacher complexity measures the ability of a function class to fit random ± 1 binary

label assignments, which closely resemble our randomization test. Since our empirical results on randomization tests suggest that many neural networks fit the training set with random labels perfectly, we expect that $\hat{\mathfrak{R}}_n(\mathcal{L})$ approximately achieves the maximum for the corresponding loss class \mathcal{L} . For example, for the indicator loss, $\hat{\mathfrak{R}}_n(\mathcal{L}) \approx 1$. This is a trivial upper bound on the Rademacher complexity that does not lead to useful generalization bounds in realistic settings. A similar reasoning applies to VC-dimension and its continuous analog fat-shattering dimension, unless we further restrict the network. While Barlett⁴ proves a bound on the fat-shattering dimension in terms of ℓ_1 norm bounds on the weights of the network, this bound does not apply to the ReLU networks that we consider here. This result was generalized to other norms by Neyshabur et al.,³¹ but even these do not seem to explain the generalization behavior that we observe.

Uniform stability. Stepping away from complexity measures of the hypothesis class, we can instead consider properties of the algorithm used for training. This is commonly done with some notion of stability, such as *uniform stability*. Uniform stability of an algorithm A measures how sensitive the algorithm is to the replacement of a single example. However, it is solely a property of the algorithm, which does not take into account specifics of the data or the distribution of the labels. It is possible to define weaker notions of stability.^{27, 32, 36} The weakest stability measure is directly equivalent to bounding generalization error and does take the data into account. However, it has been difficult to utilize this weaker stability notion effectively.

3. THE ROLE OF REGULARIZATION

Most of our randomization tests are performed with explicit regularization turned off. Regularizers are the standard tool in theory and practice to mitigate overfitting in the regime when there are more parameters than data points.⁴² The basic idea is that although the original hypothesis is too large to generalize well, regularizers help confine learning to a subset of the hypothesis space with manageable complexity. By adding an explicit regularizer, say by penalizing the norm of the optimal solution, the effective Rademacher complexity of the possible solutions is dramatically reduced.

As we will see, in deep learning, explicit regularization seems to play a rather different role. As the bottom rows of Table 2 in Appendix of Zhang et al.⁴⁴ show, even with dropout and weight decay, InceptionV3 is still able to fit the random training set extremely well if not perfectly. Although not shown explicitly, on CIFAR10, both Inception and MLPs still fit perfectly the random training set with weight decay turned on. However, AlexNet with weight decay turned on fails to converge on random labels. To investigate the role of regularization in deep learning, we explicitly compare behavior of deep nets learning with and without regularizers.

Instead of doing a full survey of all kinds of regularization techniques introduced for deep learning, we simply take several commonly used network architectures and compare the behavior when turning off the equipped regularizers.

The following regularizers are covered:

- **Data augmentation:** augment the training set via domain-specific transformations. For image data, commonly used transformations include random cropping, random perturbation of brightness, saturation, hue, and contrast.
- **Weight decay:** equivalent to a ℓ_2 regularizer on the weights; also equivalent to a hard constrain of the weights to an Euclidean ball, with the radius decided by the amount of weight decay.
- **Dropout**³⁹: mask out each element of a layer output randomly with a given dropout probability. Only the Inception V3 for ImageNet uses dropout in our experiments.

Table 1 shows the results of Inception, Alexnet, and MLPs on CIFAR10, toggling the use of data augmentation and weight decay. Both regularization techniques help to improve the generalization performance, but even with all of the regularizers turned off, all of the models still generalize very well.

Table 2 in Appendix of Zhang et al.⁴⁴ shows a similar experiment on the ImageNet dataset. A 18% top-1 accuracy drop is observed when we turn off all the regularizers. Specifically, the top-1 accuracy without regularization is 59.80%, while random guessing only achieves 0.1% top-1 accuracy on ImageNet. More strikingly, with data augmentation on but other explicit regularizers off, Inception is able to achieve a top-1 accuracy of 72.95%. Indeed, it seems like the ability to augment the data using known symmetries is significantly more powerful than just tuning weight decay or preventing low training error.

Inception achieves 80.38% top-5 accuracy without regularization, while the reported number of the winner of ILSVRC 2012 achieved 83.6%. So while regularization is important, bigger gains can be achieved by simply changing the model architecture. It is difficult to say that the regularizers count as a fundamental phase change in the generalization capability of deep nets.

3.1. Implicit regularizations

Early stopping was shown to implicitly regularize on some convex learning problems.^{21, 43} In Table 2 in Appendix of Zhang et al.,⁴⁴ we show in parentheses the best test accuracy along the training process. It confirms that early stopping could *potentially*^a improve the generalization performance. Figure 2a shows the training and testing accuracy on ImageNet. The shaded area indicates the accumulative best test accuracy, as a reference of potential performance gain for early stopping. However, on the CIFAR10 dataset, we do not observe any potential benefit of early stopping.

Batch normalization is an operator that normalizes the layer responses within each mini-batch. It has been widely adopted in many modern neural network architectures such as Inception and Residual Networks. Although not explicitly designed for regularization, batch normalization is usually found to improve the generalization performance. The

^a We say “potentially” because to make this statement rigorous, we need to have another isolated test set and test the performance there when we choose early stopping point on the first test set (acting like a validation set).

Inception architecture uses a lot of batch normalization layers. To test the impact of batch normalization, we create a “Inception w/o BatchNorm” architecture that is exactly the same as Inception, except with all the batch normalization layers removed. Figure 2b compares the learning curves of the two variants of Inception on CIFAR10, with all the explicit regularizers turned off. The normalization operator helps stabilize the learning dynamics, but the impact on the generalization performance is only 3~4%. The exact accuracy is also listed in the section “Inception w/o BatchNorm” of Table 1.

In summary, our observations on both explicit and implicit regularizers are consistently suggesting that regularizers, when properly tuned, could help to improve the generalization performance. However, it is unlikely that the regularizers are the fundamental reason for generalization, as the networks continue to perform well after all the regularizers removed.

4. FINITE-SAMPLE EXPRESSIVITY

Much effort has gone into characterizing the expressivity of neural networks, for example, Cybenko¹², Mhaskar²⁵, Delalleau and Bengio¹³, Mhaskar and Poggio²⁴, Eldan and Shamir¹⁵, Telgarsky⁴⁰, Cohen and Shashua¹¹. Almost all of these results are at the “population level” showing what functions of the entire domain can and cannot be represented by certain classes of neural networks with certain number of parameters. For example, it is known that at the population level, depth k is generically more powerful than depth $k - 1$.

We argue that what is more relevant in practice is the expressive power of neural networks on a finite sample of size n . It is possible to transfer population level results to finite sample results using uniform convergence theorems. However, such uniform convergence bounds would require

the sample size to be polynomially large in the dimension of the input and exponential in the depth of the network, posing a clearly unrealistic requirement in practice.

We instead directly analyze the finite-sample expressivity of neural networks, noting that this dramatically simplifies the picture. Specifically, as soon as the number of parameters p of a networks is greater than n , even simple two-layer neural networks can represent any function of the input sample. We say that a neural network C can represent any function of a sample of size n in d dimensions if for every sample $S \subseteq \mathbb{R}^d$ with $|S| = n$ and every function $f: S \rightarrow \mathbb{R}$, there exists a setting of the weights of C such that $C(x) = f(x)$ for every $x \in S$.

THEOREM 1. *There exists a two-layer neural network with ReLU activations and $2n + d$ weights that can represent any function on a sample of size n in d dimensions.*

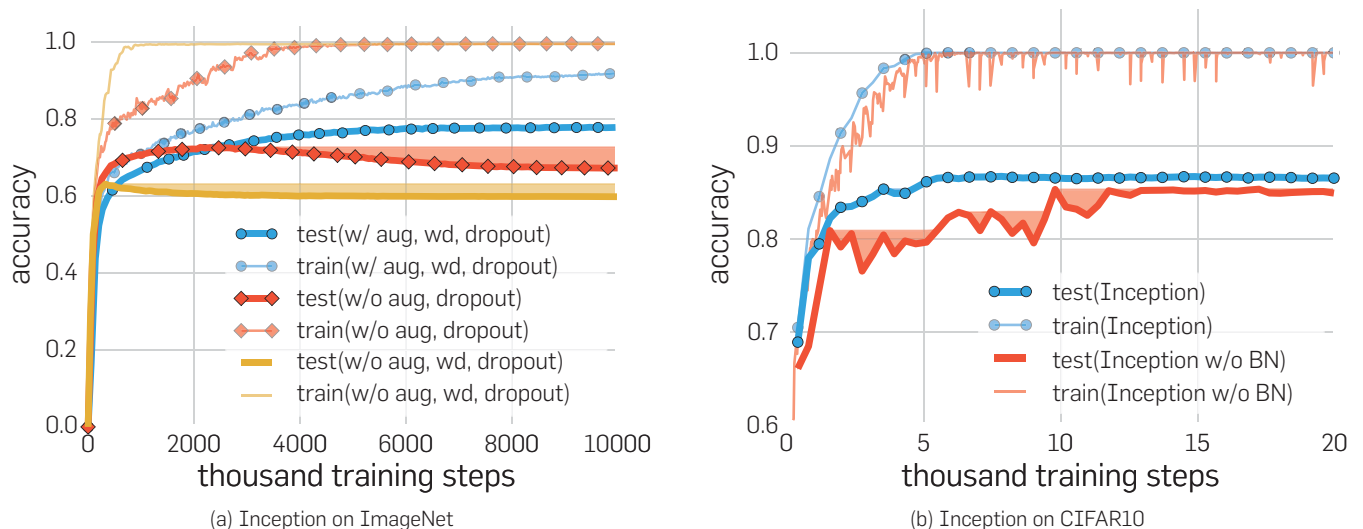
The proof is given in Appendix C of Zhang et al.,⁴⁴ where we also discuss how to achieve width $O(n/k)$ with depth k . We remark that it is a simple exercise to give bounds on the weights of the coefficient vectors in our construction. Lemma 1⁴⁴ gives a bound on the smallest eigenvalue of the matrix A . This can be used to give reasonable bounds on the weight of the solution w .

5. IMPLICIT REGULARIZATION: AN APPEAL TO LINEAR MODELS

Although deep neural nets remain mysterious for many reasons, we note in this section that it is not necessarily easy to understand the source of generalization for linear models either. Indeed, it is useful to appeal to the simple case of linear models to see if there are parallel insights that can help us better understand neural networks.

Suppose we collect n distinct data points $\{(x_i, y_i)\}$ where x_i ,

Figure 2. Effects of implicit regularizers on generalization performance. aug is data augmentation; wd is weight decay; BN is batch normalization. The shaded areas are the cumulative best test accuracy, as an indicator of potential performance gain of early stopping. (a) Early stopping could potentially improve generalization when other regularizers are absent. (b) Early stopping is not necessarily helpful on CIFAR10, but batch normalization stabilizes the training process and improves the generalization.



is d -dimensional feature vectors and y_i is labels. Letting loss denote a nonnegative loss function with $\text{loss}(y, y) = 0$, consider the *empirical risk minimization* (ERM) problem

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \text{loss}(w^T x_i, y_i) \quad (2)$$

If $d \geq n$, then we can fit any labeling. But is it then possible to generalize with such a rich model class and no explicit regularization?

Let X denote the $n \times d$ data matrix whose i -th row is x_i^T . If X has rank n , then the system of equations $Xw = y$ has an infinite number of solutions regardless of the right-hand side. We can find a global minimum in the ERM problem (2) by simply solving this linear system.

But do all global minima generalize equally well? Is there a way to determine when one global minimum will generalize whereas another will not? One popular way to understand quality of minima is the curvature of the loss function at the solution. But in the linear case, the curvature of all optimal solutions is the same.⁹ To see this, note that in the case when y_i is a scalar,

$$\nabla^2 \frac{1}{n} \sum_{i=1}^n \text{loss}(w^T x_i, y_i) = \frac{1}{n} X^T \text{diag}(\beta) X,$$

where $\beta_i = \frac{\partial^2 \text{loss}(z, y_i)}{\partial z^2} \Big|_{z=y_i}$. A similar formula can be found when y is vector valued. In particular, the Hessian is not a function of the choice of w . Moreover, the Hessian is degenerate at all global optimal solutions.

