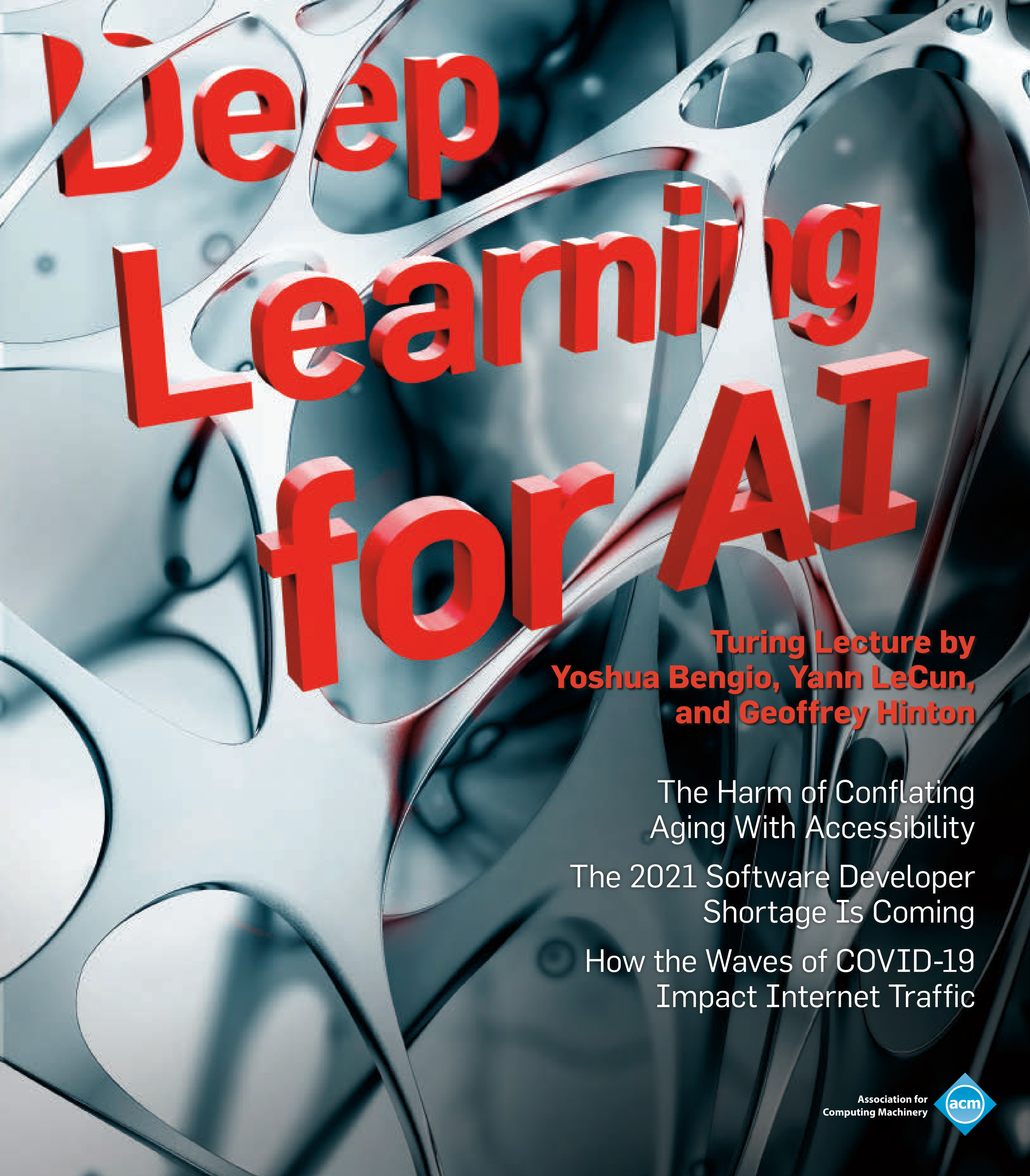
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

07/2021 VOL.64 NO.07



Deep Learning for AI

Turing Lecture by Yoshua Bengio, Yann LeCun, and Geoffrey Hinton

The Harm of Conflating Aging With Accessibility

The 2021 Software Developer Shortage Is Coming

How the Waves of COVID-19 Impact Internet Traffic

Association for Computing Machinery

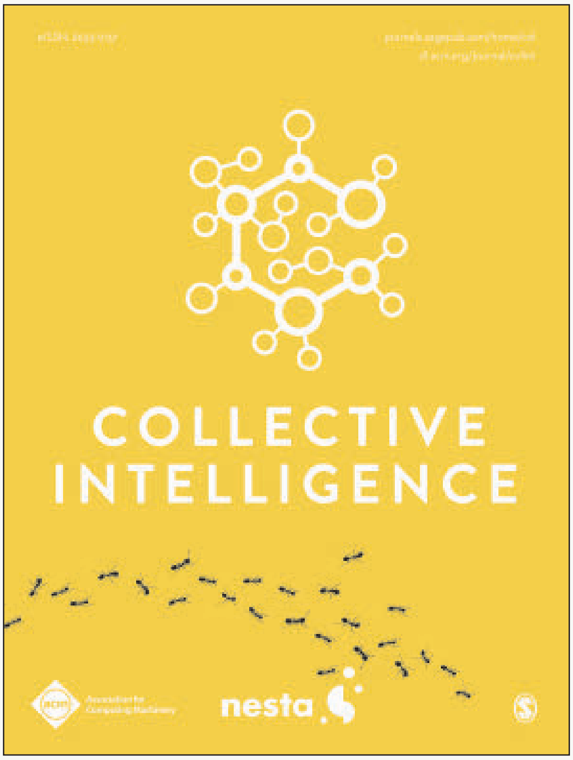
acm

SIGGRAPH ASIA 2021 TOKYO

CONFERENCE 14 - 17 DECEMBER 2021
EXHIBITION 15 - 17 DECEMBER 2021
TOKYO INTERNATIONAL FORUM, JAPAN

sa2021.siggraph.org

LIVE

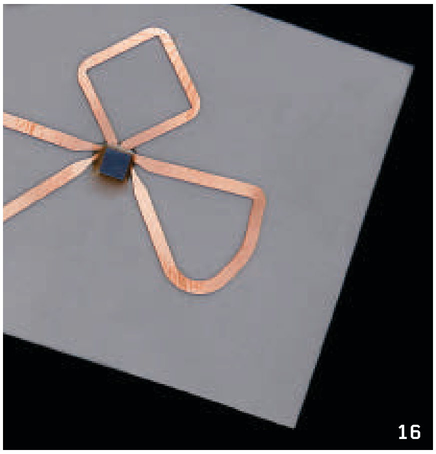

COLLECTIVE INTELLIGENCE
nesta

COMMUNICATIONS OF THE ACM

Departments

Last Byte

News



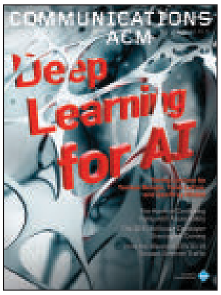
Viewpoints

PHOTO COURTESY OF WILIOT.COM/PRESS

07/2021
VOL. 64 NO. 07

Practice

Articles' development led by acmqueue
queue.acm.org



About the Cover:
In their Turing Lecture, Yoshua Bengio, Yann LeCun, and Geoffrey Hinton—recipients of the 2018 ACM A.M. Turing Award—address recent breakthroughs and future challenges of deep learning. Cover by Andrij Borys Associates, using image by Yurchanka Siarhei.

IMAGE BY TIMOFEEV VLADIMIR

Contributed Articles

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

Watch the authors discuss this work in the exclusive *Communications* video.
https://cacm.acm.org/videos/aging-accessibility

Review Articles

Research Highlights

Association for Computing Machinery
Advancing Computing as a Science & Profession

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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

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

STAFF

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

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Ralph Raiola
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
Ilia 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
Ilia Rodriguez
ilia.rodriguez@hq.acm.org

Media Kit acmmediasales@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

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

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/3469113

Moshe Y. Vardi

Program Verification: Vision and Reality

IN 1969, TONY HOARE published a classical *Communications*' article, "An Axiomatic Basis for Computer Programming." Hoare's article culminated a sequence of works by Turing, McCarthy, Wirth, Floyd, and Manna, whose essence is an association of a proposition with each point in the program control flow, where the proposition is asserted to hold whenever that point is reach.

Hoare added two important elements to that approach. First, he described a formal logic, now called *Hoare Logic*, for reasoning about programs. Second, he offered a compelling vision for the program-verification project: "When the correctness of a program, its compiler, and the hardware of the computer have all been established with mathematical certainty, it will be possible to place great reliance on the results of the program, and predict their properties with a confidence limited only by the reliability of the electronics."

Hoare's vision came under a scathing attack a decade later in an influential 1979 *Communications*' article, "Social Processes and Proofs of Theorems and Programs," by De Millo, Lipton, and Perlis. They argued that mathematical proofs are accepted through a social process. Program-correctness proofs will not be subject to a similar social process, due to their length and narrowness, so they will not be socially accepted. They concluded that "this makes the formal verification process difficult to justify and manage." Hoare himself retracted, to some extent, his 1969 vision in 1995, writing "It has turned out that the world just does not suffer significantly from the kind of problems that our research was originally intended to solve."

In a parallel development, Amir Pnueli introduced the temporal logic of programs in 1977. Clarke and Emerson, and independently, Queille and Sifakis, then built on Pnueli's work and developed, in the early 1980s, *model checking*, an algorithmic technique for checking properties of finite-state programs. That led to Pnueli receiving the ACM A.M. Turing Award in 1966, and Clarke-Emerson-Sifakis receiving the award in 2007. By the mid-1990s, several model checkers had been built and adopted for industrial usage by semiconductor and design-automation companies. Industrial temporal logics, such as PSL and SVA, based on Pnueli's work, became industry standards in the early 2000s.

The success of model checking in the semiconductor industry, where post-production error correction is very difficult, points to an important insight that was missing in the early literature on program verification. Program verification is an expensive activity. Navigating the cost-benefit trade-off of program verification is ultimately a business decision. Model checking offered a different price point than full program verification: on one hand, model checking offers less—property checking and not full program verification, on the other hand, model checking costs less, due to higher level of automation.

This cost-benefit trade-off suggests that how much verification should be done is context dependent. An operation-system microkernel is probably a more appropriate target for a major verification effort than a dating app. Such an undertaking was initiated by an Australian team, who verified the seL4 microkernel using the Isabelle proof-assistant tool. Isabelle is based on a small logical core to increase the trustworthiness of proofs. This approach acknowledges that De Millo et al. were right—proofs do require a social process to be accepted—but in Isabelle (and similar tools) this social process can be confined to the small logic core. In fact, with the help of proof assistants, formal verification today is even bringing a new standard for rigor in mathematics.

The emergence of cloud computing as the major context for much of today's computing shifts the cost-benefit trade-off of verification, due to its large scale. Because different users of the same cloud platform share hardware resources, security and privacy are of paramount interest. The Automated-Reasoning Group at Amazon Web Service (AWS) has been focusing on the development and use of formal-verification tools at AWS to increase the security assurance of its cloud infrastructure and to help customers secure themselves. At the same time, as the Spectre and Meltdown attacks have demonstrated, the large gap between the logical model (ISA) and the underlying microarchitecture of the X86 microprocessor not only provides side channels to attackers but also erects a major barrier to full verification.

In 1969, Hoare wrote about mathematical certainty, great reliance, and confidence. In retrospect, the hope for "mathematical certainty" was idealized, and not fully realistic, I believe. Verification can give us great reliance and confidence, but at a cost that must be justified by the benefits. The deployment of autonomous systems with machine learning-based components brings new urgency and excitement to this important research area.

Follow me on Facebook and Twitter.

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*.

SHAPE THE FUTURE OF COMPUTING.
JOIN ACM TODAY.

www.acm.org/join/CAPP

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

❑ Professional Membership: $99 USD

❑ Professional Membership plus ACM Digital Library: $198 USD
($99 dues + $99 DL)

ACM STUDENT MEMBERSHIP:

❑ Student Membership: $19 USD

❑ Student Membership plus ACM Digital Library: $42 USD

❑ Student Membership plus Print *CACM* Magazine: $42 USD

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

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

PAYMENT INFORMATION

Name

Mailing Address

City/State/Province

ZIP/Postal Code/Country

❑ Please do not release my postal address to third parties

Email Address

❑ Yes, please send me ACM Announcements via email

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

❑ AMEX ❑ VISA/MasterCard ❑ Check/money order

Credit Card #

Exp. Date

Signature

Purposes of ACM

ACM is dedicated to:

1) Advancing the art, science, engineering, and application of information technology

2) Fostering the open interchange of information to serve both professionals and the public

3) Promoting the highest professional and ethics standards

By joining ACM, I agree to abide by ACM's Code of Ethics (www.acm.org/code-of-ethics) and ACM's Policy Against Harassment (www.acm.org/about-acm/policy-against-harassment).

I acknowledge ACM's Policy Against Harassment and agree that behavior such as the following will constitute grounds for actions against me:

- Abusive action directed at an individual, such as threats, intimidation, or bullying
- Racism, homophobia, or other behavior that discriminates against a group or class of people
- Sexual harassment of any kind, such as unwelcome sexual advances or words/actions of a sexual nature

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.

acm Association for Computing Machinery

ACM General Post Office
P.O. Box 30777
New York, NY 10087-0777

1-800-342-6626 (US & Canada)
1-212-626-0500 (Global)
Hours: 8:30AM - 4:30PM (US EST)

Fax: 212-944-1318
acmhelp@acm.org
acm.org/join/CAPP

CAREER PATHS IN COMPUTING

DOI:10.1145/3464919

Computing enabled me to . . .

Obtain a Ph.D. and a Career in Data

NAME
Victoria Holt

BACKGROUND
Born in Yorkshire, now in Bath

CURRENT JOB TITLE/EMPLOYER
Enterprise Data Architect; Honorary Visiting Fellow, The Open University; FBCS

EDUCATION
BSc, Ph.D. The Open University, U.K.

INNOVATIVE NEW APPROACHES to learning are enabling people to gain skills that are vital in today's workplace, and I am proud to consider myself one of those people. Without much of a background in computing, I ended up taking a distance learning path, which allowed me to work with data and information that could be used for actionable intelligence.

I came from a school that had no computers nor were there any staff to advise me on a future career. It was only after working in offices that I was introduced to computing and the value of data. I quickly became hungry to learn more and the opportunity to study part-time via distanced learning at the Open University led me to a diploma in Systems Practice, which provided fascinating insight into systems thinking. Gaining practical experience at work while continuing with part-time study to complete my degree in IT and Systems Complexity was hugely beneficial to my overall appreciation and understanding of how best to put my newfound skills to use.

Next, I landed a job as a database administrator (DBA) where I became proficient at managing database systems by studying for many Microsoft data platform certifications. Much of this was possible through online self-study courses. The Microsoft database community provided an excellent opportunity to network with other professionals, to share knowledge, and to learn. I discovered many of us faced the same issues and problems. My passion for research, systems thinking, and the database industry drove me to try and discover why so many people shared these same problems.

I was fortunate that the Open University and my employer supported my pursuit of a part-time research Ph.D. Looking back, this was no small feat—not only was I working full time, but I was also a part-time researcher, and I was continuing to help others in the database community. The management of database systems is complex and each of the existing methodologies only cover a part of the problem at hand.

My doctoral thesis, "A Study into Best Practices and Procedures used in the Management of Database Systems," is a sociotechnical in-depth study on the complexities involved in the management of database systems. After completing my Ph.D., I was honored to receive the AOUG Will Swann Award for Innovation and Knowledge Development (2016) and a few years later the Microsoft Most Valuable Professional Award (2018–2021) for technology experts who passionately share their knowledge with the community for three years in a row. Along my journey, I have become a Fellow of the British Computer Society and was listed in the Top 100 Most Inspirational Women in the West during 2019.

I love putting my own experiences to use to help improve the relationship between academia and industry to solve real-world problems and business use cases. Data is transformative and required for businesses to grow and innovate. Organizations must continually invest and work to upgrade and manage all their data assets. Building data use cases is a first step in the road to accessing the power of data in a complex world. Incorporating data governance and striving for good data quality is crucial for almost any successful business plan.

In conclusion, I would like to encourage young scientists to investigate different options available for learning. Believe in yourself, your ambition, and try not to be discouraged. It's never too late to study in a way that suits you. Due to mandatory training courses requiring regulatory compliance knowledge, e-learning and distance learning are rising in prominence. I've found more companies are encouraging employees to upgrade their skills through courses provided by Coursera, Udemy, edX, Google Cloud tutorials, IBM Online Academy, Microsoft Azure e-learning tutorials and more.

Due to COVID-19 and the need to stay at home, I believe many people have become more receptive to the importance and ease of upgrading skills to stay relevant during this fast-paced digital revolution. My career to date has taught me to aim to be extraordinary. It is never too late to shine and there are many ways to learn! No one's path is the same.

DOI:10.1145/3466562

Two Sides of the Software Engineering Coin

My observations relate to two columns from the January 2021 issue of *Communications*: Michael A. Cusumano's "Technology Strategy and Management" and Thomas Haigh's "Historical Reflections." Reading one column right after the other, I could not help but notice the stark contrast between Haigh's and Cusumano's accounts with respect to the engineering profession. Whereas, on one side, there is a glorification of the "pure and noble" ethos, on the other side there is perhaps the saddest statement, when engineers brag in email about how they "'tricked' the FAA regulators."

While there are certainly human idiosyncrasies involved, this discrepancy shows to me that engineers adapt their belief system to the frame in which they operate: if you cannot get recognition from management for classical engineering skills, like safety and longevity, engineers—in desperation?—will adapt their ethos accordingly.

With the financial crises, we have seen how rapidly the perception of a profession can deteriorate: the word "bankster" can be found now in the most important German dictionary, the Duden. Let's hope, we won't see "encheat-eer" anytime soon. Unfortunately, with the German automakers' diesel manipulations and Boeing's 737 MAX, we are already a good way down that path. I am afraid the public's changing perception and the critical scrutiny of (computer) engineers will come to haunt our profession in the future … .

Holger Kienle, Berlin, Germany

Authors' Response:

I fully agree with Kienle's comment that "engineers adapt their belief system to the frame in which they operate." Sometimes companies require this to create a successful business, but the consequences can be deadly. Let's look at Thomas Haigh's column on the book, *The Soul of a New Machine*, in that light. Yes, the book is an inspiring account of the startup culture at Data General and the development of a new minicomputer model. However, what the book and column do not discuss is that, essentially, the project built the wrong product at the wrong time. The minicomputer business was already under threat from high-end workstations and soon would be permanently disrupted by personal computers. Data General had to shift to making data storage equipment and then was acquired by EMC. We see no evidence in this story that the engineers and managers understood the business context and how technology and markets were changing. We might view the 737 MAX debacle in this context, but without making excuses for Boeing's mistakes. Boeing was struggling to catch up with Airbus and retrofitted an old product for a hot new market segment. Both managers and engineers made decisions that would cost hundreds of lives and billions of dollars in losses, and severely damage Boeing's reputation for engineering excellence and safety. I think the lesson is that both managers and engineers need to understand the financial or competitive pressures in their business, but, in these situations, engineers in particular need to resist compromising their training. Technical experts know how to estimate risk and the probability of catastrophic failure, even though they are subject to the same human frailties as everyone else.

Michael A. Cusumano,
Cambridge, MA, USA

Even in Kidder's romantic portrayal, the Data General engineers rushed the design process to get the machines out the door quickly, but the minicomputers they were producing were not as safety-critical as the systems aeronautical engineers deal with. Maybe the problem at Boeing was management's urge to treat planes like other devices.

Thomas Haigh, Milwaukee, WI, USA

I read with great interest Thomas Haigh's discussion of *The Soul of a New Machine* by Tracy Kidder in the January, 2021, issue of *Communications*. I have fond memories of being in a packed auditorium at Purdue University in the early 1980s to hear Kidder discuss his recent book. Because his appearance was organized by the English Department, the head of the English Department introduced him and told us how he selected Kidder as a guest speaker. He had read a review of the book and gone to the local bookstore to buy it only to find that the book was out of stock. When he asked who was buying the book, he was told that "computer people" were the customers. If the book was so popular at Purdue, he reasoned, hosting Kidder as a guest speaker would definitely help the English Department.

Kidder was an excellent speaker. I remember two comments he made with respect to the book. He said when he talked to the Eagle project's engineers, he discovered every one of them had been required to take one or more English or literature courses in college. Kidder added that he, as an English major, had not been required to take any computer science or engineering courses when he was in college and said that, if he were in charge, he would make a computer science or engineering course a requirement for every English major.

The second comment involved his process for writing the book. In exchange for being allowed access to the Eagle project, Kidder agreed that the project members could review the manuscript before it was published. Many of the engineers took advantage of this opportunity to review the manuscript and suggested changes. Kidder was pleasantly surprised that not one of the changes involved the descriptions of the people he had written about, even when he may have presented them in an unflattering manner. His engineer-manuscript-readers only suggested changes to improve the technical accuracy of the book, and Kidder felt that this attention to technical accuracy was a major contributor to the book's success.

Herbert Schwetman, Austin, TX, USA,
Member of ACM since 1965

Another Component of the Human Mind

Information compression (IC) is a surprising omission from the features of the human mind described by Gary Marcus and Ernest Davies (*Communications*, Jan. 2021). Research into the role of IC in human learning, perception, and cognition was pioneered by Fred Attneave and Horace Barlow and others in the 1950s and has continued up to the present. There is a recent review in Wolff.[3]

A possible reason for IC to be overlooked as a unifying principle in the human mind is that it can be hidden in plain sight. For example:

- Imagine that you are viewing a scene, then close your eyes for a moment, and then open them again. What do you see? You see the same scene as before, perhaps with small changes. This means you have merged the 'before' and 'after' views and thus compressed them. This is entirely different from an old-style cine camera which makes no attempt to merge the two views. The merging of the two views is shown schematically in Figure 8 in Wolff.[3]
- A popular technique for IC is to use a relatively short identifier or 'code' to stand for a relatively large 'chunk' of information. A little reflection will show that, in natural language, every noun, verb, adjective, and adverb may be seen to function as such a code. A word like 'house' represents a relatively complex concept, with doors, windows, walls, etc. In short, IC is a pervasive feature of natural languages, and how we use them. Among other evidence for the importance of IC in the human mind is that, in an AI system founded on IC,[2] several different aspects of intelligence flow from it without ad hoc programming.

It is true as Marcus has argued[1] that in many respects the human mind is a kluge, perhaps because of the haphazard nature of evolution. But it is possible, at the same time, that IC can be a unifying principle in understanding the human mind.

References

1. Markus, G. *Kluge: The Haphazard Construction of the Human Mind*. Faber and Faber, London, U.K., 2008.
2. Wolff, J.G. The SP theory of intelligence: An overview. *Information 4*, 3 (2013), 283–341; extracted from *Unifying Computing and Cognition*.
3. Wolff, J.G. Information compression as a unifying principle in human learning, perception, and cognition. *Complexity* (2019), Article ID 1879746; doi.org/10.1155/2019/1879746.

Gerry Wolff, Menai Bridge, U.K.

Metaphor Model

Regarding "Disputing Dijkstra" by Mark Guzdial (March 2021), although I am not enough of a philosopher to confirm his argument for metaphors in any academic sense, there does seem to be considerable value in the concept. And there is of course the universal metaphor—"It's turtles all the way down!"

An interesting point raised by Dijkstra is the notion that software spans many layers of abstraction. I have often thought of software as a way to "make your own physics"; that is, you get to construct the behavior of objects from the lowest to the highest levels. And you can change the charge on an electron a little, if you like, and further you can change it only on Tuesdays, or when the sun is shining and the date is a prime number. This then raises the problem of layering violations which allow some relatively small change in behavior at the low level to inordinately affect behavior at the high level.

While we can identify areas of physics, biology, and other sciences where this occurs—for example nuclear radiation can interrupt the biological hierarchy—in software it seems to be everywhere. Analogies to physical models such as this can be useful in assessing, for example, system structure and modularity.

So, while the idea that metaphors are an essential guide seems well-established, looking for ways in which software and computer science does not fit into our common mental models is, I think, an important and useful exercise. We don't necessarily need to do so with an eye toward abolishing the metaphorical models, but we should be on the lookout for new metaphors that help us describe and explicate the behavior of our systems.

Larry Stabile, Cambridge, MA, USA

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

Coming Next Month in COMMUNICATIONS

The Frama-C Software Analysis Platform

Unveiling Unexpected Training Data in Internet Video

Multimedia Data Delivery Based on IoT

Scaling Up Chatbots for Corporate Service

PL and HCI: Better Together

WebRTC: Real-Time Communication for the Open Web Platform

Biases in AI Systems

A Wearable System for Blood Pressure Monitoring From User's Ear

Optimal Auctions Through Deep Learning

Responsible AI

Science Needs to Engage With Society

Recruit Domestic Computer Science Students

Plus the latest news about better security through obfuscation, fixing the Internet, and the unionization of technology.

BLOG@CACM

The *Communications* website, 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:10.1145/3464931 http://cacm.acm.org/blogs/blog-cacm

Securing Seabed Cybersecurity, Emphasizing Intelligence Augmentation

John Arquilla considers the outlook for undersea cyberwar, while Judi Fusco, Pati Ruiz, and Jeremy Roschelle discuss how building equitable applications in education requires an emphasis on augmenting intelligence.

John Arquilla
From U-boats to 'U-bots'

https://bit.ly/3nnBWrl

April 26, 2021

Of all the perils he faced during World War II, Winston Churchill said German submarine wolfpacks were his greatest concern, because their attacks on merchant ship convoys threatened to choke Britain's economic lifelines. Today, it seems there is another emerging undersea threat, one that has the potential to disrupt the global economy by severing fiber-optic lines of communication that run along the world's various seabeds.

There are nearly 400 undersea cables that stretch for almost three-quarters of a million miles, the densest concentrations of them being in the North Atlantic and the North Sea, the Mediterranean, and in Southeast Asia and around Japan. They carry virtually all (97%) international communications, and their exact locations are reasonably well-known. They are also increasingly vulnerable to being tapped or even cut by advanced submarine craft of a range of types, from manned mini-subs to remotely operated undersea drones, and even fully autonomous "U-bots."

The Russian Navy seems to have taken to heart the late historian John Keegan's statement, in his *Price of Admiralty* (https://amzn.to/2Sg2nn5), that by the 1980s the submarine had become more important than the aircraft carrier as an instrument of sea power. Russia's undersea capabilities are exceptional and include a range of vessel types that can approach even cables located at great depths, thanks to the operating capabilities of their U-bots. Admiral Nikolai Yevmenov, the overall Russian naval commander, is himself a deeply experienced submariner. His forces reflect his expertise.

Given the vast majority of the world's international communications still run via wires, any country with a capability for tapping into or severing the undersea cables that drive globalization is a very serious concern. In Russia's case, possible "mass disruption" of this sort would be a less grave matter given its much lower dependence upon undersea cables than other countries. Russia can, therefore, be viewed as having a strategic advantage in this aspect of cyberwar, which in this form is about conducting physical attacks upon or

"Any country with a capability for tapping into or severing the undersea cables that drive globalization is a very serious concern."

exploitations of critical information infrastructure.

This concern has led NATO member states, for example, to establish initiatives for the protection of this essential—and almost entirely privately owned—element of the "global commons." Indeed, by September of this year there are intended to be two naval commands up and running—one in the U.S., the other in Europe—that will have the principal responsibility for the defense of undersea cables.

Beyond such military measures, it seems to me this is a situation that calls for diplomacy as well. The world community does not hesitate to craft agreements controlling production and intending to ban use of weapons of mass destruction. So, too, there should be no hesitation about putting limits on those things able to cause "mass disruption." Because the U-bot threat is directed at information systems, it should be seen as falling under the rubric of cyberwar. Like the other weapons that operate in this realm, in and out of cyberspace, there is very little probability of reaching an arms control agreement to prevent their further development. This still leaves open the possibility of crafting behavior-based agreements, binding on all, to refrain from interfering with global communications that flow through the world's undersea cables.

During his second term, President Barack Obama met with President Xi Jinping to discuss the possibility of reaching an agreement to refrain from attacking critical information infrastructures. Both leaders saw it as in the interest of the U.S. and China to pursue such an agreement, but momentum was lost in recent years. It is time now to rekindle that kind of creative thinking about how to secure the global commons. Along with many other nations, I believe the Russians would also join in support of such an initiative. My belief derives from my early personal experience (back in the '90s) with Russian cyber experts who *introduced* the possibility of cyber arms control in the week-long session we had together.

The alternative? Continue to grow a global economy increasingly dependent on ever-more-vulnerable lines of communication. Simply put, an unacceptable risk.

Judi Fusco, Pati Ruiz, and Jeremy Roschelle

AI or Intelligence Augmentation for Education?

https://bit.ly/2RrMgCp
March 15, 2021

On December 7, 1968, Douglas Engelart presented a demonstration (https://bit.ly/3xM7ZpG) that showed how newly emerging computing technologies could help people work together. More generally, Engelbart devoted his professional life to articulating his view of the role of computing in addressing societal problems. He emphasized the potential for technology to augment (https://bit.ly/3th3ik3) human intelligence. Since that time, many others have developed the concept of intelligence augmentation (IA).

For example, the field of healthcare sees IA as a more ethical framing. One report (https://bit.ly/3b14X77) defines IA as "an alternative conceptualization that focuses on AI's assistive role, emphasizing a design approach and implementation that enhances human intelligence rather than replaces it." This report argues "health care AI should be understood as a tool to augment professional clinical judgment."