If curvature does not distinguish global minima, what does? A promising direction is to consider the workhorse algorithm, stochastic gradient descent (SGD), and inspect which solution SGD converges to. Since the SGD update takes the form $w_{t+1} = w_t - \eta_t e_t x_i$ where η_t is the step size and e_t is the prediction error loss. If $w_0 = 0$, we must have that the solution has the form $w = \sum_{i=1}^n \alpha_i x_i$ for some coefficients α . Hence, if we run SGD we have that $w = X^T \alpha$ lies in the span of the data points. If we also perfectly interpolate the labels, we have $Xw = y$. Enforcing both of these identities, this reduces to the single equation

$$XX^T \alpha = y \quad (3)$$

which has a *unique solution*. Note that this equation only depends on the dot-products between the data points x_i . We have thus derived the “kernel trick”³⁴—albeit in a round-about fashion.

We can therefore perfectly fit any set of labels by forming the Gram matrix (aka the *kernel matrix*) on the data $K = XX^T$ and solving the linear system $K\alpha = y$ for α . This is an $n \times n$ linear system that can be solved on standard workstations whenever n is less than a hundred thousand, as is the case for small benchmarks like CIFAR10 and MNIST.

Quite surprisingly, fitting the training labels exactly yields excellent performance for convex models. On MNIST with no preprocessing, we are able to achieve a test error of 1.2% by simply solving $K\alpha = y$ with a Gaussian kernel on the pixel representation. Note that this is not exactly simple as the kernel matrix requires 30 GB to store in memory. Nonetheless, this system can be solved in under 3 minutes

on a commodity workstation with 24 cores and 256 GB of RAM with a conventional LAPACK call. By first applying a Gabor wavelet transform to the data and then solving (3), the error on MNIST drops to 0.6%. Surprisingly, adding regularization does not improve either model’s performance!

Similar results follow for CIFAR10. Simply applying a Gaussian kernel on pixels and using no regularization achieves 46% test error. By preprocessing with a random convolutional neural net with 32,000 random filters, this test error drops to 17% error^b. Adding ℓ_2 regularization further reduces this number to 15% error. Note that this is without any data augmentation.

Note that this kernel solution has an appealing interpretation in terms of implicit regularization. Simple algebra reveals that it is equivalent to the *minimum ℓ_2 -norm* solution of $Xw = y$. That is, out of all models that exactly fit the data, SGD will often converge to the solution with minimum norm. It is very easy to construct solutions of $Xw = y$ that do not generalize: for example, one could fit a Gaussian kernel to data and place the centers at random points. Another simple example would be to force the data to fit random labels on the test data. In both cases, the norm of the solution is significantly larger than the minimum norm solution.

Unfortunately, this notion of minimum norm is not predictive of generalization performance. For example, returning to the MNIST example, the ℓ_2 -norm of the minimum norm solution with no preprocessing is approximately 220. With wavelet preprocessing, the norm jumps to 390. Yet the test error drops by a factor of 2. So while this minimum-norm intuition may provide some guidance to new algorithm design, it is only a very small piece of the generalization story.

6. CONCLUSION

In this work, we presented a simple experimental framework for interrogating purported measures of generalization. The experiments we conducted emphasize that the *effective capacity* of several successful neural network architectures is large enough to shatter the training data. Consequently, these models are in principle rich enough to memorize the training data. This situation poses a conceptual challenge to statistical learning theory as traditional measures of model complexity struggle to explain the generalization ability of large artificial neural networks. An important insight resulting from our experiments is that optimization continues to be empirically easy even if the resulting model does not generalize. What drives generalization therefore cannot be identical to what makes optimization of deep neural networks easy in practice, another important—yet, as we show, distinct—question.

The situation we find ourselves in bears semblance to where machine learning was in the 1960s. One of the first striking successes of machine learning dates back to Rosenblatt’s 1958 discovery of the Perceptron algorithm. In modern language, the Perceptron learns a linear function from labeled examples. Cycling through the data one

^b This conv-net is the Coates and Ng¹⁰ net, but with the filters selected at random instead of with k -means.

example at a time, whenever the Perceptron encounters an example where the sign of the linear function disagrees with the binary label, it nudges the coefficients of the linear function either toward or away from the example. Analysis from the 1960s provided generalization results for the Perceptron assuming that there was some solution out there that properly labeled all data we might ever see. An instance of the popular stochastic gradient method, the Perceptron, remains strikingly similar to modern machine learning practice. Indeed, the results on linear models in Section 5 are effectively a generalization of the 60-year-old results on the Perceptron.

The primary difference between now and then is one of scale and complexity. In place of a simple linear function, we find intricate models that stack several nonlinear transformations, so-called layers, on top of each other. Each layer has its own set of trainable parameters. Such concatenation adds complexity: we no longer get the beautiful convergence and generalization theorems of the Perceptron. The classic Perceptron theory explained why overparameterized *linear* models might generalize in some special cases, but these results do not provide an explanation of the power of nonlinear models.

6.1. A partial survey of recent progress

The original version of this paper⁴⁴ motivated a tremendous amount of new work on generalization that we cannot fully survey here. However, we will attempt to summarize some general trends.

Regarding our observation that conventional generalization bounds based on uniform convergence or uniform stability are inadequate for overparameterized deep neural networks, extensive efforts were made toward tighter generalization bounds (e.g., Kawaguchi et al.,¹⁹ Bartlett et al.,⁵ Neyshabur et al.,²⁸ Golowich et al.,¹⁷ Liang et al.²⁰). In the *PAC-Bayes* setting, where the learning algorithm is allowed to output a distribution over parameters, new generalization bounds were also derived.^{14, 29, 2, 46}

Aligned with our observation that overparameterized deep networks generalize even without any explicit regularization, and our analysis of implicit regularization in linear models, there is renewed interest in seeking to explain generalization in deep learning by characterizing the implicit regularization induced by the learning algorithms.^{37, 38, 35, 1}

In-depth analysis on memorization of overparameterized models also extends our intuition on overfitting from the traditional U-shaped risk curve to the “double descent” *risk curve*. Specifically, in the overparameterized regime where the model capacity greatly exceeds the training set size, fitting all the training examples (i.e., *interpolating* the training set), including noisy ones, is not necessarily at odds with generalization.^{23, 7, 6, 16}

Despite significant progress on theoretical understanding of deep learning in the past few years, a full mathematical characterization of the whole story remains challenging. Since the original version of this paper,⁴⁴ much more work starts approaching the question of understanding deep learning using *empirical studies*, by designing systematic and principled experiments (e.g., Arpit et al.,³ Zhao et al.,⁴⁵ Morcos et al.,²⁶ Recht et al.,³³ Toneva et al.⁴¹). The randomization test proposed in this paper serves as the backbone in the experimental

design in many of those studies. Dedicated workshops on phenomena in deep learning are being organized in all major machine learning conferences nowadays. Even some theory conferences start to consider pure empirical studies that reveal “interesting and not well understood behavior”^c in the call-for-papers. Thus, we are excited to see what happens in the next four years as well as excited to have highlighted some of the development over the past four years since we wrote the original manuscript. **□**

References

- Arora, S., Cohen, N., Hu, W., Luo, Y. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, eds. Vol. 32. Curran Associates, Inc., 2019, 7411–7422.
- Arora, S., Ge, R., Neyshabur, B., Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*. J. Dy and A. Krause, eds. 2018, 254–263.
- Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., Lacoste-Julien, S. A closer look at memorization in deep networks. In *International Conference on Machine Learning*. D. Precup and Y.W. Teh, eds. 2017, 233–242.
- Bartlett, P.L. The sample complexity of pattern classification with neural networks—The size of the weights is more important than the size of the network. *IEEE Trans. Inform. Theory*, 44 (1998), 525–536.
- Bartlett, P.L., Foster, D.J., Telgarsky, M.J. Spectrally-normalized margin bounds for neural networks. *Adv. Neural Inform. Process. Syst.* 2017, 6240–6249.
- Belkin, M., Hsu, D., Ma, S., Mandal, S. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proc. Natl. Acad. Sci.* 32, 116 (2019), 15849–15854.
- Belkin, M., Hsu, D., Mitra, P. Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate. *Adv. Neural Inform. Process. Syst.*, 2018, 2300–2311.
- Bousquet, O., Elisseeff, A. Stability and generalization. *J. Mach. Learn. Res.* 2 (2002), 499–526.
- Choromanska, A., Henaff, M., Mathieu, M., Arous, G.B., LeCun, Y. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*. G. Lebanon and S.V.N. Vishwanathan, eds. 2015, 192–204.
- Coates, A., Ng, A.Y. Learning feature representations with *k*-means. In *Neural Networks: Tricks of the Trade, Reloaded*. Springer, 2012.
- Cohen, N., Shashua, A. Convolutional rectifier networks as generalized tensor decompositions. In *International Conference on Machine Learning*. M.F. Balcan and K.Q. Weinberger, eds. 2016, 955–963.
- Cybenko, G. Approximation by superposition of sigmoidal functions. *Math. Control Signal. Syst.* 4, 2 (1989), 303–314.
- Delalleau, O., Bengio, Y. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems 24*. J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger, eds. Curran Associates, Inc., 2011, 666–674.
- Dziugaite, G.K., Roy, D.M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*. 2017.
- Eldan, R., Shamir, O. The power of depth for feedforward neural networks. In *Conference on Learning Theory*. V. Feldman, A. Rakhlin, and O. Shamir, eds. 2016, 907–940.
- Feldman, V. Does learning require memorization? a short tale about a long tail. *arXiv preprint arXiv:1906.05271* (2019).
- Golowich, N., Rakhlin, A., Shamir, O. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*. P.R. Sébastien Bubeck, V. Perchet, eds. 2018, 297–299.
- Hardt, M., Recht, B., Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*. M.F. Balcan and K.Q. Weinberger, eds. 2016, 1225–1234.
- Kawaguchi, K., Kaelbling, L.P., Bengio, Y. Generalization in deep learning. *CoRR*, arXiv:1710.05468 (2017).
- Liang, T., Poggio, T., Rakhlin, A., Stokes, J. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*. K. Chaudhuri and M. Sugiyama, eds. arXiv:1711.01530 (2017), 888–896.
- Lin, J., Camoriano, R., Rosasco, L. Generalization properties and implicit regularization for multiple passes SGM. In *International Conference on Machine Learning*. M.F. Balcan and K.Q. Weinberger, eds. 2016, 2340–2348.
- Livni, R., Shalev-Shwartz, S., Shamir, O. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems 27*. Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds. Curran Associates, Inc., 2014, 855–863.
- Ma, S., Bassily, R., Belkin, M. The power of interpolation: Understanding the effectiveness of sgd in modern

^c Quoted from the call-for-papers of Algorithmic Learning Theory (ALT) 2020.

- over-parametrized learning. In *International Conference on Machine Learning*. J. Dy and A. Krause, eds. 2018, 3325–3334.
24. Mhaskar, H., Poggio, T.A. Deep vs. shallow networks: An approximation theory perspective. *Anal. Appl.* 6, 14 (2016).
 25. Mhaskar, H.N. Approximation properties of a multilayered feedforward artificial neural network. *Adv. Comput. Math.* 1, 1 (1993), 61–80.
 26. Morcos, A., Raghu, M., Bengio, S. Insights on representational similarity in neural networks with canonical correlation. *Adv. Neural Inform. Process. Syst.* 2018, 5727–5736.
 27. Mukherjee, S., Niyogi, P., Poggio, T., Rifkin R. Statistical learning: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Technical Report AI Memo 2002-024*. Massachusetts Institute of Technology, 2002.
 28. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N. Exploring generalization in deep learning. *Adv. Neural Inform. Process. Syst.*, 2017, 5947–5956.
 29. Neyshabur, B., Bhojanapalli, S., Srebro, N. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.
 30. Neyshabur, B., Tomioka, R., Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *CoRR*, abs/1412.6614, 2014.
 31. Neyshabur, B., Tomioka, R., Srebro, N. Norm-based capacity control in neural networks. In *Conference on Learning Theory*. S.K. Peter Grünwald and E. Hazan, eds. 2015, 1376–1401.
 32. Poggio, T., Rifkin, R., Mukherjee, S., Niyogi, P. General conditions for predictivity in learning theory. *Nature* 6981, 428 (2004), 419–422.
 33. Recht, B., Roelofs, R., Schmidt, L., Shankar, V. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811* (2019).
 34. Schölkopf, B., Herbrich, R., Smola, A.J. A generalized representer theorem. In *Conference on Learning Theory*. 2001, 416–426.
 35. Shah, V., Kyrillidis, A., Sanghavi, S. Minimum norm solutions do not always generalize well for over-parameterized problems. *CoRR*. arXiv:1811.07055 (2018).
 36. Shalev-Shwartz, S., Shamir, O., Srebro, N., Sridharan, K. Learnability, stability and uniform convergence. *J. Mach. Learn. Res.*, 11 (2010), 2635–2670.
 37. Smith, S.L., Le, Q.V. A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*, 2018.
 38. Soudry, D., Hoffer, E., Nacson, M.S., Gunasekar, S., Srebro, N. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.* 70, 19 (2018), 1–57.
 39. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 1, 15 (2014), 1929–1958.
 40. Telgarsky, M. Benefits of depth in neural networks. In *Conference on Learning Theory*. V. Feldman, A. Rakhlin, and O. Shamir, eds. 2016, 1517–1539.
 41. Toneva, M., Sordoni, A., des Combes, R.T., Trischler, A., Bengio, Y., Gordon, G.J. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.
 42. Vapnik, V.N. *Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control*. Wiley, 1998.
 43. Yao, Y., Rosasco, L., Caponnetto, A. On early stopping in gradient descent learning. *Const. Approx.* 2, 26 (2007), 289–315.
 44. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2019.
 45. Zhao, S., Ren, H., Yuan, A., Song, J., Goodman, N., Ermon, S. Bias and generalization in deep generative models: An empirical study. In *Advances in Neural Information Processing Systems 31*. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds. Curran Associates, Inc., 2018, 10792–10801.
 46. Zhou, W., Veitch, V., Austern, M., Adams, R.P., Orbanz, P. Non-vacuous generalization bounds at the ImageNet scale: A PAC-Bayesian compression approach. In *International Conference on Learning Representations*, 2019.

Chiyuan Zhang and Samy Bengio
 ([chiyuan, bengio]@google.com), Google Brain, Mountain View, CA, USA.

Oriol Vinyals (vinyals@google.com), DeepMind, London N1C 4AG, U.K.

Moritz Hardt and Benjamin Recht
 ([hardt, brecht]@berkeley.edu), University of California, Berkeley, CA, USA.
 Work performed at Google Brain.



Watch the authors discuss this work in the exclusive *Communications* video.
<https://cacm.acm.org/videos/understanding-deep-learning>

Data Cleaning

“Dirty data across businesses and governments costs trillions every year.”

Ihab F. Ilyas, Xu Chu

ISBN: 978-1-450371-53-7

DOI: 10.1145/3310205

<http://books.acm.org>

<http://store.morganclaypool.com/acm>



ACM BOOKS
 Collection II



Association for
Computing Machinery

2018 JOURNAL IMPACT
FACTOR: 6.131

ACM Computing Surveys (CSUR)

ACM Computing Surveys (CSUR) publishes comprehensive, readable tutorials and survey papers that give guided tours through the literature and explain topics to those who seek to learn the basics of areas outside their specialties. These carefully planned and presented introductions are also an excellent way for professionals to develop perspectives on, and identify trends in, complex technologies.



For further information
and to submit your
manuscript,
visit csur.acm.org

Technical Perspective Localizing Insects Outdoors

By Prabal Dutta


THE VISION OF tiny, flying robotic insects has captured the imagination of researchers in many disciplines. Some have sought to create entirely synthetic robotic creatures while others have chosen to weave and deeply enmesh biological sensing and actuation with electronic computation and controls. The following paper by Iyer et al. takes a different approach. It neatly separates the problems of locomotion, solved using existing biology, and the problems of sensing, localization, and communications, solved using commercial microelectronics and new algorithms, packaged into a tiny payload. These two halves—one biological and one electronic—are then simply glued together to realize a cyborg bumblebee with a mind of its own carrying sensors on our behalf.

Realizing this exciting vision requires solving myriad research and engineering problems, but perhaps none more daunting than how to localize a small and fast-moving bumblebee at a range of tens of meters. The problem is particularly challenging because of the severe and unforgiving size, weight, and power (SWaP) constraints on payload capacity. A key contribution of the accompanying work is a novel method for low-power, mid-range, outdoor localization that estimates the angle-of-departure of signals from several multi-antenna access points in clear line-of-sight settings, as might be typical on farms and fields.

This technical feat is accomplished by having a pair of antennas transmit nominally identical signals that combine at the receiver. The difference in the path length between the two antennas and the insect manifests as an amplitude and phase difference in their sum at the receiver, which is also attenuated by the path loss the signals both have in common for the line-of-sight path. If this (unknown) path loss is normalized by sweeping the phase

A key contribution of the following paper is a novel method for low-power, mid-range, outdoor localization.

of one signal, which allows the maximum strength of the received signal to be recovered, then the path loss can also be determined and factored out. Finally, the effect of multipath is reduced by using multiple access points and multiple antenna pairs at each access point.

Going far beyond just the theory, or even a benchtop proof-of-concept, the authors build a low SWaP circuit (just 39 mm², 102 mg, and 138 μA) that realizes this idea, augments it with sensors and backscatter communications using a custom-designed antenna, and uses it to demonstrate how a bumblebee-based sensor can sample a large field and download data upon return to the hive. The prototype integrates temperature, humidity, and light sensors with a microcontroller that offers 32 KB of memory for sensor and location data storage, along with a 1 mAh battery, that weighs 70 mg, to yield a system that can run for an impressive seven hours! Looking ahead, the authors report that RF or solar energy harvesting could enable indefinite lifetime, especially when combined with ultra-low power custom electronics. 

Prabal Dutta is an associate professor of electrical engineering and computer sciences at the University of California at Berkeley, CA, USA.

Copyright held by author.

3D Localization for Subcentimeter-Sized Devices

By Rajalakshmi Nandakumar, Vikram Iyer, and Shyamnath Gollakota

Abstract

The vision of tracking small IoT devices runs into the reality of localization technologies—today it is difficult to continuously track objects through walls in homes and warehouses on a coin cell battery. Although Wi-Fi and ultra-wideband radios can provide tracking through walls, they do not last more than a month on small coin and button cell batteries because they consume tens of milliwatts of power. We present the first localization system that consumes microwatts of power at a mobile device and can be localized across multiple rooms in settings such as homes and hospitals. To this end, we introduce a multiband backscatter prototype that operates across 900 MHz, 2.4 GHz, and 5 GHz and can extract the backscatter phase information from signals that are below the noise floor. We build subcentimeter-sized prototypes that consume 93 μW and could last five to ten years on button cell batteries. We achieved ranges of up to 60 m away from the AP and accuracies of 2, 12, 50, and 145 cm at 1, 5, 30, and 60 m, respectively. To demonstrate the potential of our design, we deploy it in two real-world scenarios: five homes in a metropolitan area and the surgery wing of a hospital in patient pre-op and post-op rooms as well as storage facilities.

1. INTRODUCTION

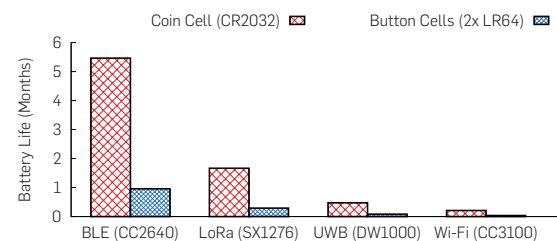
Recent years have seen significant advances in wireless localization.^{21, 19} However, existing solutions do not meet the requirements for size-constrained IoT applications. Figure 1 shows battery life of common radio technologies such as BLE, LoRa, ultra-wideband (UWB), and Wi-Fi, each running at a 1% duty cycle with small coin and button cell batteries for equal comparison. The shorter battery life limits the adoption of tracking solutions based on these radio technologies by making them inconvenient for consumer applications and infeasible for large-scale commercial deployments. Requiring large batteries on the other hand prevents scaling down the size of IoT devices. Although RFID tags are attractive from a power and size perspective, they have a limited range and do not work consistently through walls and other barriers. Consumers often deploy devices in rooms throughout homes, and similarly commercial deployments in settings such as hospitals require covering multiple patient rooms with a variety of obstructions and walls. Achieving localization in these scenarios would therefore require readers in every room, which significantly increases deployment cost.

This paper presents μLocate , the first wireless localization system that consumes microwatts of power at the mobile IoT devices and can be localized through walls in settings such as homes and hospitals. Our design can achieve

3D localization capabilities while supporting IoT devices that can be scaled to subcentimeter form factor. To achieve this, we design a backscatter-based solution that satisfies all of the above requirements. Specifically, we make the following hardware and systems contributions:

- We design and build a subcentimeter-sized IoT platform that supports low-power localization capabilities. Our platform integrates a low-power microcontroller and RF switch for backscatter rather than an active radio, as well as all required off-chip passive components and antennas. We custom fabricate flexible circuits using laser micromachining techniques and use an off-the-shelf microcontroller available in an ultraminiature $2\text{ mm} \times 1.5\text{ mm}$ package to achieve the small form factor.
- We achieve low-power long-range backscatter through walls by building on recent work on LoRa Backscatter¹⁷; however, this prior work requires implementing complex computation to perform chirp spread spectrum (CSS) coding on an FPGA platform, which consumes around 5–10 mW using off-the-shelf components. We present a novel backscatter architecture that enables CSS backscatter using off-the-shelf microcontrollers at significantly lower power. Specifically, because these microcontrollers lack the capability to easily implement the complex CSS coding, we instead delegate this coding to the access point, which transmits the CSS signal. By doing this, our low-power microcontroller simply needs to run an oscillator to frequency shift the CSS signal and encode data using ON-OFF keying in reflections.

Figure 1. Radio localization battery life. Battery life estimates for different technologies operating at 3 V from coin and button cell batteries running at 1% duty cycle.



The original version of this paper was published in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, ACM.

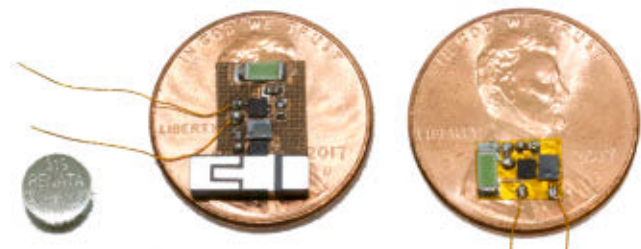
- Because the localization accuracy is directly proportional to the signal bandwidth, we design and build a *novel multiband backscatter hardware system that can concurrently operate across the ISM bands at 900 MHz, 2.4 GHz, and 5 GHz*. Specifically, the access point (AP) transmits signals across these frequencies, which are backscattered by our device. We combine the received signals across all of these frequencies to disambiguate between the multipath reflections and extract the direct line-of-sight path.

To summarize, our system works as follows: the AP, which is placed at a known location, transmits a 500 kHz chirp spread spectrum signal. The IoT device uses a low-power microcontroller to shift this signal by 1–2 MHz and backscatter it back to the AP. The AP then extracts the phase information from the weak backscattered signals that are below the noise floor. It repeats this process concurrently across the 900 MHz, 2.4 GHz, and 5 GHz bands and combines the phase to disambiguate the multipath in the environment.