In education, applications of artificial intelligence are now rapidly expanding. Not only are innovators developing intelligent tutoring systems (https://bit.ly/3aZMpUA) that support learning how to solve tough Algebra problems, AI applications also include automatically grading essays or homework (https://bit.ly/3egrRt6), as well as early warning systems (https://eric.ed.gov/?id=ED594871) that alert administrators to potential drop-outs. We also see AI products for online science labs that give teachers and students feedback. Other products listen to classroom discussions and highlight features of classroom talk that a teacher might seek to improve or observe the quality of teaching in videos of preschool children. A recent expert report (https://bit.ly/3vHN6tW) about AI and education uncovered visions for AI that would support teachers to orchestrate classroom activities, extend the range of student learning outcomes that can be measured, support learners with disabilities, and more.

"We think it's important to always have the human in the loop to understand if things are working and, if not, to understand why and make creative plans for change."

In colloquial use, the term AI calls forth images of quasi-human agents that act independently, often replacing the work of humans, who become less important. AI is usually faster and based on more data, but is it smarter? In addition, there are difficult problems of privacy and security—society has an obligation to protect children's data. And there are even more difficult issues of bias, fairness, transparency, and accountability. Here's our worry: a focus on AI provides the illusion that we could obtain the good (superhuman alternative intelligences) if only we find ways to tackle the bad (ethics and equity). We believe this is a mirage. People will always be intrinsic to learning, no matter how fast, smart, and data-savvy technological agents become. People are why agents exist. We think it is important to always have the human in the loop to understand if things are working and, if not, to understand why and make creative plans for change.

Today, students and teachers are overwhelmed by the challenges of teaching and learning in a pandemic. The problems we face in education are whole child problems. Why are parents clamoring to send children back to school? It's not just so they can get some work done! Learning is fundamentally social and cultural; enabling the next generation to construct knowledge, skills, and practices they will need to thrive is work

"We recommend a focus on IA in education that would put educators' professional judgment and learners' voice at the center of innovative designs and features."

that requires people working together in a learning community. Schools also provide needed social and emotional support. We are simultaneously at a critical juncture where the need to address ethics and equity are profound. In addition to trust and safety considerations, prioritizing the impact, and understanding how it changes interactions and what those implications are for students and teachers is essential when evaluating AI or any technology.

Thus, we recommend a focus on IA in education that would put educators' professional judgment and learners' voices at the center of innovative designs and features. An IA system might save an educator administrative time (for example, in grading papers) and support their attention to their students' struggles and needs. An IA system might help educators notice when a student is participating less and suggest strategies for engagement, perhaps even based on what worked to engage the student in a related classroom situation. In this Zoom era, we also have seen promising speech recognition technologies that can detect inequities in which students have a voice in classroom discussions over large samples of online verbal discourse. In some forward-looking school districts, teachers have instructional coaches. In those situations, the coach and teacher could utilize an IA tool to examine patterns of speaking in their teaching and make plans to address inequities. Further, the IA tool might allow the coach and teacher to specify smart alerts to the teacher—for example, for expected patterns in future classroom discussions that would signal a good time to try a new and different instructional move. Later, the IA tool might make a "highlights reel" that the coach and teacher could review to decide whether to stay with that new instructional move, or to try another.

The important difference between AI and IA may be *when* an educator's professional judgment and student voice are in the loop. The AI perspective typically offers opportunities for human judgment before technologies are adopted or when they are evaluated; the IA perspective places human judgment at the forefront throughout teaching and learning and should change the way technologies are designed. We worry the AI perspective may encourage innovators to see ethics and equity as a barrier they have to jump over once, and then their product is able to make decisions for students autonomously. Alas, when things go wrong, educators may respond with backlash that takes out both the bad and the good. We see the IA perspective as acknowledging ethics and equity issues in teaching and learning as ongoing and challenging.

By beginning with the presumption that human judgment will always need to be in the loop, we hope IA for education will focus attention on how human and computational intelligence could come together for the benefit of learners. With IA, restraint is built into the design and technology is not given power to fully make decisions without a diverse pool of humans participating. We hope IA for education will ground ethics and equity not in a high-stakes disclosure/consent/adoption decision, but rather in cycles of continuous improvement where the new powers of computational intelligence are balanced by the wisdom of educators and students.

John Arquilla is Distinguished Professor of Defense Analysis at the U.S. Naval Postgraduate School; his forthcoming book is *Bitskrieg: The New Challenge of Cyberwarfare*. The views expressed herein are his alone. **Judi Fusco** (jfusco@digitalpromise.org) is a Senior Researcher focusing on STEM teaching and learning at Digital Promise. **Pati Ruiz** (pruiz@digitalpromise.org) is a Computer Science Education Researcher at the non-profit Digital Promise. **Jeremy Roschelle** is Executive Director of Learning Sciences Research at Digital Promise and a Fellow of the International Society of the Learning Sciences.

N news

Science | DOI:10.1145/3464933

Samuel Greengard

Formal Software Verification Measures Up

Verified coding techniques use mathematical proofs to ensure code is error-free and hacker-resistant. Can the approach revolutionize software?

THE MODERN WORLD runs on software. From smartphones and automobiles to medical devices and power plants, executable code drives insight and automation. However, there is a catch: computer code often contains programming errors—some small, some large. These glitches can lead to unexpected results—and systematic failures.

"In many cases, software flaws don't make any difference. In other cases, they can cause massive problems," says Kathleen Fisher, professor and chair of the computer science department at Tufts University and a former official of the U.S. Defense Advanced Research Projects Agency (DARPA).

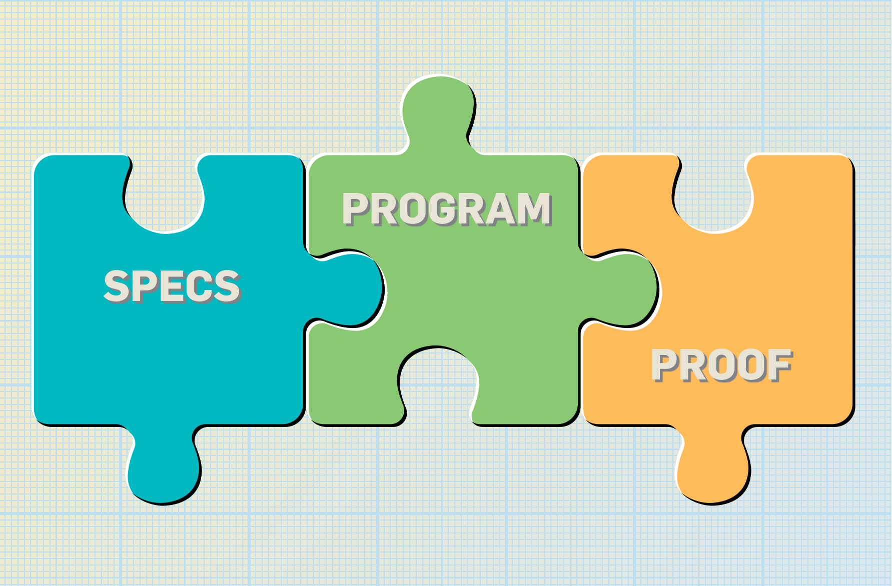


IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

For decades, computer scientists have imagined a world where software code is formally verified using mathematical proofs. The result would be applications free from coding errors that introduce bugs, hacks, and attacks. Software verification would ratchet up device performance while improving cybersecurity and public safety. By applying specialized algorithms and toolkits, "It's possible to show that code completely meets predetermined specifications," says Bryan Parno, an associate professor in the computer science and electrical and computer engineering departments at Carnegie Mellon University.

At last, the technique is being used to verify the integrity of code in a growing array of real-world applications. The approach could fundamentally change computing. Yet, it is not without formidable obstacles, including formulating algorithms that can validate massive volumes of code at the speed necessary for today's world. The framework also suffers from the same problem every computing model does: if a verified proof is based on the wrong assumptions, it can validate invalid code and produce useless, and even dangerous, results.

"Formal verification is simply a way to up the ante," Fisher explains. "It's a way to modernize and improve the way software is written and ensure that it runs the way it is supposed to operate."

Looking for Proof

Modern software development revolves around a straightforward concept. Coders define a task, write code, and validate it by testing the code in the real world. How a smartphone app, medical device, or autonomous vehicle acts, reacts, and interacts is probed and studied. As programmers discover flaws or ways to improve the code, they rewrite and update the software to improve the performance of the application. Unfortunately, new code often introduces other errors and new vulnerabilities. This cycle sometimes continues *ad infinitum* because there is no way to ensure code is error-free, and it is impossible to imagine every possible scenario, or *corner case*.

That is where formal verification enters the picture.

The idea of writing and executing error-free software isn't new. In 1973, Edsger W. Dijkstra floated the concept of using mathematical proofs to verify code. Three years later, his seminal book *A Discipline of Programming* introduced a conceptual framework for putting verified code to work in the real world. It presented the then-revolutionary idea that programming tasks should be treated more like mathematical challenges. Along the way, other computer scientists, including Tony Hoare, formulated key ideas that demonstrated the feasibility of software verification.

Formal verification advanced slowly, however. Breakthroughs proved difficult because the proofs required are incredibly complex and developing user-friendly, off-the-shelf tools for verifying code is extraordinarily difficult. Adding to the challenge: software validation can be a slow process. Over the last couple of decades, scattered examples of formal verification have appeared, including the use of the technique by the National Aeronautics and Space Administration (NASA) for critical flight software. More recently, organizations such as Amazon, Intel, AMD, IBM, Google, Firefox, and Microsoft have begun to use the technique for assorted software components, ranging from microkernels in processors to Web browsers and cloud infrastructure.

"We have witnessed significant advances in how to think about computer programs mathematically and how to formally specify how a programming language is going to behave," Parno says. Simply put, enormous increases in computing power, significant advances in modeling languages, and the steady evolution of algorithms and off-the-shelf toolkits for verifying code have pushed the field to a tipping point. With a critical mass of knowledge and practical ways to verify code, the method is now moving into the mainstream.

Successfully verifying code hinges on a critical factor: the ability to construct a mathematical model that proves the code is error-free *and* results in the desired outcome.

Successfully verifying code hinges on a critical factor: the ability to construct a mathematical model that proves the code is error-free *and* results in the desired outcome. There are essentially three pieces to this puzzle. First, there is the specification—the mathematical description of what someone wants to achieve, along with their assumptions about the environment. Second, there is the program that actually does the work; this program is written in ordinary code that is essentially the same as any other code without verification. Third, there is the proof that demonstrates the code is actually computing what the specification says it should.

Suppose a software developer wants to write a program to sort an array of A of N numbers into a new array of B. She might write a specification that defines the quality of B being sorted as follows:

For all values j and k such that $0 <= j < k < N$, it must be the case that $B[i] < B[j]$.

In this case, the developer would write a program with normal code to actually sort the values in the array A. She would then write a proof explaining to the verifier why the program correctly sorted the numbers. She might show each step sorts two numbers into their correct places and preserves the correctness of the numbers already sorted. Although the specification in this example is not entirely complete because it's focused only on sorting—there would still be a need to ensure B contains the same elements as A—it certifies that this operation meets desired specifications.

In the real world, "All three of these pieces are fed to an automated verifier, which is responsible for checking whether the proof is correct," Parno says. "Assuming there are no bugs in the verifier and there are no bugs in the specification, then if the verifier signs off, the code is provably/mathematically correct." However, if the code or the proof is incorrect relative to the specification, then the verifier is guaranteed to reject it. "In that case, as a developer, you can go back and fix your code and/or proof until the verifier accepts it," Parno adds.

Getting all three components right is the key. "Suppose you want to write software that only opens a door when someone swipes an authorized badge. If you accidentally write your system specification to say that the door should open when a swipe card fails, then your software might provably meet that specification, but you would be unhappy about the result," Parno explains.

Applying Logic

Temporal logic, which attempts to capture a complete set of actions and behavior over time, is at the heart of formal verification. It is built into the proof used to check the software. Not surprisingly, the complexity and sheer volume of today's code—along with the enormous number of outcomes and possibilities that can occur with computerized tasks—makes it nearly impossible to verify every code component in every library or device. However, the beauty of verified coding is that it is not necessary to ensure every piece of code is free from errors large and small. "There's lots of software where it simply doesn't matter," Fisher points out. "If a component or system doesn't work right within the overall framework or environment, nothing serious happens. The overall structure continues to operate without any serious consequences."

Indeed, the key to effectively using formal verification is to apply it at the right place and in the right way. In some situations, this may mean using the technique for a specific component within a software application. For instance, Google and Firefox have installed code-verified components in their Web browsers. Amazon Web Services (AWS) has turned to this methodology to address design problems in critical cloud systems. This includes using the formal specification language TLA+ (Temporal Logic of Actions) to develop "fault tolerant distributed algorithms for replication, consistency, concurrency control, auto-scaling, load balancing, and other coordination tasks."

In other scenarios, formal verification may translate into building a verified computing system from the processor up. For instance, DARPA has contributed to the development of a binary verified microkernel for the ARM processor it can place in various systems and devices. The kernel, built on C language, required 26 man-years to build. "We know these chips are faithful to the ARM reference design and we can add software on top with a high degree of confidence," says Raymond Richards, program manager in the information innovation office at DARPA. "We want to apply this foundation in places where we know the code isn't going to change quickly and where it's really important to make sure every bit and byte is correct."

DARPA has aggressively pursued formal verification as part of its High-Assurance Cyber Military Systems (HACMS) project and a more recent Cyber Assured Systems Engineering (CASE) initiative. In 2015, the agency generated attention when it held an exercise that encouraged a team of experts to try to hack their way into an unmanned military helicopter. Over the span of the six-week event, the hackers failed to gain access to the software residing in the onboard flight control system. "We removed entire classes of vulnerabilities and created a highly resilient system," Richard says.

Code Compliance

Formal verification continues to advance. A growing array of tools and resources are available to ensure software is mathematically sound. These include Coq Proof Assistant, HOL, Isabelle, Metamath and Z3 Satisfiability Modulo Theories (SMT) solver. The latter, for example, uses several decision procedures to deliver extended static checking, test case generation, and predicate abstraction. The Microsoft-developed tool includes an API library and it works with numerous programming languages, including C, C++, Java, Haskell, OCaml, Python, WebAssembly, and .NET/Mono.

Project Everest, which Parno helped create, aims to build a verified secure implementation of HTTPS. It is supported by Microsoft Research. Yet another high-profile example is seL4, a formally verified operating system microkernel designed to serve as a platform for building safety- and security-critical systems, with no dropoff in performance. Because the kernel controls access to all resources, it dramatically decreases the risk of hacks and attacks. SeL4 has been mathematically proven to be bug-free relative to its specification. Different variations of the kernel are available via a usage model that mimics the Linux Foundation. In fact, SeL4 was used in the DARPA-funded HACMS program to protect the helicopter in its hacking competition.

In the coming years, advances in formal verification likely will lead to far more reliable and secure computers, applications, medical devices, automobiles, robots, IoT systems, and more, Parno explains. The technique will also likely aid in protecting cryptography from far more powerful quantum computers, and it will enable more advanced homomorphic techniques that allow data scientists to analyze and study encrypted data without the ability to view it. While formal verification does not guarantee that a complex software system is completely protected, it greatly improves the odds.

"Formal verification doesn't result in perfect code; it simply narrows the possibility for errors and vulnerabilities to creep in," Parno says. "What makes the technique so attractive is that you push the uncertainty or scope of problems down to smaller and smaller windows."

Further Reading

Barthe, G., Blazy, S, Grégoire, B., Hutin, R., Laporte, V., Pichardie, D., and Trieu, A.
Formal verification of a constant-time preserving C compiler
***Proceedings of the ACM on Programming Languages*, November 2019.**
https://dl.acm.org/doi/abs/10.1145/3371075

Bhargavan K., et al.
Everest: Towards a Verified, Drop-in Replacement of HTTPS SNAPL 2017,
http://www.andrew.cmu.edu/user/bparno/papers/everest-snapl.pdf

Fisher, K., Launchbury, J. and Richards, R.
The HACMS program: using formal methods to eliminate exploitable bugs, 2017,
***Philosophical Transactions of the Royal Society*, 375, 2104.**
https://royalsocietypublishing.org/doi/full/10.1098/rsta.2015.0401

HOL Interactive Theorem Prover
https://hol-theorem-prover.org

Isabelle
https://isabelle.in.tum.de/

Metamath Site Selection
http://us.metamath.org/mm.html

Miller, S.P., Anderson, E.A., Wagner, L.G., Whalen, M.W., and Heimdahl, M.P.E.
Formal Verification of Flight Critical Software
AIAA Guidance, Navigation, and Control Conference and Exhibit, August 2005,
https://shemesh.larc.nasa.gov/fm/papers/FormalVerificationFlightCriticalSoftware.pdf

Project Everest—Verified Secure Implementations of the HTTPS Ecosystem
https://www.microsoft.com/en-us/research/project/project-everest-verified-secure-implementations-https-ecosystem/#!groups

Protzenko, J., Parno, B., Fromherz, A., Hawblitzel, C., Polubelova, M., Bhargavan, K., Beurdouche, B., Choi, J., Delignat-Lavaud, A., Fournet, C., Kulatova, N., Ramananandro, T., Rastogi, A., Swamy, N., Wintersteiger, C.M., and Zanella-Beguelin, S.
EverCrypt: A Fast, Verified, Cross-Platform Cryptographic Provider
http://www.andrew.cmu.edu/user/bparno/papers/evercrypt.pdf

Richards, R.
Cyber Assured Systems Engineering
Defense Advanced Research Projects Agency,
https://www.darpa.mil/program/cyber-assured-systems-engineering

Richards, R.
High-Assurance Cyber Military Systems
Defense Advanced Research Projects Agency,
https://www.darpa.mil/program/high-assurance-cyber-military-systems

The Cog Proof Assistant
https://coq.inria.fr/

The seL4 Foundation
https://sel4.systems/Foundation/About/

Z3 Theorem Prover
https://github.com/Z3Prover

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

Technology | DOI:10.1145/3464937

Esther Shein

A Battery-Free Internet of Things

The Internet of Things can thrive without hardwired or consumable power sources.

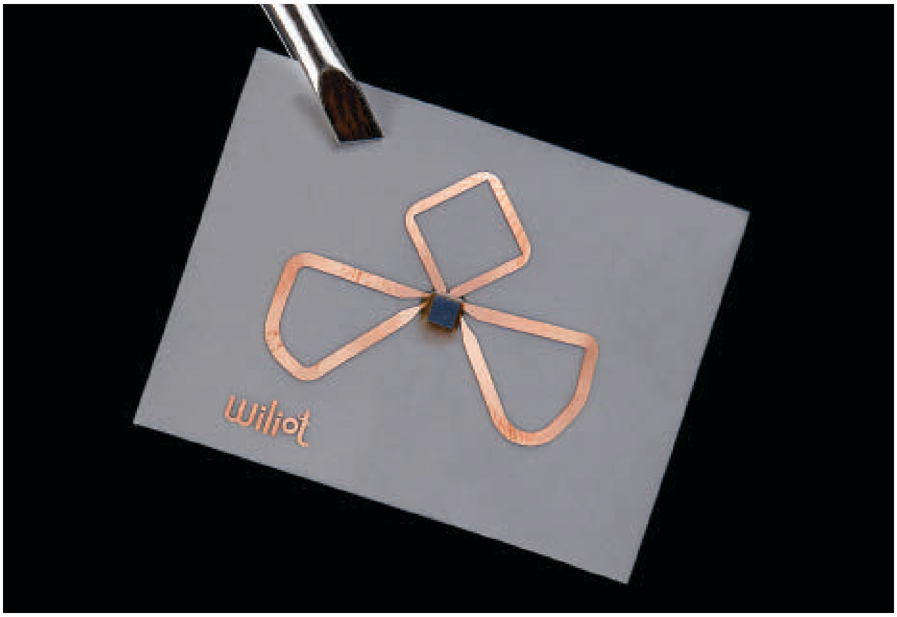
When NVIDIA purchased mobile-chip designer Arm Holdings from SoftBank last year, NVIDIA CEO Jensen Huang made the bold prediction that in the years ahead, there will be trillions of artificial intelligence (AI)-enabled Internet of Things (IoT) devices. Regardless of whether that holds true, it is safe to say the growth of IoT devices is exploding. All those devices will require power sources, and the way Josiah Hester sees it, that's problematic for the environment and society.

"When I see the 'trillion' number, I see a trillion dead batteries, basically,'' says Hester, an assistant professor of computer engineering at Northwestern University. "There's piles of batteries in landfills in China and elsewhere sitting there unrecycled; or they're put in furnaces and melted down, which is not a carbon-neutral event."

As a native Hawaiian, Hester also is concerned about the impact of microplastics and dead batteries turning up in oceans, and about lithium mining, which uses water supplies that people depend on to live. That got him thinking about how to design computer systems without batteries that instead harvest energy, thus reducing their carbon footprint and the impact on the environment.

Hester and other researchers at Northwestern designed a battery-free Nintendo Game Boy that is powered by button presses and sunlight, harvesting energy from the movement of tiny magnets and through tightly wound coils every time a user presses a button.

Now, the team is working on smart face masks that are powered by a person's breathing or movement, that will be able to capture heart or respiration rates, and also to determine whether the person is wearing the mask correctly.



A Wiliot battery-free sensor tag, which contains an energy-harvesting chip that draws power from ambient radio frequencies.

PHOTO COURTESY OF WILIOT.COM/PRESS

"A smart mask can gather these signals and report back to your phone that your mask is slipping and you need to take care of it before you're exposed,'' Hester says, "or if you're wearing it longer than regulatory guidance states the mask will be effective."

The Northwestern researchers have come up with a prototype mask that gathers biological signals and notifies the wearer if it detects problems, "and you never have to plug the stupid thing into the wall,'' Hester adds. "There's a real benefit to this in reducing the burden on the mask wearer. … We're all tired of seeing that 'low battery' indicator" on devices."

Energy harvesting technology allows IoT devices to work without a battery or wired power supply by capturing radio waves and converting them to the small amount of electricity these devices need to operate.

"Energy harvesting technologies are finding valuable opportunities in emerging IoT applications," says George Brocklehurst, a research vice president at market research firm Gartner. "Combining energy harvesting with ultra-low-power semiconductor chips is enabling new IoT use-case possibilities."

In terms of power consumption, low-power microcontrollers (MCUs) can range from requiring milliwatts of electricity down to just tens of microwatts, according to Brocklehurst. "The IoT space is so diverse and use-cases so fragmented, there is a massive sliding scale here for the electronics that support it," he says.

Aside from radio waves, energy can be harvested from other sources, including temperature differentials, vibrations, and the light of the sun, Brocklehurst says.

Others are also researching various use-cases for battery-free IoT devices, with the goal of protecting the environment.

"We want to use computing devices

that can harvest energy from their environment," says Brandon Lucia, an associate professor of electrical and computer engineering at Carnegie Mellon University. Solar panels are the simplest example of this, he says.

Energy harvesting also may be done by attaching an antenna to a device and building a circuit that will store energy from radio waves, he says. However, it takes a long time to store an appreciable amount of energy through the harvesting of radio waves on a device, because radio sources tend to be low-powered, he says. "An appreciable amount here is the amount needed to run [a machine learning] image processing computation on a single image," he explains.

Lucia's lab at CMU recently developed "intermittent computing" platforms that harvest energy and are designed to operate correctly even if there's a power failure or power is not continuously available, Lucia says. The goal is to process sensor data or do machine learning in a battery-less IoT device.

"Intermittent computing is important for correct battery-less operation because if power quits in the middle of an operation, that leads to big problems in typical software,'' he explains. For example, when a device is restarted, the system might be confused. "You forget partially computed things, so it's bookkeeping; how to make sure software is operating correctly even when power is interrupted," Lucia says.

In addition to the intermittent computing model, Lucia's lab has developed a self-powered energy harvesting IoT camera system called Camaroptera, named after a builder bird that collects things from its environment.

The Camaroptera is an inexpensive visual computing device comparable to the size of two standard dice, Lucia says. It has a camera, a long-range radio, and an energy-harvesting computer system attached. It takes pictures, and when an image is pulled off the camera, it can process it with machine learning without having to send it to cloud or the edge to do any offloaded processing, he says.

Camaroptera could be used by a city planner to see things like pedestrian flow and public safety hotspots, Lucia says, adding that since the device does not rely on batteries, it can last a long time. While solar panels and chips eventually will wear out, "the operating lifetime of even a rechargeable battery is much shorter than the lifetime of those components,'' he says. "A battery is no longer useful after a few years, but other hardware components can last decades."

"Intermittent computing is important for correct battery-less operation because if the power quits in the middle of an operation, that leads to big problems in typical software."

The device collects energy using a small array of solar panels that sit on top of it. "We found solar panels that provide enough power and can capture an image and process it,'' with reasonable frequency—every 20 seconds, he says.

Shyam Gollakota, an associate professor at the University of Washington, has been dabbling with energy harvesting for several years, starting in 2013 with ambient backscatter technology, which uses existing signals instead of generating its own. In effect, "It enables wireless communications out of thin air," he says.

In an IoT context, data is transmitted via wireless communication protocols; the drawback is that radio signals typically consume a lot of power, Gollakota says. Backscatter reflects ambient Wi-Fi signals from the environment.

Gollakota and others at the university are working on a battery-free phone. One company commercializing the technology is Jeeva Wireless, a Seattle-based startup that uses passive backscatter technology that reflects energy from existing radio frequency signals for use by IoT devices.

The University of Washington researchers have shown that backscatter signals can be used to enable long-range communications using radio frequency (RF) signal reflections on battery-free devices, he says.

"People have thought of backscatter as for passive RFID and identification,'' Gollakota says. "We have shown that we can develop technologies that use backscatter from reflecting signals and also get much longer ranges of hundreds of meters, so they can be used for IoT devices and wireless sensors."

The researchers also have shown that continuous video streaming can be done using backscatter, he says.

Another startup, Keisarya, Israel-based Wiliot, aims to scale IoT with a battery-free Bluetooth tags that are powered by harvesting RF energy.

Energy Harvesting Drawbacks

Energy harvesting is not without its obstacles. Lucia and Hester both say the main challenge with energy harvesting systems is there isn't always enough energy.

"Our appetite for things we want to have done requires a lot of energy, and it's hard to harvest the amount of energy to do something sophisticated like recognize a face in an image,'' Hester says. "That would take a good amount of energy, and that's difficult with these battery-free small devices."

Right now, Hester's team is working on figuring out how to make the energy harvester on a mask small enough that it is not burdensome, but also harvests enough energy to capture bio data. "So trying to balance that right now is the main challenge, not just with masks, but building these battery-less devices,'' Hester says.

This is why intermittent computing techniques are important, adds Lucia. "We provide the illusion … of continuous operations," while making sure things run as efficiently as possible and that the software operates properly.

A key research challenge in intermittent computing is that correctness sometimes comes with some time or energy overhead, because software needs to execute carefully to remain correct even when power fails, Lucia says. "That's the cost you pay for energy harvesting, which comes in a lot of different forms—you can design larger systems with larger energy collectors but they take up more room."

He adds, "We're shooting for devices that disappear in the environment and are small. Our work is targeting the low end of energy harvesting devices: cheap, small, yet capable devices."

While energy harvesting is not an inexpensive proposition, Gartner's Brocklehurst says when viewed from a total cost of ownership standpoint, there is savings in operational expenditures beyond the initial capital expenditure/investment.

The FCC Factor

Lucia says their devices use LoRa, which operates in an unlicensed band of spectrum, an ISM band reserved for industrial, scientific, and medical purposes so there are no FCC concerns. LoRa has quite a long range, Lucia says. "Using off-the-shelf chips in a standard configuration, we expect 1km–10km line-of-sight transmission to be possible," or perhaps even longer. "At such long distances, establishing the line of sight is the challenge, of course," he says.

Hester says eventually, there will be regulatory concerns when battery-free devices are more widespread, and they will face GDPR-type questions over who holds the data. "That's an incredibly important question that is really hard, and I don't know if the field has fully started to explore this yet,'' he says.

The FCC has not regulated backscatter because it has been limited to radio-frequency identification (RFID) applications, and those reflections are much weaker than radio signals that create interference, according to Gollakota.

Meanwhile, the future of IoT is in battery-less devices and looks exciting, Lucia says. Yet he acknowledges more research needs to be done before that lofty goal is reached. "We need systems that sense, communicate, and compute more readily. Research is being done to push systems forward that are more efficient and optimized for energy,'' which is the driving force, he says.

Hester agrees. "We have seen a steady decrease in energy consumption and an increase in the amount of power we can harvest per square inch of an energy harvester for IoT devices,'' he says. "In the past seven years, we have gone from temperature monitoring to gaming devices with energy-harvesting IoT, which is a big jump."

This is due in part to steady technology advances, he says, but there are other reasons as well.

"The research community has been developing new ways to speed up sensing, computing, and machine learning, specifically for these devices, which has paid off in a big way,'' Hester says. "Fundamentally, we are discovering a bunch of ways to do more with less energy. Together with advancements in low-power backscatter communication, we have an exciting future for energy-harvesting IoT."

Gartner's Brocklehurst is not so sure. Energy harvesting still requires some sort of storage, and "more than likely, it's a battery, still,'' he says. "But with an energy harvester, you can extend battery life."

That is still a meaningful return on investment, Brocklehurst says. 🄲

Eventually, there will be regulatory concerns when battery-free devices are more widespread, and they will face GDPR-type questions over who holds the data.