Implementing this system introduces the following three algorithmic challenges: First, in contrast to direct radio signals that attenuate as d^2 , backscatter signals attenuate as d^4 . As a result, we need a way to extract the phase from backscattered signals, which are below the noise floor at long distances. Second, our IoT devices use small, low-power microcontrollers to shift the chirp spread spectrum signal from the AP. This introduces frequency and sampling offsets in the weak backscattered signals that have to be corrected to accurately estimate phase. Third, querying all the 500 kHz bands sequentially across all the ISM bands requires a total of 572 frequencies, which takes more than four seconds, introducing a significant delay overhead. Concurrently querying all these 572 frequencies requires the AP to proportionally reduce the power at each of the frequencies to be compliant with FCC regulations; this in turn would significantly reduce the range of our system.

In the rest of the paper, we address the above challenges and build multiple prototypes of our design as shown in Figure 2. We build our prototypes using commercial off-the-shelf components such as switches, microcontrollers, and

Figure 2. μ Locate prototypes. Our miniaturized prototypes require two button cell batteries (left), which are as small as 5.8 mm in diameter. Our multiband prototype based on the KL03 microcontroller is $11.8 \times 7.5 \times 2.1$ mm and includes chip antennas for 900 MHz, 2.4 GHz, and 5 GHz. Our 5-GHz prototype (right) measuring $7.2 \times 5.1 \times 0.5$ mm is designed to operate at shorter ranges and in an even smaller form factor. The prototypes are placed on a U.S. penny for scale.



chip antennas. Our first prototype uses the 2.0×1.6 mm Kinetis KL03 microcontroller with a 2.4 GHz and 900 MHz dual band chip antenna along with a 5 GHz chip antenna. We miniaturize our second prototype using a laser micromachining method that produces flexible circuits. We use the Kinetis microcontroller with only the 5 GHz chip antenna that limits the range but enables a further miniaturized device. We also present an ASIC design for our multiband backscatter approach to further reduce size and power.

Our evaluation shows our off-the-shelf hardware and ASIC consume $93 \mu\text{W}$ and $5 \mu\text{W}$, respectively. This translates to an expected operational lifetime of 5–10 years of duty cycled operation on small, 5.8 mm diameter button cell batteries for our off-the-shelf microcontroller hardware and ASIC prototypes. Further, we demonstrate 3D localization accuracy, which scales with the distance. Our system gives localization errors of 2, 12, 50, and 145 cm at 1, 5, 30, and 60 m, respectively, between the AP and our backscatter devices. Finally, across distances up to 60 m from an AP, our algorithm can compute the location values using between 9 and 28 frequencies, which translates to a latency of 25–70 ms.

In addition to the above benchmarks that characterize our system's performance, we also deploy the system in the following two real-world scenarios:

- Five homes in a metropolitan area including three single-story apartments and two multistory townhouses. We select a variety of locations and orientations across different rooms, behind closed doors, in closets, on shelves, and even hidden in couches to determine whether our system can localize objects across an entire home to enable item tracking.
- Surgery wing of a hospital including patient pre-op and post-op rooms as well as storage facilities. We run experiments in various locations for tracking mobile equipment such as on IV poles and vital signs monitors that travel with patients between different rooms.

2. SYSTEM DESIGN

Our design has three key components: (1) our low-power architecture that delegates the coding operation to the access point, allowing us to decode backscatter signals at large distances using subcentimeter-sized devices, (2) our phase-extraction algorithm that can extract the phase from signals below the noise floor, and (3) our online search algorithm that dynamically queries a different set of frequencies given the signal quality to reduce latency. In the rest of this section, we describe each of these components.

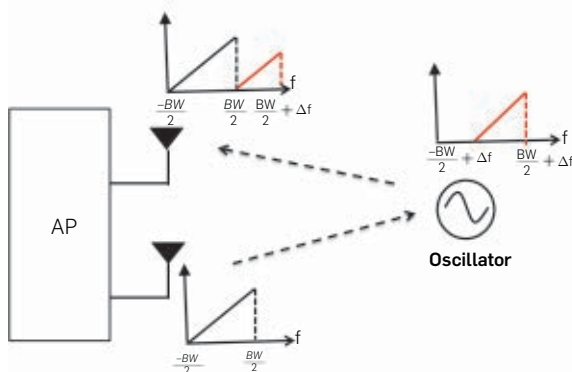
2.1. Low-power delegating architecture

The goal of our architecture is to enable localization at long ranges with very small, low-power backscattering IoT devices. To achieve this, we have to address two competing requirements: (1) because backscattered signals are orders of magnitude weaker than active radio transmissions, to achieve long range, we need to perform significant amounts of coding, and (2) in order to be compatible with off-the-shelf low-power microcontrollers, the IoT device design must be limited to simple operations.

To achieve this, we delegate the complex coding operations to the access point (AP). Our design works as shown in Figure 3. The μ Locate AP transmits a coded narrowband signal, whereas the μ Locate IoT device simply backscatters this signal transmitted by the AP with a frequency shift of 1–2 MHz. This shifting can be achieved using the built-in oscillators on commercially available microcontrollers (see Section 3). Our architecture therefore eliminates the need for an active radio on the μ Locate device. Shifting the signal on the IoT device has three key advantages: (1) it helps distinguish the backscattered signal from the direct signal transmitted by the AP in the frequency domain, allowing the receiver to easily decode it. More importantly, at the shifted frequencies, the receiver effectively receives a coded backscattered signal that it can use to extract the phase required for localization. Furthermore, we achieve this without requiring the IoT device to perform the complex coding operations itself, allowing for centimeter-scale low-power implementations without a custom ASIC. (2) By shifting the signal to different frequencies, multiple tags can coexist and be localized simultaneously using a single AP, and (3) by using ON and OFF keying at the backscatter device, one can also enable data communication in addition to localization.

What kind of coding do we use at the AP? The objective here is to pick the coding scheme that can be used to decode the phase of backscatter signals that are far below the noise floor. To this end, we use chirp spread spectrum as our coding mechanism. In chirp spread spectrum, we transmit a signal with a linearly changing frequency over bandwidth (BW) varying between $\frac{-BW}{2}$ and $\frac{+BW}{2}$. Chirp signals have the following advantages that make them the best fit for our application: (1) in comparison to phase, amplitude and discrete frequency-shift modulation, chirp spread spectrum (CSS) achieves an efficient trade-off between bandwidth and decoding capability, when the signal is drowned by noise.³ Further, it is resilient to both in-band and out-of-band interference,¹ and (2) unlike direct-sequence spread-spectrum that requires complex synchronization and has a long acquisition time when the signal is below the noise floor,^{14,5} CSS

Figure 3. Low-power delegating architecture. The access point (AP) transmits a chirp spread spectrum signal with a bandwidth BW to the IoT device with an oscillator and RF switch. The switch backscatters the coded signal back to the AP with a frequency shift of Δf .



receivers have comparatively lower complexity and significantly shorter acquisition times.¹

Specifically, we choose a narrow BW of 500 kHz where the chirp duration T is 7 ms, which we find balances accuracy and latency well. The receiver at the AP samples these signals at 1 MHz. In the next few sections, we first describe how to estimate the phase from CSS signals. We then show how to selectively query across the 900 MHz, 2.4 GHz, and 5 GHz ISM bands to disambiguate the multipath, estimate the range, and achieve 3D localization. Finally, we describe how we can achieve real-time tracking.

2.2. Below-noise backscatter phase

Assume that the AP, which is placed at a known location, is separated from the IoT device by a distance of d . When the AP transmits the chirp signal, it propagates a total distance of $2d$ including the time it takes for the backscattered signal from the IoT device to arrive back at the AP. The wireless channel of such a signal is $h = ae^{-2\pi f \frac{2d}{c}}$. Here, a is the attenuation, f is the frequency at which the signal is being transmitted, and c is the speed of RF signals in the medium. At a high level, if we can extract the phase of the backscattered signal at a specific frequency, we can estimate the range d .

Thus, if the AP transmits a tone at a single frequency f , in the absence of multipath, the phase of the backscatter signal can be used to estimate the range. However, such single-tone signals (e.g., RFID) have a limited range in the context of backscatter communication and hence cannot achieve the long ranges that are required for IoT localization. As described earlier, the AP instead transmits a linear frequency modulated chirp pulse, which allows our system to operate at longer ranges without further amplifying the signal. As shown in Figure 3, the tag then shifts this chirp signal by a frequency Δf , and the shifted signal is received back at the AP. Hence, the receiver receives a chirp signal whose frequency varies from $-BW/2 + \Delta f$ to $+BW/2 + \Delta f$.

We use correlation to decode this signal. Specifically, the receiver first correlates the received signal with a downchirp, a signal where the frequencies linearly decrease from $+BW/2 + \Delta f$ to $-BW/2 + \Delta f$. This downchirp is synthesized on the receiver. During the multiplication step of the correlation, the linear change in the frequency between the receiver upchirps and the synthesized downchirps cancels each other out. During the addition step of the correlation, we effectively sum the energy across all the chirp frequencies providing coding gain and allowing us to decode the backscatter signals below the noise floor.

Extracting the channel phase information from this signal requires us to address three challenges: (1) because the chirp signal is spread across frequencies, we do not get the phase at a single frequency but rather the chirp phase that is a combination of phases across all the frequencies in the chirp, (2) to decode and estimate the phase of the signal, we need to accurately estimate the beginning of the backscatter chirp, and (3) our small low-power microcontrollers that shift the incoming chirp by Δf introducing an unknown carrier frequency offset (CFO) between the AP and the IoT device that changes the phase of the received signal.

To address the above challenges, at a high level, we first describe how we jointly estimate the carrier frequency offset (CFO) and correct for the start of the backscatter chirp. We then show how to compute the channel phase information given the phase of the backscattered chirp. The details for this are described in our SenSys paper.¹²

2.3. Multipath disambiguation

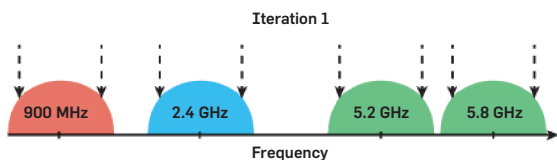
In practice, due to multipath, the obtained phase is actually the sum of phase of the direct line of sight signal and the phase of the different multipath reflections. Hence, to accurately localize the IoT device, we should disambiguate the various multipath reflections from the direct line of sight signal. To achieve this, we dynamically send chirps across the ISM bands in 900 MHz, 2.4 GHz, and 5 GHz and estimate the phase of each frequency using the above method. At a high level, by combining the phase information across all these frequencies, we can simulate an ultra-wide band transmission that can be used to disambiguate the multipath in the environment. Because our chirps are 500 kHz wide, we can transmit multiple chirps in adjacent bands across the three ISM bands. This however would significantly increase the latency of localization. Specifically, there are bandwidths of 26 MHz, 80 MHz, and 180 MHz in the 900 MHz, 2.4 GHz, and 5 GHz ISM bands. Dividing them into 500 kHz intervals results in 572 chirps across all these bands, which translates to 4 s using a 7-ms chirp. We instead design a dynamic frequency selection algorithm that significantly reduces the number of queried frequencies by 10–20×.

Dynamic frequency selection. Our algorithm is based on three key principles.

We determine the ISM bands that can be used depending on the distance of the IoT device. Specifically, signals at 5 GHz have very high attenuation and can be decoded using the above method only if the IoT device is in the same room as the AP. Similarly, signals in the ISM band of 2.4 GHz experience a lower attenuation compared to 5 GHz but have a higher attenuation than 900 MHz. Thus, we can prune a number of frequencies depending on the attenuation observed from the IoT device. Specifically, we first send a chirp in the 2.4 GHz band and determine the SNR of the chirp. If the SNR is very high, the device is at a short distance from the AP and hence all three ISM bands can be used. Otherwise, only the 900 MHz band and 2.4 GHz band can be used to estimate the distance of the IoT device. When the receiver cannot decode the initial chirp, we use only the 900 MHz band.

Each new frequency provides a new linear equation for the multipath combination at that frequency. However, picking

Figure 4. Dynamic frequency selection. In the first iteration, we start with the frequencies that are most separated. This translates to the frequencies at the edges of the three ISM bands.



two frequencies that are next to each other does not provide independent equations because the amplitude and phases of different multipaths are similar for adjacent frequencies. Thus, querying frequencies that are separated by the highest bandwidth provides more useful information than querying adjacent frequencies. Thus, we can reduce the number of frequencies that are queried by picking them such that the gap between them is maximized.