Further Reading

De Winkel, J., Kortbeek, V., and Hester, J.
Battery Free Game Boy, September 2020, ***Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies***, **https://bit.ly/3uZeZxJ**

De Winkel, J., Delle Done, C., Yildirim, K.S., Pawelczak, P., and Hester, J.
Reliable Timekeeping for Intermittent Computing, March 2020, ***Proceedings of the 25th International Conference on Architectural Support for Programming Languages and Operating Systems***, **https://bit.ly/3blhSBn**

Lucia, B. and Ransford, B.
A Simpler, Safer Programming and Execution Model for Intermittent Systems, ***Proceedings of the 36th Annual ACM SIGPLAN Conference on Programming Language Design and Implementation***, **https://bit.ly/3eebX2H**

Lucia, B., Balaji, V., Colin, A., Maeng, K., and Ruppel, E.
Intermittent Computing: Challenges and Opportunities, ***Proceedings of the Summit on Advances in Programming Languages*** **(*SNAPL*) 2017, https://bit.ly/3c36bOQ**

Gobieski, G., Lucia, B., and Beckmann, N.
Intelligence Beyond the Edge: Inference on Intermittent Embedded Systems, ***Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems*** **(*ASPLOS'19*), https://bit.ly/30f0mbs**

Colin, A., Ruppel, E., and Lucia, B.
A Reconfigurable Energy Storage Architecture for Energy-Harvesting Devices, ***Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems*** **(*ASPLOS 18*), https://bit.ly/3qmfZIK**

Nardello, M., Desai, H., Brunelli, D., and Lucia, B.
Camaroptera: A Batteryless Long-Range Remote Visual Sensing System, ***Proceedings of the 7th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*** **(*ENSsys 19*), https://bit.ly/3qmfZIK**

Talla, V., Hessar, M., Kellogg, B., Najafi, A., Smith, J.R., and Gollakota, S.
LoRa Backscatter: Enabling The Vision of Ubiquitous Connectivity, ***Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies***, **September 2017, https://dl.acm.org/doi/10.1145/3130970**

Naderiparizi, S., Hessar, M., Talla, V., Gollakota, S., and Smith, J.R.
Towards Battery-Free HD Video Streaming, ***Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation*** **(*NSDI 18*), April 2018, http://bit.ly/2O4GaH0**

Videos

Game Boy: https://www.youtube.com/watch?v=ErapUArQahM

Ambient backscatter: https://www.youtube.com/watch?v=gX9cbxLSOkE&feature=emb_title

Passive Wi-Fi: https://www.youtube.com/watch?v=AZ-tISX-7Cw

Battery-free phone: https://www.youtube.com/watch?v=5f5JJTmbO4U&feature=emb_title

Esther Shein is a longtime freelance technology and business writer based in the greater Boston area.

Society | DOI:10.1145/3464935

Paul Marks

The Future of Supply Chains

Droids, drones, and driverless technologies are fueling a supply chain revolution.

TODAY’S SUPPLY CHAINS are labor-intensive and expensive to run. A number of autonomous systems that reduce the human factor are about to change all that.

What do the sidewalks around us, the airspace above us, interstate freeways, and deep ocean shipping lanes have in common? The answer is that they are all places where developers of autonomous technology are trying to revolutionize the economics of supply chains. The plan is to use robotic technology to deliver anything from packages to take-out food, groceries, or bulk freight in ways that can reduce the logistics industry’s dependence on that most expensive of supply chain costs: human labor. If the use of electric drivetrains can cut carbon emissions too, so much the better.

Currently, van-based delivery over the last mile to homes may comprise as much as 40% of the overall transportation cost, even if a package delivered in Europe or the U.S., say, originated in Asia. This final stretch, also known as the “last mile,” has long been an early target for roboticists. A number of last-mile robots have moved from early trials into full-scale operations, as wheeled delivery robots now ply city sidewalks, delivering everything from groceries to pizzas; drones deliver food from the clubhouse to golfers on the green, and medical delivery drone services launched before and during the COVID-19 pandemic ferry drugs, tissue samples, coronavirus tests, and medical supplies to end users.

On the roads, makers of emerging breeds of driverless trucks are preparing to begin tests of autonomous, long-haul semi-tractor-trailers that they hope to run on high-volume freight routes between cities, shifting goods from city depot to city depot. At sea, the shipping industry—one of the world’s worst polluters as a result of its use of a filthy, sulphur-rich diesel called bunker fuel—is developing cleaner, electrically-driven, autonomous ships with artificial intelligence (AI) at the helm.

PHOTO BY CHARLIE LEIGHT/ASU NOW

Ground Robots Hit the Last Mile

So just who is fielding, or trialing, these technologies? One well-known last-mile robot maker is an energetic start-up called Starship Technologies, based in San Francisco. Operating in the U.S., the U.K., Germany, Denmark, and Estonia, the company’s robots “are now doing thousands of commercial deliveries a day and millions of autonomous miles per year,” says Vice President Henry Harris-Burland.

Starship’s machines are lithium battery-powered, white plastic robots that travel at 5mph (8kph) on six stubby wheels. Peppered with cameras, radars, and ultrasound sensors for collision avoidance, they use GPS, computer vision, and proprietary 2.5-cm-resolution maps to navigate. They use AI to continually improve their guidance system as they travel—by retraining a neural network when, for instance, an unmapped obstacle is regularly met on a route. “Our goal has always been to achieve 99% autonomy, and our hardware and software allow us to travel safely autonomously,” Harris-Burland says. The goal is not for 100% autonomy, he explained, because a remote human operator can assume radio (4G/5G) control if a robot gets stuck.

The COVID-19 pandemic has made contactless robotic delivery a more sought-after service. “It proved to be one of the most reliable ways to protect vulnerable populations and enable social distancing,” Harris-Burland says. In Milton Keynes, U.K., which has boulevards broad enough for many robots and pedestrians to share, Starship robot numbers tripled during 2020, with 100 droids now plying the streets—the largest robot delivery fleet anywhere in the world, the firm says.

Starship, however, is far from alone in this space. Lux Research, a market analyst in Boston, MA, says other key last-mile robotics players to watch are FedEx, Amazon, and Kiwi Campus in the U.S., plus Zhen Robotics in China. Japanese technology heavyweight Panasonic has just joined the fray, too.

Lux advises industry watchers to be on the lookout for interesting robot combinations: ground robots could be disgorged from driverless vans, or perhaps carry their own parcel delivery drones.

Drones: In a Holding Pattern

Delivery drones are not ready for prime-time everywhere just yet. While they have long been put forward as ideal last-mile platforms, the practice of drone delivery to homes has yet to match the promise, at least outside early deployments in Asia. Instead of bringing packages to homes in regions like the U.S. and Europe, drones have instead been used for a handful of limited applications such as medical delivery applications.

In February, for example, Britain’s National Health Service pressed a

40-mile-range drone into service to ferry personal protective equipment (PPE), COVID-19 testing kits, tissue samples, and drugs between medical facilities in the remote Scottish islands. In May 2020, United Parcel Service (UPS) began a drone-based service delivering medical prescriptions to Florida retirement homes.

In Japan, the e-commerce/online retailing firm Rakuten Inc. has launched drone-based delivery services for food, while in China, online retailers Alibaba and JD.com are now doing likewise. In Europe and the U.S., would-be drone delivery players like Amazon are finding the regulatory environment is tougher—with a thicket of safety, reliability, privacy, insurance, and security issues to negotiate before they can launch their long-promised services.

Among those regulatory issues are the need to safely navigate the built environment in three dimensions, avoiding the likes of people, birds, vehicles, buildings, power lines, street lamps, and trees, as well as coping with adverse weather. Then there's the thorny question of redundancy: how does an aerial drone with a failed battery, motor, or rotor recover, rather than crashing on a crowded street?

On top of all that, industry observers are getting nervous about the privacy implications of having a flying machine brimming with cameras hovering slowly into someone's yard, as the potential for privacy breaches is profound. As a result, engineers at the Indian Institute of Science in Bangalore, for instance, are working on ways to ensure delivery drones can have privacy safeguards baked into their control software, but it is an open source work-in-progress project that is nowhere near ready for deployment.

"A number of regulatory issues must be addressed before drone delivery services are cost-effective at scale," says David Kovar, CEO of Unmanned Robotics Systems Analysis (URSA Inc), a risk management firm in Manchester, NH. "Countries with more authoritarian governments are often able to drive through this process fairly quickly, but the U.S. is held back by a very slow bureaucratic process that is deeply focused on minimizing risk to manned aviation. It is very difficult to speed the process along, and we lack the data to truly understand the potential, and actual, risks."

Until regulators stop butting heads, drone delivery rollout likely will continue to lag well behind ground robots in the autonomous delivery stakes.

Jonathan Rupprecht, an attorney in Palm Beach County, FL, specializing in drone litigation, says changing a single aspect of drone regulation tends to have a domino effect on others. "One modification causes all sorts of consequences in another area of the law, or in the cost of drone operations. When you try and scale this out to multiple operational environments that have unique ground- and air-risk profiles, it causes problems somewhere else."

So the likes of the U.S. Federal Communications Commission, the Department of Transportation, and the Federal Aviation Administration end up at loggerheads. Until regulators stop butting heads, drone delivery rollout will likely continue to lag well behind ground robots in the autonomous delivery stakes. One market analyst, Urban Mobility Labs of Los Angeles, is predicting drone delivery services will not kick off proper in the U.S. until 2023.

Driverless Trucks: Station to Station

UPS's drone delivery service for prescriptions is just one of the giant freight shipping firm's autonomous delivery projects. UPS is also involved in another, more-ambitious plan: autonomous city-to-city driverless freight trucking, care of a partnership with self-driving truck systems specialist TuSimple in San Diego, CA, and its partner, truck maker Navistar of Lisle, IL.

Unlike other firms planning driverless trucks, like Tesla and Google-owned Waymo, TuSimple is not embedded in self-driving car heritage. Instead, TuSimple's aim is to develop autonomous technology that is entirely customized for the unique environment in which trucks operate, rather than shoehorning robotic technology from driverless cars into trucks.

Unlike self-driving cars, which have a LiDAR and radar range of about 200 meters (more than 650 feet), TuSimple says a wider, longer, articulated 40-ton driverless semi-truck needs to look much further ahead when traveling at highway speeds. That is why it has engineered a system in which an array of high-definition cameras give its software knowledge of what's happening on the road up to a full kilometer (more than a half-mile) ahead, which, it estimates, is twice as far as a human truck driver can anticipate upcoming road activity.

One upshot of this extreme-look-ahead technology is that the trucks need to brake far less often, saving fuel. In trials TuSimple's 40-truck fleet is running in New Mexico, Arizona, and Texas—with safety drivers who can step in if needed—the company has found, in research verified by the University of California, San Diego, that the autonomous technology is burning up to 10% less fuel per trip than similar human-driven trucks.

Autonomous Ships: All At Sea

While Tu-Simple, UPS. and Navistar hope their fully autonomous trucks will be fit for unrestricted driverless operation by 2024, one firm is already set to out-innovate them and take away a chunk of their market: Kongsberg Maritime of Norway, which is in the vanguard of industry moves to make cargo ships, tankers, and ferries operate more autonomously, with much lower crew numbers than today.

At issue, says An-Magritt Tinlund Ryste, product director for next-generation shipping at Kongsberg Maritime, is that logistics firms are being pressed to improve their carbon footprints. Also, if the beginning and end of a truck's journey is near the ocean, why not avoid trucks altogether and use an environmentally friendly, electrically driven ship that also is equipped with autonomy to keep crew costs low, to ply the sea route?

To prove such technology, the firm

is building an 80-meter-long largely autonomous container ship for Norwegian fertilizer maker Yara International. Called the Yara Birkeland, the vessel will service ports that usually require 40,000 truck trips every year. "It's not completely unmanned yet; it's designed for reduced crew. And our approach to how this is done has been agreed with by the Norwegian maritime authorities," says Ryste.

What encouraged construction of the Yara Birkeland, says Ryste, is the increasing level of autonomy being requested by operators of ferries and freight barges, in terms of navigation and docking which, if a vessel's captain agrees, can be switched to autonomous mode.

Kongsberg Maritime is far from alone in seeking greater autonomy on the high seas. In the spring of this year, IBM and Marine AI Ltd. of Plymouth, U.K., experimented with a transatlantic drone called the Mayflower Autonomous Ship (MAS) in a bid to create a generalizable AI for ship control. Aiming to mimic the journey of the original Mayflower from Plymouth, England, to what became the Plymouth colony in the New World of 1620, the navigational intelligence at the heart of MAS is called the AI Captain.

"The AI Captain uses a true hybrid AI system, making use of deep learning, prescriptive logic/inference rules, and also optimization/linear programming. And deep learning-based object detection, classification, and tracking is at the heart of the computer vision system used for confirming and identifying navigation hazards," says Don Scott, Marine AI's chief technology officer.

Are there emergent properties that no one could have predicted that could constitute a risk to vessel safety? "It's conceivable that we may encounter a novel set of circumstances that is out of the comfort zone of what we have trained the AI Captain for in the simulator, such that it temporarily isn't certain what to do next," says Andy Stanford-Clark, chief technology officer of IBM UK & Ireland.

"In this case, the 'backstop' is to stop, wait, look around, and decide calmly what to do next. That's one of the benefits of autonomy at sea: things tend to happen a lot slower."

At its heart, says Scott, the rule the AI Captain harks back to in adversity mirrors those of all the autonomous robots, drones, trucks, and ships headed for the supply chain.

"The prime directive is: Don't hit anything."

Further Reading

Constant, S.
NHS launches UK's first COVID test drone delivery service in Scotland, *Skyports*, Feb. 23, 2021
https://skyports.net/2021/02/nhs-launches-uks-first-covid-test-drone-delivery-service-in-scotland/

Edwards, D.
UPS to use drones to deliver medicines to retirement homes in Florida, *Robotics & Automation News*, May 26, 2020.
https://roboticsandautomationnews.com/2020/05/26/ups-to-use-drones-to-deliver-medicines-to-retirement-homes-in-florida/32453/

Gan, M., Qian, Q., Li, D., Ai, Y., and Liu, X.
Capturing the swarm intelligence in truckers:
A foundational analysis for future swarm robotics in road freight.
Swarm and Evolutionary Computation, April 2021.
https://doi.org/10.1016/j.swevo.2021.100845

Greengard, S.
When Drones Fly, *Communications*, November 2019, Vol. 62 no. 11
https://cacm.acm.org/magazines/2019/11/240357-when-drones-fly/fulltext

Kern, J., Robinson, C.
Automating *the* Last Mile – A Market Research Report, Lux Research, March 12, 2020,
https://www.luxresearchinc.com/automating-the-last-mile-executive-summary

Marks, P.
Eyes On The Skies: Delivery Drone Privacy Issues Arise, *Communications*, Feb. 2, 2021.
https://cacm.acm.org/news/250268-eyes-on-the-skies/fulltext

Rupprecht, J.
The Truth About Drone Delivery No One Is Talking About, Feb. 4, 2021
https://jrupprechtlaw.com/drone-delivery/

The Mayflower Autonomous Ship
https://mas400.com/

The Autonomous Ship Project: Key Facts About Yara Birkeland
https://www.kongsberg.com/maritime/support/themes/autonomous-ship-project-key-facts-about-yara-birkeland/

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

ACM Member News

DEVELOPING SOCIALLY ASSISTIVE ROBOTICS

"When I was 16, my uncle told me, 'computers are the future,'" recalls Maja Matarić, Chan Soon-Shiong Chair and Distinguished Professor of Computer Science, Neuroscience, and Pediatrics at the University of Southern California.

"He was right. I am glad I listened," she adds.

Matarić went on to earn an undergraduate degree in computer science from the University of Kansas, and both her master's and doctorate degrees in computer science and artificial intelligence (AI) from the Massachusetts Institute of Technology in Cambridge, MA.

Her research is aimed at endowing machines with the ability to help people, especially users who have special needs. Her lab's focus is on developing technologies that augment human ability, rather than merely automating/replacing the work of people.

"I am developing robots that support the human ability to cope, helping people to help themselves," Matarić says.

Lately, Matarić has been focused on anxiety, isolation, and depression, conditions on which the pandemic has shone a spotlight. While she would like to see more socially assistive robotics, such solutions are rare and not in the mainstream. Matarić's intention is to help bridge this gap and make this technology available on a larger scale.

Matarić is passionate about making a difference. Her message to her students is to work on meaningful problems, especially in computing. "You are working on the tools that are the fabric of the 21st century," she tells them.

"We have so many societal problems that can be positively impacted through computing. Choose to work on meaningful things."

—John Delaney

In Memoriam | DOI:10.1145/3467481

Simson Garfinkel and Eugene H. Spafford

Charles M. Geschke (1939–2021)

CHARLES (“CHUCK”) M. GESCHKE helped create the modern world of computing, where beautiful typography and expressive, artistic graphics are as integral to most users’ experience as numbers and text.

Geschke earned an A.B. in classics and a master’s in mathematics, both at Xavier University. He then enrolled in the computer science program at Carnegie Mellon University, where he earned his doctorate in 1972 under the supervision of William Wulf.

After graduating, Geschke went to work at the Xerox Palo Alto Research Center (PARC), the storied lab that invented the dominant desktop computing paradigm of the past three decades: high-performance single-user desktop computers with bitmap displays that display information in “windows,” connected by a local area network, printing documents on a laser printer. Geschke was tasked with developing an imaging research group at PARC.

In 1978, Geschke hired John Warnock, who had previously worked at Evans & Sutherland, a company that developed computer graphics displays. The two decided that many of the implementation details for controlling the laser printer should be hidden from programmers by using a high-level “page description language” that specified fonts, typography, and graphics. They named their creation Interpress.

Xerox declined to commercialize Interpress, so Geschke and Warnock left in 1982 to found Adobe, where they created an improved language they named PostScript. Apple Computer invested in the company and became their first customer. Apple put PostScript at the center of its Apple LaserWriter, which created the phenomena of desktop publishing soon after its launch in 1985.

But Adobe was no one-hit-wonder; PostScript was the first of a series of industry-standard products, including PhotoShop, Illustrator, and Acrobat.

PHOTO COURTESY OF THE ENGINEERING AND TECHNOLOGY HISTORY WIKI/ETHW.ORG

“Chuck and I built the company together and both were on the Adobe Board until he retired in 2019,” recalls Warnock. “He was highly respected by all employees and was instrumental in building a strong high-growth company.”

Geschke was Adobe’s chief operating officer from 1986 until 1994, president from 1989 until his retirement in April 2000, and co-chair of Adobe’s board (with Warnock) from 1997 until 2017. He remained on the board until April 2020.

For the invention of PostScript, Geschke shared the 1989 ACM Software System Award with Douglas K. Brotz, William H. Paxton, Edward A. Taft, and Warnock. He was elevated to ACM Fellow in 1999. Geschke was also a Fellow of the Computer History Museum, a Fellow of the American Academy of Arts and Sciences, and was elected to the National Academy of Engineering. Together with Warnock, Geschke was awarded the 2006 David Packard Medal of Achievement by the California Technology Council, the 2008 National Medal of Technology and Innovation by President Barack Obama, the IEEE Computer Society’s 2008 Computer Entrepreneur Award, and the 2010 Marconi Prize.

> **“He was highly respected by all employees and was instrumental in building a strong, high-growth company.”**
>
> —JOHN WARNOCK, WHO COFOUNDED ADOBE, INC. WITH CHUCK GESCHKE IN 1982.

“Chuck and I never had an argument over the 43 years we had known each other,” Warnock said. “I will always miss him.”

Simson Garfinkel is a part-time faculty member at George Washington University in Washington, D.C., USA. He is an ACM Fellow.

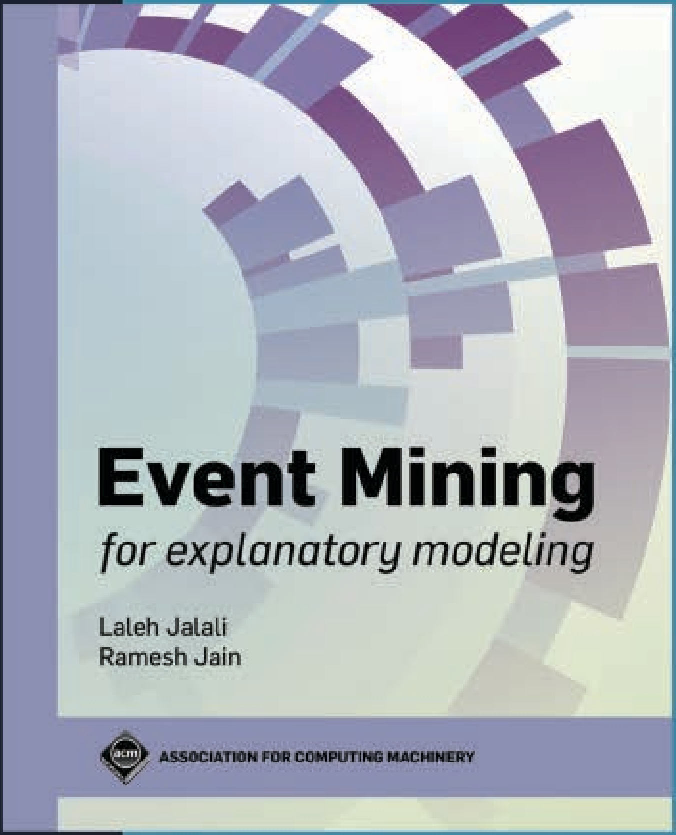
Eugene H. Spafford is a professor of computer science and the founder and executive director emeritus of the Center for Education and Research in Information Assurance and Security at Purdue University, W. Lafayette, IN, USA. He is an ACM Fellow.

ACM BOOKS

Collection II

This book introduces the concept of Event Mining for building explanatory models from analyses of correlated data. Such a model may be used as the basis for predictions and corrective actions. The idea is to create, via an iterative process, a model that explains causal relationships in the form of structural and temporal patterns in the data. The first phase is the data-driven process of hypothesis formation, requiring the analysis of large amounts of data to find strong candidate hypotheses. The second phase is hypothesis testing, wherein a domain expert's knowledge and judgment is used to test and modify the candidate hypotheses.

The book is intended as a primer on Event Mining for data-enthusiasts and information professionals interested in employing these event-based data analysis techniques in diverse applications. The reader is introduced to frameworks for temporal knowledge representation and reasoning, as well as temporal data mining and pattern discovery. Also discussed are the design principles of event mining systems. The approach is reified by the presentation of an event mining system called EventMiner, a computational framework for building explanatory models. The book contains case studies of using EventMiner in asthma risk management and an architecture for the objective self. The text can be used by researchers interested in harnessing the value of heterogeneous big data for designing explanatory event-based models in diverse application areas such as healthcare, biological data analytics, predictive maintenance of systems, computer networks, and business intelligence.



Event Mining

for explanatory modeling

Laleh Jalali
Ramesh Jain

ASSOCIATION FOR COMPUTING MACHINERY

Event Mining

for explanatory modeling

Laleh Jalali
Ramesh Jain

ISBN: 978-1-4503-8483-4
DOI: 10.1145/3462257

http://books.acm.org
http://store.morganclaypool.com/acm

DOI:10.1145/3466607

Pamela Samuelson

Legally Speaking
Reimplementing Software Interfaces Is Fair Use

A multifactored rationale for denying Oracle's claim against Google.

A long-standing, generally accepted norm in the computing field distinguishes between software interfaces and implementations: Programmers should have to write their own implementing code, but they should be free to reimplement other developers' program interfaces. This norm, of which Sun Microsystems, the developer of Java, was once the software industry's foremost proponent, is now the law of the land in the U.S. after the Supreme Court's decision in *Google Inc. v. Oracle America, Inc.*, which overturned a lower court ruling that reimplementing an interface infringed copyright.

The Supreme Court took Google's appeal on two issues. One was whether program interfaces are protectable by copyright law. The Supreme Court declined to decide that issue, even though many amicus curiae (friend of the court) briefs filed by software developers, organizations such as the Electronic Frontier Foundation, the Center for Democracy & Technology, and the Computer & Communications Industry Association, as well as numerous intellectual property scholars, supported Google's argument that program interfaces are uncopyrightable.

A second issue was whether Google's reimplementation of 11,500 declarations from 37 Java Application Program Interface (API) packages in its Android smartphone platform was fair use or infringement. Although a jury rendered a verdict in favor of Google's fair use defense after a two-week trial, the Court of Appeals for the Federal Circuit (CAFC) overturned this verdict. The CAFC concluded that no reasonable jury could have found Google's appropriation of that many lines of computer code was fair use. This is the ruling that Supreme Court's decision reversed.

The penultimate sentence in Justice Breyer's opinion for the 6-2 majority succinctly states the Court's conclusion: "where Google reimplemented a user interface, taking only what was needed to allow users [that is, programmers] to put their accrued talents to work in a new and transformative program, Google's copying of the Sun Java API was a fair use of that material as a matter of law."

After explaining Oracle's claims against Google, this column reviews the Court's reasons for rejecting Oracle's arguments on the fair use issue.

Oracle's Claims

Oracle's main argument on the copyrightability issue was that Sun's engineers had exercised considerable creativity in articulating, naming, and organizing the Java API declarations in an overall taxonomy of Java API packages, classes, and methods. As creative elements of the Java Standard Edition, Oracle contended they were eligible for copyright protection. It characterized the declarations as "declaring code," saying these lines of code were equally as copyrightable as implementing code.

Oracle's arguments against Google's fair use defense were more multifaceted. Google's purpose in copying the Java API declarations, in its view, was commercial and non-transformative (that is, Google used the declarations for exactly the same purpose as if Oracle or Sun had licensed this use). Both considerations, Oracle asserted, tipped against fair use.

IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

Because Google went ahead and used the Java API in Android after its negotiations with Sun for a license to use Java technologies fell through, Google had acted in bad faith. Furthermore, Google had taken unfair advantage of the popularity of those declarations with experienced Java programmers. Applications that Java programmers have developed for the Android platform violate the "write once, run everywhere" objective of Java.

Also tipping against fair use, said Oracle, was Google's exact copying of 11,500 lines of highly creative declaring code, which was quantitatively and qualitatively substantial. This taking had caused Oracle to suffer market harm because Oracle should have been getting substantial license fees from Google for its use of the Java API.

The phenomenal success of the Android platform had made it impossible for Oracle to enter or to license Java for smartphone platforms. Oracle claimed it was entitled to $9 billion in damages due to Google's misappropriation of the declarations.

The CAFC found Oracle's arguments persuasive on virtually every point.

The Jury Verdict Mattered

One reason the Supreme Court overturned the CAFC's fair use ruling was because the CAFC substituted its judgment for the jury's about Google's fair use defense. Appellate courts are supposed to defer to jury verdicts unless they are demonstrably wrong. The Supreme Court decided that the CAFC's anti-fair use judgment was demonstrably wrong, not the jury's.

By taking an extremely narrow view of the jury's role in fair use cases, the CAFC had, in effect, usurped the legitimate role of the jury. Justice Breyer's opinion took that court to task for ignoring evidence that supported many of Google's arguments, including evidence showing that Google's use of the Java API had not caused Oracle to suffer cognizable market harm.

Differences Between Declaring and Implementing Code

In most fair use cases, the nature of the copyrighted work factor is quite insignificant. In the *Google* case, the Court elevated its significance by starting its fair use analysis with a discussion of this factor.

The Court characterized the Java API declarations as a "user interface" because it was a way for programmers to control performance of specific tasks by computers through a set of commands. The Court characterized Java API declaring code as having different kinds of capabilities and as "embody[ing] a different type of creativity" from implementing code.

Declaring code was, moreover, "inextricably bound up with the use of specific commands known to programmers as method calls" that Oracle does not claim as copyrightable. Yet declarations are also "inextricably bound with implementing code, which is copyrightable, but which was not copied." The Court viewed implementing code as much more conventionally expressive in a copyright sense than declaring code.

Also significant was that the value of declaring code derived chiefly from the investments that Java programmers had made in learning them so they could create new programs.

The Court also noted that Sun's business strategy had been to make APIs open and to compete on implementations. Oracle may have adopted a different strategy, but that did not change Sun's history of promoting open interfaces.

Google's Purpose

The Court viewed Google's purpose in using the Java declarations in a positive light. It characterized the Android platform as "a highly creative and innovative tool for a smartphone environment." This platform also enabled experienced Java programmers to take advantage of their investment in learning the Java declarations to invoke common tasks when developing new programs. The Court regarded this as the very kind of creativity that copyright law was designed to promote.

The jury had, moreover, heard evidence that reimplementation of APIs was common and accepted in the software industry as long as developers reimplemented the interfaces in independently written code. Sun executives had believed the widespread use of Java would benefit it.

Substantiality of the Taking?

Although 11,500 lines of code may seem like a lot, the Court pointed out the Java API consists of 2.86 million lines of code. Google had used only 0.4% of the total amount. Those lines of code were, moreover, a small part of the 15 million lines of Android code.

Moreover, Google had produced evidence it copied only the declarations that Java programmers needed to write code for the Android smartphone platform. This new platform was different in kind than the laptop and desktop platforms for which the Java API had initially been developed.

Also relevant was that Google had copied the 11,500 declarations "not because of their creativity, their beauty, or even (in a sense) because of their purpose." Rather, it did so "because programmers had already learned to work with the Sun Java API system." It would be difficult "to attract programmers to build its Android smartphone system without them." Use of the Java declaring code "was the key [Google] needed to unlock the programmers' creative energies."

Market Harm?

Oracle's market harm arguments were unavailing, in part because the "jury could have found that Android did not harm the actual or potential markets" for the Java interfaces.

Although 11,500 lines of code may seem like a lot, the Court pointed out the Java API consists of 2.86 million lines of code.

The failed negotiations with Sun had been over much more than the use of the 37 packages Google used in Android. Hence, Oracle's lost license fee argument failed.

The Court pointed to trial testimony that Sun's efforts to enter the mobile phone market with Java had been failures. Google's economic expert had explained that Android was "part of a distinct (and more advanced) market than Java software." Because the jury found Google's use of the Java declaration to be fair use, it must have decided against Oracle's market harm claims.