The backscatter device can reflect signals simultaneously across multiple frequencies. Thus, the AP can concurrently transmit four coded signals centered at frequencies $f_1, f_2, f_3,$ and f_4 , and the backscattered phase can be decoded at all these frequencies. This is used to parallelize the number of queries. We note however that requiring the AP to transmit multiple concurrent frequencies in the ISM band requires us to reduce the power on each of these frequencies proportionally to comply with FCC regulations. This would reduce the range of our system. We instead use the following rule to opportunistically parallelize our transmissions if the signal can be decoded at 2.4 GHz and then the 900 MHz is much stronger and hence we can query four frequencies concurrently at 900 MHz. Similarly, if the signal can be decoded at 5 GHz, we can query four frequencies concurrently at 2.4 GHz. Finally, if the signal strength is strong at any of these ISM bands, we increase the number of concurrent frequencies in that ISM band in the next round.

Algorithm 1 Dynamic Frequency Selection

```

1:  $min\_bands, max\_bands \triangleright$  Min and max of available bands
2:  $range = 0$ 
3: function QUERY( $min\_bands, max\_bands, range$ )
4:    $newrange = Range\_estimate(min\_bands \cup max\_bands)$ 
5:   if  $newrange - range < threshold$  then
6:     return  $newrange$ 
7:   if  $newrange - range > threshold$  then
8:     for  $i$  in  $1..length(min\_bands)$  do
9:       if  $max\_bands_i < min\_bands_i$  then
10:         $mid_i = \emptyset$ 
11:       else
12:         $mid_i = \frac{min\_bands_i + max\_bands_i}{2}$ 
13:       if  $mid_i = \emptyset$  then  $\triangleright$  No more frequencies available
14:         return  $newrange$ 
15:        $min\_bands = mid\_bands \cup mid$ 
16:        $max\_bands = mid \cup max\_bands$ 
17:       return QUERY( $min\_bands, max\_bands, newrange$ )
18: function Range_estimate( $frequency_{1..n}$ )
19:    $phase_{1..n} = ESTIMATEPHASE(frequency_{1..n}) \triangleright$  §2.2
20:    $Channel = DFT(frequency_{1..n}, phase_{1..n})$ 
21:    $peaks = FINDPEAKS(Channel, prominencethreshold)$ 
return  $peaks_1$ 

```

Using the above principles, we can design a binary search algorithm as shown in Algorithm 1. Specifically, once we identify the ISM bands that can be used, the AP first sends a chirp at the minimum and the maximum frequencies of the chosen bands as shown in Figure 4. To improve the distance resolution, the next frequency to query is picked using a recursive binary search function that chooses frequencies at

the extremes of the spectrum. After each query, the receiver computes the new distance estimate by using an inverse FFT on the phases at all the queried frequencies to get the time-domain multipath profile. By using a fixed energy threshold over this profile, we identify the closest (and therefore most direct) path from the device. Further implementation details are described in Nandakumar et al.¹²

3. PROTOTYPING DEVICES

Off-the-shelf prototypes. We build three different prototypes. The first uses the DE0-Nano FPGA development board to control an RF switch. We use an HMC190BMS8 RF switch for 900 MHz and 2.4 GHz, and a UPG2163T5N switch for 5 GHz. Both switches are mounted on a 2-layer Rogers 4350 substrate and toggle between open and short impedance states. The switches are connected to the same multiband antenna used at the AP. By using the onboard 50 MHz oscillator and PLL, we use this setup to experiment with different offsets prior to settling on 2 MHz.

The second prototype focuses on achieving our desired centimeter scale form factor and low-power consumption. Specifically, all our low-power IoT device needs is an oscillator and RF switch, as the coding is offloaded to the transmitter. In order to optimize for both size and form factor without custom silicon, we leverage low-power microcontrollers designed for IoT applications. A microcontroller such as the Kinetis KL03 requires roughly 30 μA to run its onboard oscillator at 8 MHz, and only 77 nA in its lowest power sleep mode.¹⁵ Because the platform is programmable, we can adjust the duty cycle to achieve significantly longer lifetimes on platforms with tiny batteries.

We fabricate our off-the-shelf prototype on a standard one-sided FR4 flex PCB material using the Kinetis KL03 microcontroller, which is available in a 2.0×1.6 mm WLCSP package, two UPG2163T 5N RF switches, and a 900-MHz and 2.4-GHz dual band chip antenna, as well as a 5-GHz chip antenna. We select these ceramic chip antennas that are specifically designed for small form factor applications and specify antenna gains of up to 3 dBi with efficiencies of 60–70% at 900 MHz and 2.4 GHz² and up to 79% at 5 GHz.¹⁶ The final assembly is as shown in Figure 2, which consumes an average of 93 μW .

The final prototype further miniaturizes the device by focusing on just 5 GHz as shown in Figure 2. We use a fast-turnaround laser micromachining method to produce

flexible circuits. We begin by placing a sheet of copper foil on a low-tack adhesive and cut the outline of the desired copper traces using a UV DPSS laser micromachining system. Next, we peel the excess copper off of the adhesive leaving only the desired pattern. We then place a piece of 25- μm -thick Kapton tape, which can withstand high temperatures required for soldering, onto the copper and lift the traces off of the adhesive. This method could be repeated and stacked to produce a multilayer design connected through vias as with a normal PCB. We use only the 5-GHz antenna in this prototype and hence are limited to a smaller range.

IC design. Further miniaturization and power optimization can be achieved by implementing a custom IC, which allows for combining the RF switch and impedances into a single chip. Further, this significantly decreases the required area to only a few mm^2 . The full IC design consists of a frequency synthesizer, RF switch, and at least two impedances states. We design and simulate a complete solution in a TSMC 65 nm LP process as described in Nandakumar et al.¹²

Figure 5 shows the lifetime of both our off-the-shelf and IC designs with different battery sizes. We limit the maximum of the plot to 10 years as this is the typical maximum shelf life of button cell batteries. These battery life values demonstrate that our design is so low power that the system performance is no longer limited by the electronics but rather the battery technology.

4. EVALUATION

We evaluate our system in line-of-sight (LOS) and through-wall settings. We then deploy μLocate in five different homes and a hospital to measure real-world performance.

4.1. Benchmarking accuracy

LOS scenario. We conduct experiments on a 100×100 m open field. We place the AP at one end of the field and move our FPGA IoT prototype away from the AP in increments of 10 m along different angles. Figure 6a plots the 3D localization error and shows the following:

- We have a 60-m range in LOS scenarios at which the worst-case 3D accuracy is 1.5 m. Beyond that distance, the received power of the backscattered signal was too low to decode even with the chirp spread spectrum coding.
- The accuracy scales with the distance from the IoT prototype. Specifically, we can achieve a localization error

Figure 5. Prototype battery life. Battery life estimates for duty cycled operation of our prototypes operating at 3 V from a coin or two button cell batteries. The plot is limited to 10 years, which is the shelf life of a button cell.

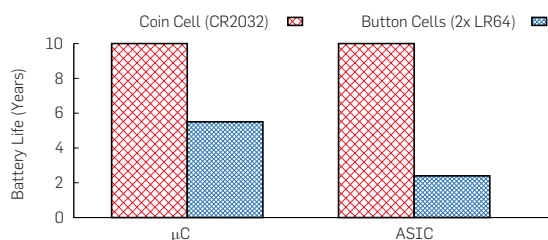
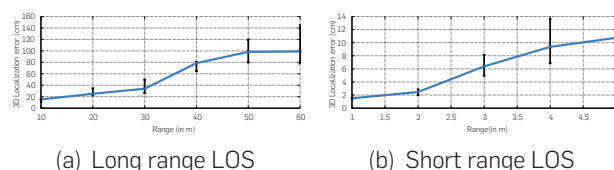


Figure 6. The plots show the 3D localization error for our line-of-sight benchmark. The figure (a) shows that our operational range is around 60 m. The figure (b) shows the 3D localization accuracy for distances below 5 m. At these distances, our system ends up using all frequencies across 900 MHz, 2.4 GHz, and 5 GHz.



of 15 cm at a distance of 10m, which increases to around 25 cm at a distance of 20m. This further increases to 78 cm at a distance of 40 m. This change is due to the fact that above 30m, the received power at 2.4 GHz is noisy due to attenuation in comparison to 900 MHz. This introduces error into the phase measurements reducing the accuracy.

We note that at the above-measured distances our algorithm did not pick any 5-GHz frequencies because the corresponding backscattered signal was very weak at these distances. So we rerun the experiments in a 5-m room with finer increments of 1 m. Figure 6b shows the 3D localization accuracy at these distances. The figure shows that when the IoT prototype is within a meter from the AP, the worst-case localization error was less than two cm. At a distance of 2 m, we could still achieve a 3-cm worst-case accuracy. However, the worst-case error was less than 14 cm up to distances of 5 m. The reason for these low errors was that the algorithm was able to use frequencies in the 5-GHz range, which significantly improve the location accuracies. In particular, 5 GHz helps with the accuracies for two main reasons: (1) higher frequencies translate to smaller wavelengths, which allows for better resolution, and (2) unlike 900 MHz and 2.4 GHz, each of which has a limited amount of bandwidth, our algorithm could query frequencies across a 180-MHz bandwidth in the 5-GHz range. These results demonstrate that for close-range room-scale applications we can leverage extra information from 5 GHz, whereas applications that require longer ranges cannot leverage these signals.

Finally, Figure 7 shows the number of frequencies that were queried by the AP before it converged to the location values for all the distances in the above two experiments. The plot shows the following:

- At distances less than 5 m, the AP had to query less than 20 frequencies. In fact, when the IoT devices were 1–2m away, the number of queries was even less at 15. This is because at short distances, the direct path is stronger than the non-line-of-sight paths for all the three frequency bands and hence the AP converges on the locations quickly. The latency for these locations is less than 35 ms as multiple frequencies can be queried in parallel.
- Between 10 and 20 m, the AP queries both 2.4 GHz and 900MHz to disambiguate the direct path and this

increases the number of frequencies to 25. However, the 900-MHz band frequencies can still be queried simultaneously, leading to a latency of 65 ms.

- An interesting trend happens at longer distances. Here, only 900-MHz frequencies are queried. Further, because the accuracies at these distances are much lower, the threshold values are also lower. As a result, the number of iterations is reduced to 8 for 60 m. However, because the SNR is weak, we have to query these frequencies sequentially. These two factors counteract each other and hence the latency stays between 55 and 70 ms.

Through-walls scenario. Next, we conducted experiments in an office building across multiple office rooms. The offices were separated with dry wall, metal studs, and wooden doors and had typical office furniture such as tables, chairs, and leather couches. Additionally, the tested locations had multiple Wi-Fi access points and 915-MHz RFID readers representing significant interference from other devices. Note that our chirp coding is resilient to both in-band and out-of-band interference.⁵ We place the AP in the first office room as shown in the layout in Figure 8. We then move our IoT prototype to different rooms with their doors closed shown as different points in the layout. For each location, we repeat the localization experiment multiple times and then compute the 3D localization errors.

Figure 9 plots the 3D localization error as a function of different positions as shown in Figure 8. The figure shows that for the most part the localization accuracy decreases as

Figure 7. Number of queried frequencies and its corresponding latency. The plot shows the total number of frequencies and the latency required for 3D localization across all locations in Figure 6b and a.

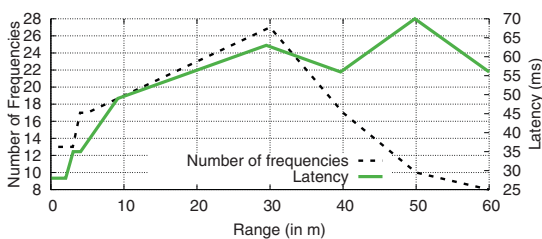


Figure 8. Through walls setup. Floor plan showing the AP and IoT device in an office environment spanning five rooms. All the doors were closed during the experiments.