Significantly, the Court concluded that to enforce Oracle's claim of copyright infringement against Google would "risk harm to the public." In particular, it would make reimplementation of APIs "a lock that limited the future creativity of new programs," thereby "interfer[ing] with, not further[ing], copyright's basic creative objectives." Hence, the Court concluded Google's use of the Java declarations was fair use as a matter of law.

What About Copyrightability?

The controversy over whether program APIs can be copyrighted dates back to the 1980s. When the issue was finally litigated in the late 1980s and early 1990s, several appellate courts ruled that reimplementation of interfaces, when necessary to achieving compatibility with other programs, was not copyright infringement. The CAFC's ruling on the copyrightability of APIs in *Oracle v. Google* was contrary to more than 20 years of other court precedents on this issue.

Google, along with dozens of amicus curiae briefs filed in support of Google's legal positions, would have preferred for the Court to have decided the *Google* case on copyrightability grounds. In an amicus curiae brief, 71 intellectual property scholar colleagues and I argued that APIs should be deemed unprotectable methods or procedures as a matter of copyright law, as pro-interoperability lower appellate court decisions had done.

A common reason why software industry briefs urged the Court to rule on the copyrightability issue was that this ruling would provide greater certainty in the software industry so that developers could reimplement others' interfaces free from fear of lawsuits.

Fair use, by contrast, is typically very fact-intensive and decided on a case-by-case basis. Developers do not want to have to spend precious resources fighting litigation over their reimplementations of APIs.

Conclusion

The Court's *Google v. Oracle* analysis of the fair use issue is strongly favorable to fair use defenses involving reimplementations of APIs and hints in several places that APIs may not be copyrightable.

The Court's decision leaves untouched several appellate court rulings that denied copyright protection to program interfaces, even citing to some of them approvingly. Those decisions all involved situations in which true program-to-program interoperability was at stake.

Google v. Oracle was not as convincing as a compatibility case because apps developed for the Android platform do not necessarily run on other Java platforms. And apps written for Java-compliant laptops do not necessarily work on Android. Fair use was, consequently, a better defense than challenging their copyrightability.

In view of the Court's decision and the intact precedents in true interoperability cases, I predict there will be very few software interface infringement cases brought any time soon. Software developers can now breathe a deep sigh of relief as Oracle's aggressive claims have been debunked. C

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.

DOI:10.1145/3466609

Lorrie Faith Cranor

Privacy
Lessons From the Loo

Illustrating privacy concepts with potty talk.

IN 2014, I began the Privacy Illustrated project in which I asked people to draw pictures of what privacy meant to them. I visited schools and community events and entreated people to draw something, even if they had no artistic skills. I paid crowd workers for drawings of privacy and collected drawings from students in my classes and people attending my talks. I now have a large collection of colorful drawings that includes many elements that do not come of much surprise: locks, doors, windows, eyes, blinds, shields, houses, and cameras.[4] And I have more than two dozen drawings featuring what is perhaps the most quintessential example of a private space: the bathroom.

I first noticed the bathroom drawings among the contributions from children: Simply drawn toilets, some with stick figures perched upon them, some with doors and siblings depicted waiting on the other side, others with heads sticking out from shower curtains or hovering above bath water. Accidental intrusions on bathroom privacy are depicted with cartoon bubble screams.

Bathroom privacy resonates with people of all ages. Adults drew feet peeking out from below public bathroom stalls and smiling stick figures enjoying a shower with a locked door (see Figure 1). While children drew the bathroom as a refuge from siblings, adults drew themselves sitting on a toilet and enjoying a break from spouses, children, and pets.[a]

IMAGE BY ANDRIJ BORYS ASSOCIATES, USING PHOTO BY AFRICA STUDIO

Quantified Toilets

The association between bathrooms and privacy in these drawings reminded me of a thought experiment carried out at CHI 2014 in which a team of hackathon participants[b] posted rather convincing signs around the Toronto Convention Center announcing that the Quantified Toilets company had installed smart toilets that were analyzing biological waste and tracking individual data (Figure 2). An accompanying website displayed a real-time stream of anonymized data, purportedly from the convention center's toilets, including sex, size, and information on odor, blood alcohol, drugs, pregnancy, and infections.

Although the website was a hoax, the thought experiment was an eye-opening experience for many CHI attendees. While it is not difficult to imagine beneficial uses of smart toilets in hospitals and homes, the idea of putting them in public places was a bridge too far for many.

A few months later, as I was writing an exam for my Usable Privacy and Security course at Carnegie Mellon University, I struggled to develop a good

[a] More drawings from the Privacy Illustrated collection are available at https://bit.ly/3olLZxp

[b] Participants included Matt Dalton, Angela Gabereau, Sarah Gallacher, Lisa Koeman, David H. Nguyen, and Larissa Pschetz.

Figure 1. A 20-year-old's drawing of privacy in a bathroom stall.

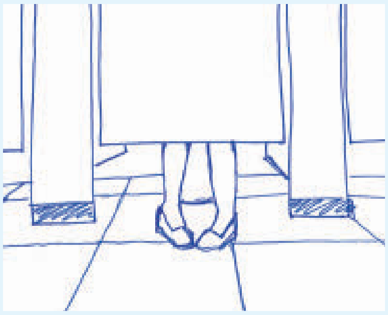
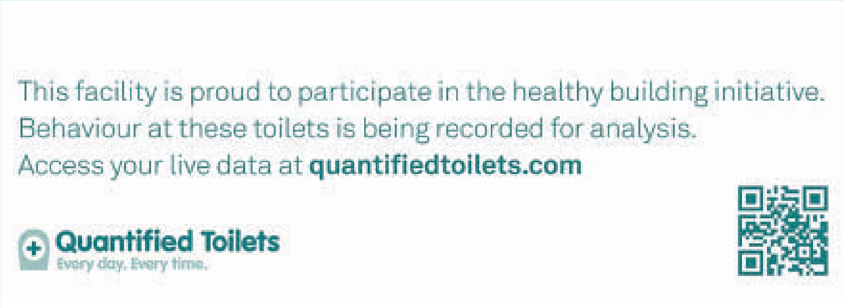


Figure 2. Notices posted at CHI 2014 alerting attendees to the presence of "quantified toilets."



This facility is proud to participate in the healthy building initiative.
Behaviour at these toilets is being recorded for analysis.
Access your live data at **quantifiedtoilets.com**

Quantified Toilets
Every day. Every time.

question that would force students to go beyond traditional policies and checkboxes when answering questions about usable privacy notices and consent experiences. Recalling Quantified Toilets, I asked my students to propose a usable approach to notice and consent for smart toilets in public bathrooms. The students' responses were thoughtful and creative. Since then, I have used this design problem as a group exercise in my university classes as well as in conference tutorials.

At conferences, I distribute markers and chart paper to attendees at each banquet table. After talking about privacy design, I tell them about some existing smart toilets, smart urinals, and bodily waste surveillance systems. Then I introduce them to the fictitious Quantified Toilets company and explain that in order to sell their toilets for use in public restrooms they need to determine how to provide notice and choice. I ask attendees at each table to develop a design proposal, thinking about how notice is displayed, and considering how the bathroom might be redesigned.

Although most participants have never heard of smart toilets (I expect that will soon change), this is a design problem they can immediately relate to, and each participant tends to come at it from a slightly different angle. After some uncomfortable laughter, the room starts to fill with eager discussion. Some start thinking about legal requirements, others think about ergonomics. Some are concerned that people will not want to touch a physical button and instead design buttons activated by voice or with foot pedals. Others note that people touch toilet flush handles anyway and propose to integrate choice mechanisms into those handles. Some are concerned that when people hang a purse or coat on a bathroom stall hook it might conceal a privacy notice. Others wonder about concerns of transgender people and how visually impaired people might be notified. Still others ponder whether children can legally provide consent and what should be done if someone uses a toilet without making a choice.

After some discussion about where to place notices and buttons, participants often consider the timing of the choice. Do people have to choose before they use the toilet? Can they choose before they flush? Can they revoke consent after they flush? Some wonder why people might be willing to consent and whether people might be interested in seeing their own data. Maybe people would like to take a copy of their data on a receipt or view it on their smartphone. If people decline to consent will they trust their data is not actually being collected? Maybe it would be better to just have two toilets: one with sensors and one without. What if there is a long line and people are coerced into using the toilet with sensors because they do not want to wait for the other one?

Although most participants have never heard of smart toilets, this is a design problem they can relate to immediately and each participant tends to come at it from a slightly different angle.

Some start to question why organizations would install these smart toilets in public restrooms. Will they be effective for monitoring the spread of disease? Indeed, wastewater testing has been used by universities, meatpacking plants, and municipalities to help pinpoint COVID-19 outbreaks.[1] In Pune, India, sensors in public toilets will provide early detection of outbreaks of cholera and other diseases as well as information about vitamin deficiencies.[2] Might this be used by employers to find out which of their employees are pregnant or on medication? Would law enforcement use data from stadiums and shopping malls to catch illegal-drug users? What other privacy issues might arise?

Some applications of smart toilets require tying toilet sensor data back to individuals. Fingerprint readers could do that, but a 2020 paper by Stanford University researchers suggests that unique anus patterns may be a more foolproof biometric, although one that may not be acceptable to users.[5]

Putting Notices to the Test

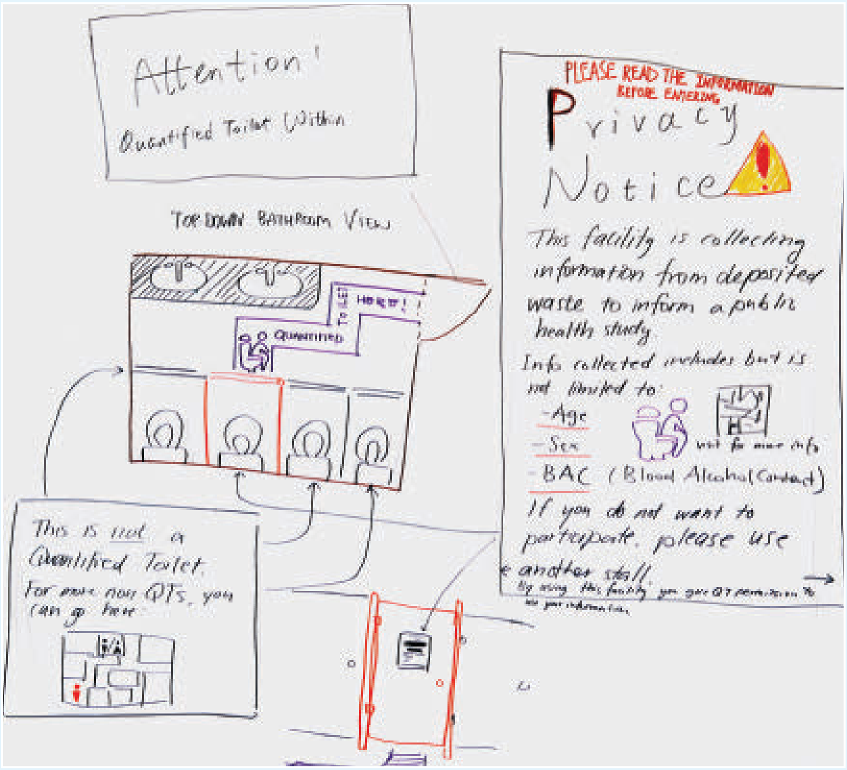
The smart toilet notice design problem also lends itself to a discussion of evaluation methods for privacy notices and consent mechanisms. Unfortunately, such evaluation is not yet the norm. Without evaluation, we are left with privacy notices that people do not understand (and most do not even try to read), and consent mechanisms that are difficult to find and often confuse people.[3] Researchers who study consumer notices emphasize the importance of evaluating disclosures through user studies.[6]

Before evaluating a smart toilet privacy notice and choice mechanism we must identify some goals and metrics. As with most notices, the purpose is to inform consumers, so we can measure the extent to which users understand the key points of the notice, as well as what their choices are. After exercising a choice, we can test whether users understood what they selected and whether their selection matched their actual preferences. These evaluations could be done in a lab or online study by presenting the notice and choice options to users. This will provide insights that will help improve wording and result in better comprehension, but user behavior in such a study may not match user reaction to a privacy notice when nature is calling.

Some applications of smart toilets require tying toilet sensor data back to individuals.

To evaluate the notice and choice mechanisms in context, we may want to set up an experiment in an actual bathroom outfitted with prototypes of the proposed notice and choice mechanisms. The toilets in the bathroom need not have working sensors—indeed, there is less risk for participants if data is not actually collected. Participants could be given an exit survey after they leave the bathroom. They may be told up-front that the smart toilet sensors are hypothetical and asked to behave as they would if they were real, or researchers may use a deceptive approach, as was done in the CHI 2014 thought experiment, and debrief study participants after they finish an exit survey. The logistics of conducting a user study in a bathroom are certainly more complicated than conducting such a study online or in a lab, but an in situ study is likely to reveal real-world factors that otherwise would not be observed. (See Figure 3.)

Figure 3. Students' smart toilet notice and choice designs.



Attention! Quantified Toilet Within

This is not a Quantified Toilet. For more non QTs, you can go here

PLEASE READ THE INFORMATION BEFORE ENTERING

Privacy Notice

This facility is collecting information from deposited waste to inform a public health study

Info collected includes but is not limited to:

- Age
- Sex
- BAC (Blood Alcohol Content)

If you do not want to participate, please use another stall.

By using this facility you give QT permission to use your information.

Pictures Worth 1,000 Words

Besides the examples I described in this column, I have many images of difficult-to-use bathroom fixtures and interesting public restroom features that I include in my usable privacy and security lectures to illustrate the need for usable privacy mechanisms. Example photos include: hotel showers I struggled to turn on and high-tech toilets with icon-laden buttons, illustrating the need for privacy controls that are intuitive or accompanied by clear instructions; sinks full of water with no apparent way to release the stopper, illustrating the need to ensure important privacy features are not hidden; compact but awkward fixtures, reminding us that inconvenient interfaces annoy users and that we should not sacrifice usability to save space; and public restrooms with glass walls that can be turned opaque at the touch of a button, which raise questions about whether people trust technology to protect their privacy.

While it may not be a topic people typically talk about (unless they are parents of young children), bathrooms are surprisingly useful for conveying concepts related to both privacy and usability. **C**

References

1. Balderrama, A.P. How Wastewater Testing Serves as An Early Warning System For COVID-19 Infection Spikes. KAWC Colorado River Public Media/Border Radio. (Feb. 9, 2021).
2. Givetash. L. and Gupta, P. India's city of Pune focuses on sanitation system of the future. NBC News (Jan. 10, 2019); https://nbcnews.to/2RSqJmV
3. 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: https://doi.org/10.1145/3313831.3376511
4. Oates, M. Turtles, locks, and bathrooms: Understanding mental models of privacy through illustration. In *Proceedings on Privacy Enhancing Technologies 4* (2018), 5–32; https://bit.ly/3y969zd
5. Park, S. et al. A mountable toilet system for personalized health monitoring via the analysis of excreta. *Nat. Biomed. Eng. 4*, (2020), 624–635; https://doi.org/10.1038/s41551-020-0534-9
6. U.S. Federal Trade Commission. Putting Disclosures to the Test. Staff Summary. (Nov. 2016); https://bit.ly/3hnJ7P6

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.