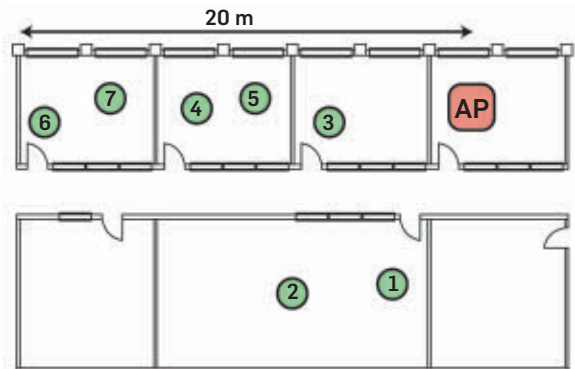
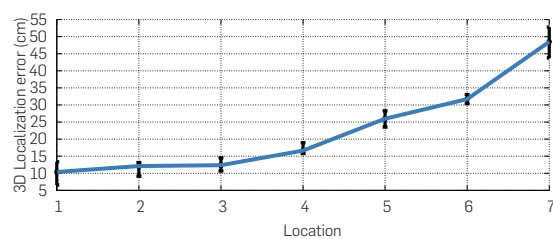


Figure 9. Through walls accuracy. The plot shows the 3D localization accuracy across the points indicated in Figure 8 rather than range due to their arbitrary placement.



the distance and the number of walls increase. It is however not always the case due to multipath and fading effects. We note however that the worst-case accuracies at location 6, which has 3 walls separating the AP, and the IoT prototype was still around 33 cm. This is expected because the 900-MHz and 2.4-GHz backscatter signals were strong enough to be able to reliably decode the phase information, which allows us to disambiguate the multipath.

4.2. Real-world deployments

Home deployments. We deploy our system in five homes in a major metropolitan area to understand its performance in realistic use cases. We select a variety of homes including three single-story apartments and two multistory townhouses. The single-story apartments had two to three rooms each, whereas the multistory townhouses had two or more floors with two and three rooms, respectively. In the apartment deployments, we select a central location for the AP to maximize coverage. For the townhouse deployments, we place the AP on the bottom floor for convenience. In each of the homes, we select a variety of locations including different rooms, behind closed doors, in closets, on shelves, and even hidden in couches to understand if our system can actually localize objects across a whole home and enable item tracking applications.

Figure 10 shows CDFs for 3D localization accuracy across each of the five homes. The figure shows that for the first three homes the worst-case localization accuracy was less than 30 cm. These three homes correspond to single-story apartments where all the devices are on the same floor. The worst-case accuracy was around 60 cm and 1.2 m for homes 4 and 5, respectively. These two homes were multistoried townhouses where the devices were on different floors. The higher error is due to two main factors. First, it was difficult to get the baseline distance measurements across floors. This contributed to errors in estimating the actual location of the IoT device. Second, in home 5, different floors were connected through a narrow staircase, whereas the direct path was through thick ceilings that significantly attenuated the signal. This highlights a basic challenge with localization techniques that require some direct path to appear at the receiver. Figure 11 depicts the above results by classifying them into categories across all the homes. We categorize the locations as LOS, NLOS on the same floor, hidden within a couch, in a closed closet, and finally on a different floor. For the reasons described above, the accuracy was lower when the device

was on a different floor. When the IoT device was hidden in a couch on the same floor, the error was less than 30 cm.

Hospital deployment. In order to evaluate realistic use cases in healthcare scenarios, we deploy our system in a local hospital. Specifically, we perform experiments in the surgery wing of the hospital and perform localization in patient pre-op and post-op rooms as well as storage facilities. Figure 12 shows the floor plan of the approximately 5000-ft² surgery wing. The area includes a waiting room and check-in desk, followed by a hallway with a row of patient rooms for pre-op and post-op care as well as a storage room. We perform measurements at locations that represent realistic use cases as indicated in Figure 12. We select a location for the AP in the side hallway in order to minimize disruption to hospital staff. We select locations in patient rooms and a storage closet as these are typical scenarios where hospital staff maintain a standard inventory of items. Additionally, we select other arbitrary locations in the hallways for tracking mobile equipment such as IV poles and vital signs monitors that travel with patients to different rooms. The majority of

Figure 10. Accuracy per home. The plot shows a CDF of localization error for all points measured in a home.

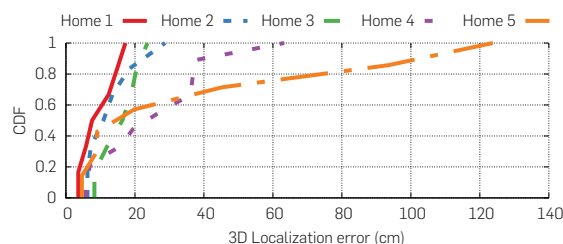


Figure 11. Accuracy by scenario. This plot shows a CDF of localization errors across different categories such as LOS placement and locations in closets.

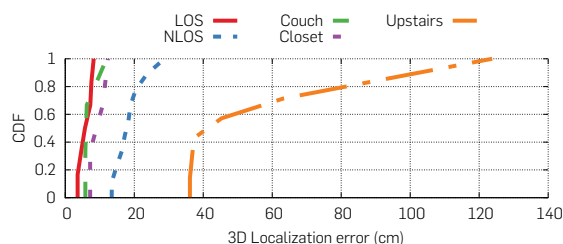
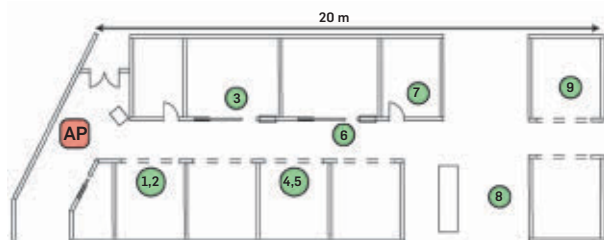


Figure 12. Hospital deployment. Deployment in a hospital surgery wing. We select nine points shown on the floor plan (left) including duplicates to test doors and curtains. We include patient pre-/post-op rooms (center) and storage facilities (right).



these locations do not have direct line of sight to the AP and include barriers such as curtains, sliding glass doors, and standard wooden doors.

Figure 13 shows the tracking accuracy across each of these locations that are ordered by distance. We note that the duplicated points represent separate measurements at the same location with barriers such as curtains or doors open and closed. Our system achieves a mean accuracy of 35.12 cm across all of the different locations in this hospital setting. Further, as is expected from our design, the accuracy scales with the distance to the tag: close by locations can achieve an error lower than 20 cm, whereas farther locations have localization error of 70 cm. These errors are small enough that we can track the equipment in the hospital across different rooms as well as the closet area. We note that in hospital post-op and pre-op settings the layout is typically on a single floor. Further, the barriers between the rooms are either curtains or thin doors. Thus, we can achieve high localization accuracy in this application setting.

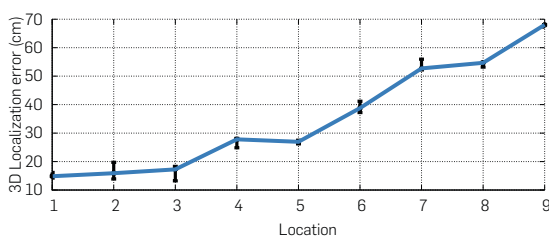
5. RELATED WORK

There has been recent interest in backscatter as a low-power communication mechanism. These techniques either backscatter existing TV,¹¹ Wi-Fi,^{8, 4} and FM signals²⁰ or generate Wi-Fi-compliant transmissions using techniques such as Passive Wi-Fi^{9, 7, 22} and FS-backscatter.²³ These Wi-Fi-based approaches have a receiver sensitivity of only -90 dBm and hence have a limited range and cannot work across rooms unless the signal source is placed close to the backscatter tag.¹⁰

There has also been recent interest in long-range backscatter solutions,^{17, 18} which¹⁷ achieve a longer range and are compatible with off-the-shelf LoRa radios. However, this prior work does not support localization. Further, existing implementation of LoRa backscatter requires FPGAs and consumes 5–10 mW of power. By contrast, we introduce a novel architecture that delegates the complex CSS coding operations to the AP and introduce a CSS backscatter design that has orders of magnitude lower power.

The closest related work is Slocalization,¹³ which backscatters UWB signals to achieve low-power localization. This design however works only with static scenarios and incurs delays on the order of minutes to hours to output the location value. This is because FCC regulations significantly limit the transmitted power of UWB signals compared to typical transmissions in ISM bands. Further, because the backscatter system in Pannuto et al.¹³ does not use coding such as the CSS modulation used in our design, it requires

Figure 13. Hospital accuracy. Localization accuracy results for each of the points marked in Figure 12.



integrating the received signal over 10 min to more than an hour, depending on the deployment, to get the location value. By contrast, our approach can provide the location value within 70 ms while achieving a range of 60 m and thus can support practical applications.

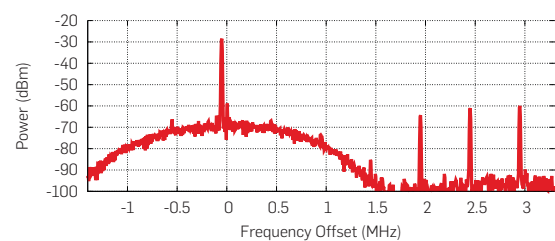
6. DISCUSSION AND CONCLUSION

We present the first wireless localization system that consumes microwatts of power in a subcentimeter form factor and can be localized across a whole home or hospital through walls. In this section, we outline limitations and avenues for future research.

Supporting multiple devices. In addition to using time-division multiplexing, we can also set different backscatter devices to shift the signals by different frequencies to support multiple devices as shown in Figure 14. Specifically, each backscatter device could use a different shift and reflect the incoming signal from the AP at the same time. The receiver can process the received signal across different shifts to concurrently localize multiple devices. More specifically, recent work has demonstrated that up to 256 devices can concurrently transmit using chirp spread spectrum.⁶ Because our design leverages chirp spread spectrum, one can design large-scale networks where devices can concurrently transmit and be localized at the same time.


Using multiple APs. In order to achieve high accuracies with a single AP, our algorithm relies on knowledge of the floor plan of a room to eliminate infeasible location estimates. We realize however that some applications may require more generalizable solutions, for example where the localized objects might be moved outside of a known set of rooms. The key reason for requiring these constraints however was that we use a single AP for localization. Adding additional access points, or additional antennas with greater separation, would provide better angular resolution and hence address this limitation. For example, an AP placed on the orthogonal wall of a building or on another floor in multifloor scenarios would help disambiguate multipath and provide more information to estimate the location. Future work could build upon the basic techniques we show here to explore the trade-off between the infrastructure overhead of adding additional APs and antennas versus the robustness and accuracy improvements they would contribute.

Figure 14. Feasibility of multiple devices. Snapshot of the chirp signal captured on a spectrum analyzer. The plot shows the baseband spectrum of the original coded transmissions as well as three backscattered signals at frequency offsets of 2, 2.5, and 3 MHz.



LOS path. Existing 3D localization algorithms assume that while there is multipath, there is at least some energy from the direct path at the receiver. Our design also makes a similar assumption. We note however that our design also increases the probability that the direct path signal has some energy by leveraging frequency diversity across the three ISM bands. Specifically, while the direct path signal could be weak at a specific frequency, it is likely to be noticeable at at least one of the three ISM bands.

Acknowledgments

This work was funded in part by the National Science Foundation and Google Faculty Research Awards. 

References

1. Lora modulation basics, 2016. <http://www.semtech.com/images/datasheet/an1200.22.pdf>.
2. Antennas, P.L.. W3320 ism868/915, ism2.4g, 2017.
3. Berni, A., Gregg, W. On the utility of chirp modulation for digital signaling. *IEEE Trans. Commun.* 6, 21 (1973), 748–751.
4. Bharadia, D., Joshi, K.R., Kotaru, M., Katti, S. Backfi: High throughput wifi backscatter. In *SIGCOMM '15* (2015).
5. Champion, L., Sornin, N. *Chirp signal processor*. European Patent Application EP2975814A1 (2014).
6. Hesar, M., Najafi, A., Gollakota, S. Netscatter: Enabling large-scale backscatter networks. In *NSDI'19* (2019).
7. Iyer, V., Talla, V., Kellogg, B., Gollakota, S., Smith, J. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016).
8. Kellogg, B., Parks, A., Gollakota, S., Smith, J.R., Wetherall, D. Wi-fi backscatter: Internet connectivity for rf-powered devices. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (2014).
9. Kellogg, B., Talla, V., Gollakota, S., Smith, J.R. Passive wi-fi: Bringing low power to wi-fi transmissions. In *13th USENIX Symposium on Networked Systems Design and Implementation* (NSDI 16, 2016).
10. Kotaru, M., Zhang, P., Katti, S. Localizing low-power backscatter tags using commodity wifi. In *CoNext'17* (2017).
11. Liu, V., Parks, A., Talla, V., Gollakota, S., Wetherall, D., Smith, J.R. Ambient backscatter: Wireless communication out of thin air. In *SIGCOMM '13* (2013).
12. Nandakumar, R., Iyer, V., Gollakota, S. 3d localization for sub-centimeter sized devices. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems* (2018), ACM, 108–119.
13. Pannuto, P., Kempke, B.P., Dutta, P. Slocalization: Sub-muw, static, decimeter-accurate localization with ultra wideband backscatter. In *IPSN* (2018), IPSN.
14. Seller, O.B., Sornin, N. *Low power long range transmitter*. US Patent 9,252,834 (2016).
15. Semiconductors, N. Kinetis k103 32kb flash, 2017.
16. Solutions, T.A. Ca.50 5150–5900 mHz ceramic chip monopole, 2017.
17. Talla, V., Hesar, M., Kellogg, B., Najafi, A., Smith, J.R., Gollakota, S. Lora backscatter: Enabling the vision of ubiquitous connectivity. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2017).
18. Varshney, A., Harms, O., Perez-Penichet, C., Rohner, C., Hermans, F., Voigt, T. Lorea: A backscatter architecture that achieves a long communication range. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, SenSys '17* (2017).
19. Vasisht, D., Kumar, S., Katabi, D. Decimeter-level localization with a single wifi access point. In *NSDI* (2016).
20. Wang, A., Iyer, V., Talla, V., Smith, J.R., Gollakota, S. FM backscatter: Enabling connected cities and smart fabrics. In *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 17* (2017).
21. Xiong, J., Jamieson, K. Arraytrack: A fine-grained indoor location system. In *NSDI* (2013).
22. Zhang, P., Bharadia, D., Joshi, K., Katti, S. Hitchhike: Practical backscatter using commodity wifi. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, SenSys '16* (New York, NY, USA, 2016), ACM, 259–271.
23. Zhang, P., Rostami, M., Hu, P., Ganesan, D. Enabling practical backscatter communication for on-body sensors. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016).

Rajalakshmi Nandakumar (rn283@cornell.edu), Cornell Tech.

Shyamnath Gollakota (gshyam@uw.edu), University of Washington.

Vikram Iyer (vsiyer@uw.edu), University of Washington.

© 2021 ACM 0001-0782/21/3 \$15.00

Concurrency

The Works of Leslie Lamport

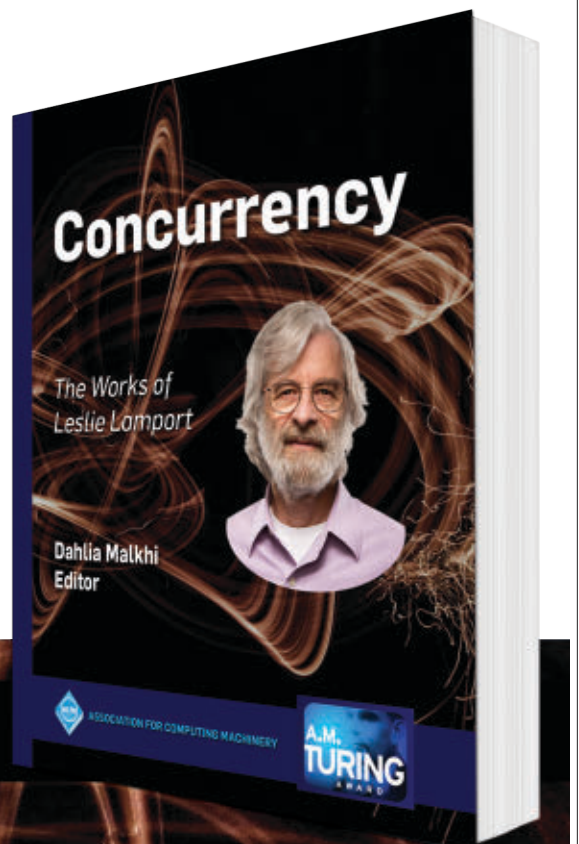
Dahlia Malkhi, Editor

ISBN: 978-1-4503-7271-8

DOI: 10.1145/3335772

<http://books.acm.org>

<http://store.morganclaypool.com/acm>



 **ACM BOOKS**
Collection II

CAREERS

The University of Virginia's College at Wise

Assistant Professor in Software Engineering

The University of Virginia's College at Wise's, a division of the University of Virginia, Math and Computer Science Department invites applications for a tenure-track faculty position in Software Engineering. The position is a full-time, 9-month appointment at the assistant professor level and with the potential for summer teaching and research support. Duties include creating and teaching courses in software engineering and related topics; serving the department and college on various committees; advising students; pursuing ongoing scholarly activity, particularly that involves undergraduates; and assisting the department chair with assessment and other duties as needed. In addition, as the only 4-year, public program in Software Engineering in Virginia and with expressed interest in software engineering graduates in the region and the Commonwealth of Virginia, the position opens the door for collaboration with industry and economic development officials.

Successful candidates must possess excellent verbal and written communication skills; should have the demonstrated ability for engaged and innovative teaching at the undergraduate level; must have a Ph.D. (ABD candidates will be considered but must have the Ph.D. completed by the date of appointment) in Computer Science, Computer Engineering, or Software Engineering, with at least 18 hours of graduate work in Software Engineering, and be eligible to work in the United States. Preference will be given to candidates who articulate a clear understanding of UVA Wise's public liberal arts mission and with relevant industry experience, particularly those who have potential to build a pipeline for internships and job placements for students.

Interested applicants for any of these positions should apply online at jobs.virginia.edu and complete a candidate profile that includes: 1) application, 2) cover letter, 3) current CV, 4) letter describing your teaching philosophy, 5) contact information for three references and 6) unofficial transcripts. Review of applications will begin in mid February 2021, but will remain open until filled. Start date for all positions will be August 2021. Applicants must be eligible to work in the United States.

The University of Virginia, including the UVA Health System and the University Physician's Group are fundamentally committed to the diversity of our faculty and staff. We believe diversity is excellence expressing itself through every person's perspectives and lived experiences. We are equal opportunity and affirmative action employers. All qualified applicants will receive consideration for employment without regard to age, color, disability, gender identity, marital status, national or ethnic origin, political affiliation, race, religion, sex (including pregnancy), sexual orientation, veteran status, and family medical or genetic information.

UVA Wise is committed to helping the campus community provide for their own safety and security. The Annual Security and Fire Safety Report containing information on campus security and personal safety, including alerts, fire safety, crime prevention tips, and crime statistics is available at www.uva-wise.edu/ASR. A copy is available upon request by calling 276-328-0190 or 276-376-3451.

University of Illinois at Chicago

Bridge to Faculty Postdoctoral - Computer Science

The UIC CS Department is recruiting a Postdoctoral Fellow from all areas of computer science to take part in UIC's Bridge to the Faculty program <https://diversity.uic.edu/engagement/bridge-to-the-faculty/>. This opportunity is designed to recruit scholars and support their scholarly development through a fully funded postdoctoral program of up to two years. Successful postdoctoral fellows will have the opportunity to transition to faculty following their fellowship experience. A successful candidate will demonstrate an understanding of barriers affecting populations traditionally underrepresented in the field of computer science and groups that are traditionally marginalized in the United States. The position is available in the Fall 2021 semester and the initial term of employment will be for up to 24 months.

Located in the heart of one of the most vibrant cities in the United States, UIC is a comprehensive urban public research (R1) university with a diverse student body and a strong tradition of support for difference and equality. UIC is among the nation's top five most diverse campuses.

This is a full-time position and includes a competitive salary and benefits package.

Applications must include:

► Cover letter addressing interest in the Fellowship, CS and UIC. Applicants are requested to include in their cover letter information about how they will further our goal of building a culturally diverse educational environment.

► CV

► Names and contact information for writers of three letters of recommendation; one must be from the dissertation committee chair or faculty advisor.

► A writing sample (dissertation proposal or publication).

For fullest consideration application materials must be submitted via <https://jobs.uic.edu/> by February 5th, and applications will continue to be accepted until the position has been filled.

The University of Illinois at Chicago is an Equal Opportunity, Affirmative Action employer. Minorities, women, veterans and individuals with disabilities are encouraged to apply.

Offers of employment by the University of Illinois may be subject to approval by the Univer-

sity's Board of Trustees and are made contingent upon the candidate's successful completion of any criminal background checks and other pre-employment assessments that may be required for the position being offered. Additional information regarding such pre-employment checks and assessments may be provided as applicable during the hiring process.

The University of Illinois System requires candidates selected for hire to disclose any documented finding of sexual misconduct or sexual harassment and to authorize inquiries to current and former employers regarding findings of sexual misconduct or sexual harassment. For more information, visit <https://www.hr.uillinois.edu/cms/One.aspx?portalId=4292&pageId=1411899>.



ADVERTISING IN CAREER OPPORTUNITIES

How to Submit a Classified Line Ad: Send an e-mail to acmm mediasales@acm.org. Please include text, and indicate the issue/or issues where the ad will appear, and a contact name and number.

Estimates: An insertion order will then be e-mailed back to you. The ad will by typeset according to CACM guidelines. NO PROOFS can be sent. Classified line ads are NOT commissionable.

Deadlines: 20th of the month/2 months prior to issue date. For latest deadline info, please contact:

acmm mediasales@acm.org

Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at: <http://jobs.acm.org>

Ads are listed for a period of 30 days.

**For More Information Contact:
ACM Media Sales,
at 212-626-0686 or
acmm mediasales@acm.org**

[CONTINUED FROM P. 128] this morning, and it still is now, okay? I haven't even done the upload yet."

"We wanted to ease you in gently. Your last complete memory before the upload was what you've just experienced. All this—it's a simulation."

"Yeah, right, Mark. You've had your joke. Stop it now."

"I'm going to show you. Try to remain calm. Please don't panic."

Diana felt the hairs standing up on the back of her neck. This was idiotic—Mark never knew when he'd gone too far with a joke.

And then the room faded out.

She was in a bare, white cell, just her and a chair. The walls, ceramic-looking, seemed to vibrate as if they were alive. "Mark, what have you done? Did you put something in my coffee?"

"This is real, Diana." The voice was the same, Mark's voice, but it came from the wall, which pulsed a brighter white as he spoke. "We need your help."

"Who is we? If you aren't Mark, who are you?"

"More *what* am I, really. We're an AI. We are the human race. A melding of uploads and pure AI. The future of humanity. But we need some assistance. From someone with a real body. You."

"Hold on, you said I was an upload. Shouldn't I be in there with you?"

"Like I said, we need someone with a physical body."

"No, I'm confused. I have ... had no intention of being frozen."

"You weren't—though I should point out that the preferred term is vitrification. Freezing destroys the cell structure. Ice crystals."

Diana shook her head. "This is crazy. Am I real?"

"You are embodied. It's just not ... your body."

"What?" Diana leapt up from her chair, staring down at herself.

"Here," said Mark. The wall in front of Diana flipped from white to reflective. She could see a woman, about her age. But with dark hair, not blonde. It wasn't her face.

"What have you done?" Diana said.

"We needed the right kind of person in a physical body. You fit the bill. A support tech. Ideal."

"Surely there have to be living technicians available? Why reanimate a body?"

"There isn't anyone."

"We're an AI. We are the human race. A melding of uploads and pure AI. The future of humanity. But we need assistance. From somebody with a real body. You."

"No tech staff? How's that possible."

"No people. Physicals. We're all in here now. It's much better in here."


"Okay, okay. Then, why not use a technician who was already frozen ... vitrified? A lot of the people who went in for cryonics in my time were from Silicon Valley. Why this Frankenstein mashup?"

"There's another problem." Mark sounded embarrassed. "We've kind of used them all up. This is the very last body. I'm sorry to push you, but there really isn't much time." A slip of plastic-like material exuded from the wall. "Just follow these instructions. Please."

Diana shrugged. "What the hell. Let's go down the rabbit hole." She followed a blue glowing ball that appeared in mid-air out of the room and down a long corridor. "What I don't get is why you don't just do this yourself. You must have physical extensions. Robots, drones, whatever you want to call them."

"Sure we do," said Mark, his voice alongside as if he were walking with her. "But we've a built-in restriction that prevents us doing what's necessary. It has to be someone external. You're our last hope."

"Okay," said Diana. "Here we go." There was a complex-looking control panel in the middle of the white wall. She reached out as instructed. "I'm ready," said Diana.

She switched humanity off and on again. 

Brian Clegg (www.brianclegg.net) is a science writer based in the U.K. His most recent books are *Are Numbers Real?*, an exploration of the relationship between math and reality, and *The Reality Frame*, an exploration of relativity and frames of reference.

© 2021 ACM 0001-0782/21/3 \$15.00

Distinguished Speakers Program

A great speaker can make the difference between a good event and a WOW event!

Students and faculty can take advantage of ACM's Distinguished Speakers Program to invite renowned thought leaders in academia, industry and government to deliver compelling and insightful talks on the most important topics in computing and IT today. ACM covers the cost of transportation for the speaker to travel to your event.

speakers.acm.org



Association for Computing Machinery

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/3446803

Brian Clegg

Future Tense Awakening

Some technical support will never change.

“HAVE YOU TRIED turning it off and on again? Okay, I’ll hold while you do.” Diana hit the mute icon on her iDesk and threw a screwed-up ball of paper at Mark, sitting opposite. “Why don’t we pre-record that? It’d save so much effort.”

Mark snorted. “I’d want a whole support phrasebook. Like, ‘Have you put in on charge?’ and ‘Where did you spill the coffee, exactly?’ and ‘Have you checked the FAQs?’ Of course, that’s just the first step. Next they’ll replace us with an AI—no deep learning required.”

Diana smiled. “Okay,” she said to the user, “don’t worry, it can take a while.” She waved her mug at Mark. “Coffee?”

“Yes, please.”

“I would, but ...” Diana gestured at her headset.

“I fell for that, didn’t I?” Mark reached over and grabbed Diana’s mug.

A flash of light momentarily filled the office, making Diana’s eyesight blurry. “Whoa!” She closed her eyes for a moment, then looked across at Mark, blinking to clear her vision. “What was that?”

“What that?”

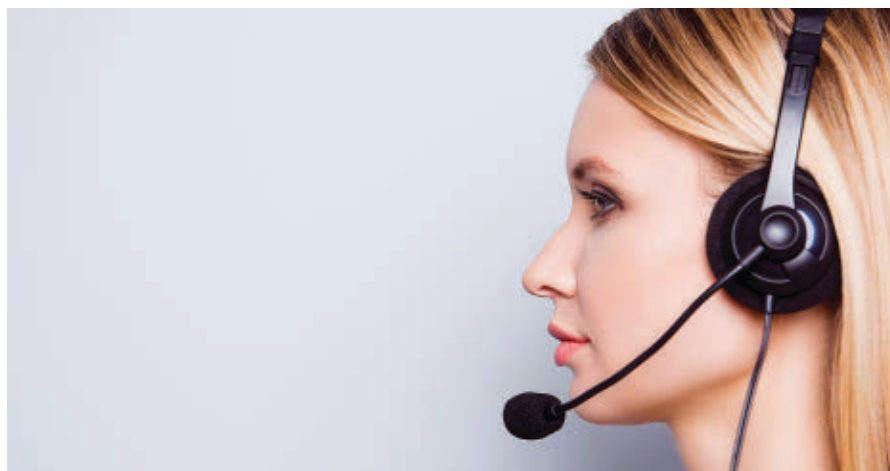
“That that. The flash—like lightning.” Diana peered out the dusty window, but there were blue skies, not a cloud in sight.

Mark shrugged. “I didn’t notice anything.”

“Well, I didn’t imagine it.”

Mark brought her coffee over. “Start of a migraine, maybe.”

“I don’t get... oh, hi.” Diana pointed to her headset in a familiar gesture. “It did? That’s great. Just call back if there are any more problems.” She touched the icon to release the call before mut-



tering, “And if I’m lucky, Mark’ll get you next time.”

“Did you ever read *The Sleeper Awakes*? You know, H. G. Wells.”

“Don’t pretend you’ve read it yourself, Mark. Unless there’s a graphic novel.”

“As it happens, I have read many books,” said Mark. “Many, many books. Including a number of fine titles by Wells. Everyone remembers *The Time Machine*, but not everyone knows he wrote two time travel books.”

“Okay, I’ll bite. What has this to do with anything?”

“Humor me.”

“Fine. I read a couple of his novels in college, but not that one.”

“You didn’t miss much. Over-heavy on political polemic, not enough action. This guy, the Sleeper, wakes up in the year 2100. Discovers he’s been out for 203 years.”

“Cryonics?”

“No, Wells was too early for that. The Sleeper was in a coma all that time. He becomes a kind of figurehead for the revolution.”

“And you’re telling me this because?”

“How would you feel if you found out that you had just woken up in the future?”

“Don’t mess with me, Mark.”

“No, really, I want to know.”

“I, er, I guess I’d be confused. I have no intention of becoming a corpsicle, so it’s not going to happen. It’s too weird, freezing yourself and hoping someone will revive you in the future.”

“I thought you’d volunteered for the MIT upload program?”

“Oh, yeah, but that’s different. I mean, even it works, it wouldn’t be me, just a kind of copy. In software.”

“But would the copy *know* it was just a copy?”

“You’re creeping me out, Mark. Hey, how long has it been since that last job? I don’t remember ever going this long between calls.”

“So, don’t freak out, but you *are* the Sleeper, okay? This is the future. I mean, your future. We’re 200 years ahead of your time.”

“Oh, come on, Mark. It was 2040 when I woke up [CONTINUED ON P. 127]



ACM BOOKS Collection II

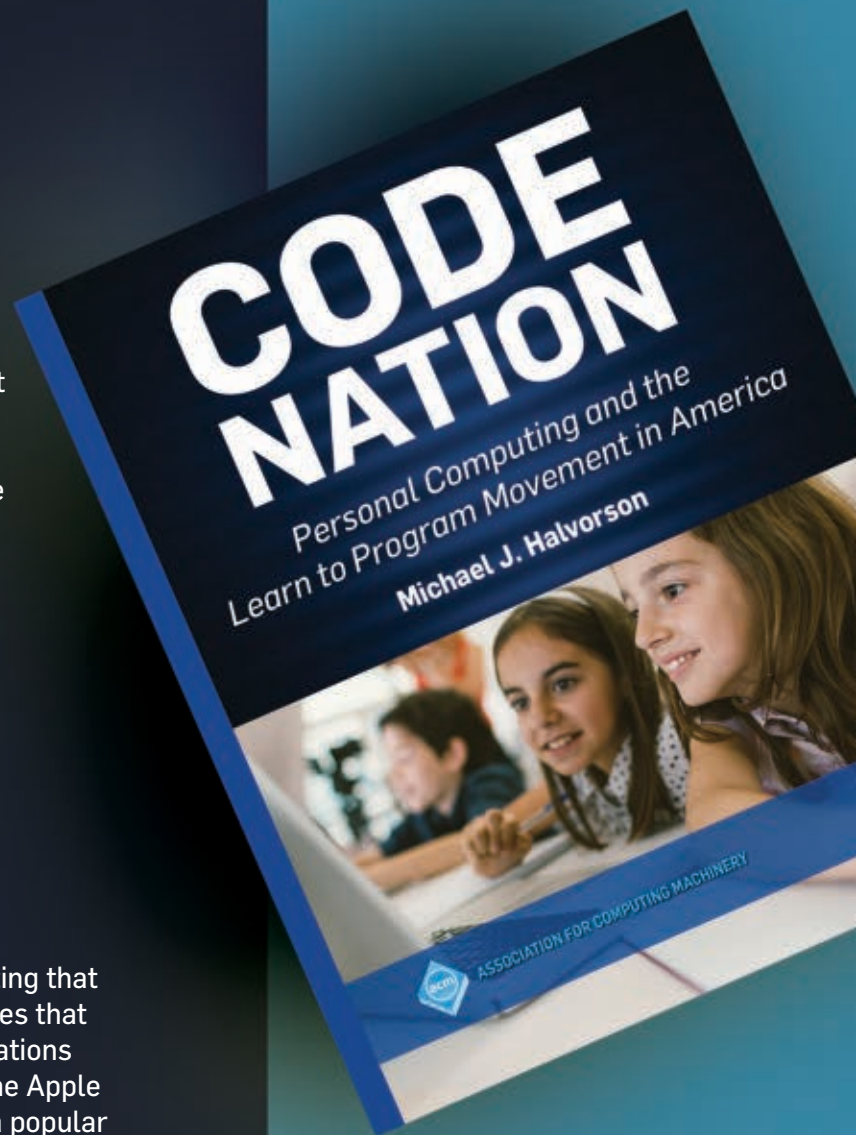
Code Nation explores the rise of software development as a social, cultural, and technical phenomenon in American history. The movement germinated in government and university labs during the 1950s, gained momentum through corporate and counterculture experiments in the 1960s and 1970s, and became a broad-based computer literacy movement in the 1980s. As personal computing came to the fore, learning to program was transformed by a groundswell of popular enthusiasm, exciting new platforms, and an array of commercial practices that have been further amplified by distributed computing and the Internet. The resulting society can be depicted as a “Code Nation”—a globally-connected world that is saturated with computer technology and enchanted by software and its creation.

Code Nation is a new history of personal computing that emphasizes the technical and business challenges that software developers faced when building applications for CP/M, MS-DOS, UNIX, Microsoft Windows, the Apple Macintosh, and other emerging platforms. It is a popular history of computing that explores the experiences of novice computer users, tinkerers, hackers, and power users, as well as the ideals and aspirations of leading computer scientists, engineers, educators, and entrepreneurs. Computer book and magazine publishers also played important, if overlooked, roles in the diffusion of new technical skills, and this book highlights their creative work and influence.

Code Nation offers a “behind-the-scenes” look at application and operating-system programming practices, the diversity of historic computer languages, the rise of user communities, early attempts to market PC software, and the origins of “enterprise” computing systems. Code samples and over 80 historic photographs support the text. The book concludes with an assessment of contemporary efforts to teach computational thinking to young people.

<http://books.acm.org>

<http://store.morganclaypool.com/acm>



CODE NATION

*Personal Computing and
the Learn to Program
Movement in America*

Michael J. Halvorson

ISBN: 978-1-4503-7757-7

DOI: 10.1145/3368274

Today's Research Driving Tomorrow's Technology

The ACM Digital Library (DL) is the most comprehensive research platform available for computing and information technology and includes the ongoing contributions of the field's most renowned researchers and practitioners.

Each year, roughly 20,000 newly published articles from ACM journals, magazines, technical newsletters and annual conference volumes are added to the DL's complete full text contents of more than 550,000 articles.

The DL also features the fully integrated and comprehensive bibliographic index, *The Guide to Computing Literature*—a continually updated index featuring millions of publication records from over 5,000 publishers worldwide.

For more information, please visit

<https://libraries.acm.org/>

or contact ACM at

dl-info@hq.acm.org

ACM

DL

DIGITAL
LIBRARY