DOI:10.1145/3466612

Luke Stark et al.

▸ Susan J. Winter, Column Editor

Computing Ethics
Responsible Computing During COVID-19 and Beyond

Navigating the ethical and societal impacts of technologies.

THE COVID-19 PANDEMIC has both created and exacerbated a series of cascading and interrelated crises whose impacts continue to reverberate. From the immediate effects on people's health to the pressures on healthcare systems and mass unemployment, millions of people are suffering. For many of us who work in the digital technology industry, our first impulse may be to devise technological solutions to what we perceive as the most urgent problems when faced by crises such as these. Although the desire to put our expertise to good use is laudable, technological solutions that fail to consider broader social, political, and economic contexts can have unintended consequences, undermining their efficacy and even harming the very communities that they are intended to help.[10] To ensure our contributions achieve their intended results without causing inadvertent harm, we must think carefully about which projects we work on, how we should go about working on them, and with whom such work should be done. In this column, we offer a series of guidelines for navigating these choices. As current and former members of the Fairness, Accountability, Transparency, and Ethics (FATE) group at Microsoft Research, we have been working actively on the ethical and societal impacts of technologies such as artificial intelligence since 2016. While we originally developed these guidelines to help our colleagues at Microsoft respond to the first wave of the pandemic in the spring of 2020, we believe they are general enough that their value extends beyond Microsoft and beyond projects focused on the COVID-19 pandemic.

IMAGE FROM SHUTTERSTOCK.COM

▸ **Ask yourself if your project is worth pursuing.** Before investing in your project, do a risk–benefit analysis.[8] Are there other responses (technological or otherwise) that would have a greater impact with fewer potential downsides? This is an important question to ask when trying to address problems that are more societal than technological in nature. Depending on the answer, proceeding with your project may not be the right decision after all.

▸ **Question your assumptions about contexts.** At the start of your project, question the assumptions you are making about the social, political, and economic contexts in which your technological response will take place. For example, consider a contact-tracing project based on cellphones.[5] Does it assume: everyone will have access to the same digital tools, such as smartphones; people are willing to share personal information;[3] people understand the risks of doing so and are comfortable accepting those risks; people will be able to give truly voluntary consent if the response is adopted by employers, schools, or governments; there is widely available testing; people have sufficient financial resources and social support to self-quarantine; and people can afford medical care? In other words, ask what other institutional processes and structures need to be in place for your technological response to work effectively.

▸ **Collaborate with experts in other disciplines.** Recognize the limitations of your own expertise. For many of us who work in the technology industry, it is easy to assume that technological responses, such as tracking people's locations, collecting information about their contacts, and issuing "immunity" passports, are clearly worth pursuing. Yet the usefulness of such approaches is contested by both public health and privacy experts.[7] In many cases, you can have the greatest impact by finding experts who know more than you do about a problem, asking them what they need to make progress, and then helping them accomplish their goals.

▸ **Be clear about expected benefits and beneficiaries.** Think carefully about what your project specifically offers, how it will be beneficial, and whether those benefits will be widely accessible to those who need them. In many cases, the intended beneficiaries may not be on a level playing field. For example, the enormous racial differences in health outcomes observed during the pandemic illustrate how existing societal inequalities affect who suffers and in what ways.[6] Does your project take these dynamics into account and work to mitigate them?

If your project relies on data, ask where that data came from and how it was collected.

▸ **Work with and for communities.** Ask the intended beneficiaries of your project—whether they are health-care workers, public health experts, or senior citizens—if your project addresses their needs. If you believe that you have additional insights, have you presented them with evidence that supports your beliefs and asked them for their input? You should provide opportunities for communities to collaboratively shape the project, give ongoing feedback and voice their concerns, and make informed decisions for themselves.

▸ **Mitigate risks.** Try to anticipate the risks posed by your project and how such risks might impact different communities. For example, new technologies to facilitate working from home will also provide new opportunities for companies to track and monitor workers. In many cases, the communities that are most vulnerable to the pandemic are also those that are most at risk of being harmed by technological responses.[4]

▸ **Understand and protect your data.** Many technological responses to the pandemic either rely on or collect data, including data about people's health and locations. If your project relies on data, ask where that data came from and how it was collected. Did you consider the unusual circumstances under which the data might have been generated?[2] What are the resulting limitations, if any? Does the data capture what you need or what you think it captures? Does it reflect a representative sample of the relevant population (for example, the intended

beneficiaries)? Does the data involve restrictive, problematic, or harmful classifications, such as only binary genders? If your project collects data, ask whether this will pose risks, perhaps resulting from unanticipated uses or abuse. For example, when correlated with other data, the data collected by a contact-tracing project could be used to identify and persecute undocumented immigrants. Failure to guard against these risks will limit people's willingness to rely on your project and may undermine the solidarity needed to maintain public health. Consequently, privacy and security must be paramount.[1]

▸ **Have a plan for when and how your project will end.** A crucial—yet commonly overlooked—feature of any project is a plan for when and how it will end. If you decide to proceed with your project, ask how long it should last. When you have an answer, you can then plan for a "graceful dismantling."[9] If you are not able to devise such a plan, you should reconsider your decision to proceed. For example, systems built to support contact tracing during the pandemic can be repurposed for other goals or kept in place even after the pandemic is over. Your plan should therefore include controls—technological, legal, or otherwise—that enable you to limit the functionality of your project to its intended purpose for the desired duration. Your plan should also address what will be done with the data when your project is over. Equally, people may come to depend on your project: Will ending it harm the intended beneficiaries? If so, how will you guard against this?

Following these guidelines can make it more likely that projects achieve their goals, while minimizing harm—helping their intended beneficiaries and other communities, without putting them at risk. C

References

1. Brill, J. and Lee, P. Preserving privacy while addressing COVID-19. *Microsoft On The Issues.* (2020); https://bit.ly/33QKspB
2. Crawford, K. and Finn, M. The limits of crisis data: Analytical and ethical challenges of using social and mobile data to understand disasters. *GeoJournal 80*, 4 (2015), 491–502; https://bit.ly/3tWv3i0
3. Langford, J. Critical issues in digital contract tracing. *Machine Learning Theory.* (2020); https://bit.ly/3eS5nPo
4. National Research Council U.S. Panel on Monitoring the Social Impact of the AIDS Epidemic. The practice of public health. In A.R. Jonsen and J. Stryker, Eds. *The Social Impact Of AIDS In The United States*, Washington, D.C., 1993; https://bit.ly/3wirxjV
5. O'Neil, C. The Covid-19 tracking app won't work. *Bloomberg.* (2020); https://bloom.bg/3yegtGl
6. Owen, W.F. Jr, Carmona, R., and Pomeroy, C. Failing another national stress test on health disparities. *JAMA 323*, 19 (2020), 1905–1902; https://bit.ly/3eQWoOt
7. Soltani, A., Calo, R., and Bergstrom, C. Contact-tracing apps are not a solution to the COVID-19 crisis. *Brookings TechStream.* (2020); https://brook.gs/3bwhmjO
8. The National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. *The Belmont Report: Ethical Principles and Guidelines for the Protection of Human Subjects of Research. Department of Health, Education, and Welfare* (Apr. 18, 1979); https://bit.ly/3opJKsR
9. Veale, M. This system was designed so it could not be co-opted. (2020); https://bit.ly/3tPKade
10. Winner, L. Do artifacts have politics? In *The Whale and the Reactor*. University of Chicago Press, Chicago, 1988, 19–39.

Solon Barocas (sbarocas@cornell.edu) is Principal Researcher at Microsoft Research and Adjunct Assistant Professor, Information Science, Cornell University, Ithaca, NY, USA.

Asia J. Biega (jbiega@mpi-inf.mpg.de) is a member of the tenure-track faculty at Max Planck Institute for Security and Privacy, Bochum, Germany.

Margarita Boyarskaya (mboyarsk@stern.nyu.edu) is a doctoral candidate at the NYU Stern School of Business, New York, NY, USA.

Kate Crawford (kate@microsoft.com) is Senior Principal Researcher at Microsoft Research and Research Professor of Communication and STS, USC Annenberg, Los Angeles, CA, USA.

Hal Daumé III (hal3@microsoft.com) is Principal Researcher at Microsoft Research and Professor of Computer Science at the University of Maryland-College Park, MD, USA.

Miroslav Dudík (mdudik@microsoft.com) is Senior Principal Researcher at Microsoft Research, New York, NY, USA.

Benjamin Fish (benjamin.s.fish@gmail.com) is Postdoctoral Fellow, Mila–Quebec AI Institute, Montreal, Canada.

Mary L. Gray (mlg@microsoft.com) is Senior Principal Researcher, Microsoft Research, New England/Faculty Associate, the Berkman Klein Center for Internet and Society Harvard University/Associate Professor, Luddy School of Informatics, Computing and Engineering, Indiana University, Bloomington, IN, USA.

Brent Hecht (Brent.Hecht@microsoft.com) is Director of Applied Science, Experiences and Devices, Microsoft/Associate Professor, Human-Centered AI and Spatial Computing, Northwestern University, Evanston, IL, USA.

Alexandra Olteanu (Alexandra.Olteanu@microsoft.com) is Principal Researcher, Microsoft Research, Montreal, Canada.

Forough Poursabzi-Sangdeh (fpoursabzi@microsoft.com) is Senior Program Manager, Office of Chief Scientific Officer, Microsoft, New York, NY, USA.

Luke Stark (cstark23@uwo.ca) is Assistant Professor, Faculty of Information and Media Studies, University of Western Ontario, London, Canada.

Jennifer Wortman Vaughan (jenn@microsoft.com) is Senior Principal Researcher, Microsoft Research, New York, NY, USA.

Hanna Wallach (wallach@microsoft.com) is Senior Principal Researcher, Microsoft Research, New York, NY, USA.

Marion Zepf (mazepf@microsoft.com) is a software developer on the Turing Team at Microsoft in Montreal, Canada.

DOI:10.1145/3433638

Josep Domingo-Ferrer, David Sánchez, and Alberto Blanco-Justicia

Viewpoint
The Limits of Differential Privacy (and Its Misuse in Data Release and Machine Learning)

Differential privacy is not a silver bullet for all privacy problems.

THE TRADITIONAL APPROACH to statistical disclosure control (SDC) for privacy protection is utility-first. Since the 1970s, national statistical institutes have been using anonymization methods with heuristic parameter choice and suitable utility preservation properties to protect data before release. Their goal is to publish analytically useful data that cannot be linked to specific respondents or leak confidential information on them.

In the late 1990s, the computer science community took another angle and proposed privacy-first data protection. In this approach a privacy model specifying an ex ante privacy condition is enforced using one or several SDC methods, such as noise addition, generalization, or microaggregation. The parameters of the SDC methods depend on the privacy model parameters, and too strict a choice of the latter may result in poor utility. The first widely accepted privacy model was k-anonymity, whereas differential privacy (DP) is the model that currently attracts the most attention.

DP was originally proposed for interactive statistical queries to a database.[5] A randomized query function κ (that returns the query answer plus some noise) satisfies ϵ-DP if for all datasets D_1 and D_2 that differ in one record and all $S \subset Range(\kappa)$, it holds that $\Pr(\kappa(D_1) \in S) \leq \exp(\epsilon) \times \Pr(\kappa(D_2) \in S)$. In other words, the presence or absence of any single record must not be noticeable from the query answers, up to an exponential factor of ϵ. The smaller ϵ, the higher the protection. The noise to be added to the answer to enforce a certain ϵ depends on the global sensitivity of the query to the presence or absence of any single record. Mild noise may suffice for statistical queries such as the mean, while a very large noise is needed for identity queries returning the contents of a specific record.

IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

DP offers a very neat privacy guarantee and, unlike privacy models in the k-anonymity family, does not make assumptions on the intruder's background knowledge (although it assumes all records in the database are independent.[3,10] For this reason, DP was rapidly adopted by the research community to the point that previous approaches tend to be regarded as obsolete. Researchers and practitioners have extended the use of DP beyond the interactive setting it was designed for. Extended uses include: data release, where privacy of respondents versus data analysts is the goal, and collection of personal information, where privacy of respondents versus the data collector is claimed. Google, Apple, and Facebook have seen the chance to collect or release microdata (individual respondent data) from their users under the privacy pledge "don't worry, whatever you tell us will be DP-protected."

However, applying DP to record-level data release or collection (which is equivalent to answering identity queries) requires employing a large amount of noise to enforce a safe enough ϵ. As a result, if $\epsilon \leq 1$ is used, as recommended in Dwork and Roth[5] to obtain a meaningful privacy guarantee, the analytical utility of DP outputs is likely to be very poor.[2,7] This problem arose as soon as DP was moved outside the interactive setting. A straightforward way to mitigate the utility problem is to use unreasonably large ϵ.

Let us look at data collection. Apple reportedly uses $\epsilon = 6$ in MacOS and $\epsilon = 14$ in iOS 10 (with some beta versions using even $\epsilon = 43$).[9] In their RAPPOR technology, Google uses ϵ up to 9. According to Frank McSherry, one of the co-inventors of DP, using ϵ values as high as 14 is pointless in terms of privacy.[9] Indeed, the privacy guarantee of DP completely fades away for such large values of ϵ.[5]

As to data release, Facebook has recently released DP-protected datasets for social science research, but it is unclear which ϵ value they have used. As pointed out in Mervis,[12] this makes it difficult both to understand the privacy guarantees being offered and to assess the trustworthiness of the results obtained on the data. The first released version of this DP dataset had all demographic information about respondents and most of the time and location information removed, and event counts had been added noise with $\sigma = 200$.[6] The second version looks better, but is still analytically poorer than the initial version released in 2018 under utility-first anonymization based on removal of identifiers and data aggregation. In fact, Facebook researchers have acknowledged the difficulties of implementing (and verifying) DP in real-world applications.[11]

The U.S. Census Bureau has also announced the use of DP to disseminate Census 2020 results.[8] A forecast of the negative impact of DP on the utility of the current Census is given in Santos-Lazoda et al.[14] Additionally, Ruggles et al.[13] have remarked that DP "is a radical departure from established Census Bureau confidentiality laws and precedents": the Census must take care of preventing respondent re-identification, but masking respondent characteristics—as DP does—is not required. A more fundamental objection in Ruggles et al.[13] is against the very idea of DP-protected microdata. As introduced previously, publishing useful record-level microdata under DP is exceedingly difficult. This is only logical: releasing DP-microdata, that is, individual-level data derived from real people, contradicts the core idea of DP, namely that the presence or absence of any individual should not be noticeable from the DP output.

A further shortcoming arises when trying to use DP to protect continuous data collection like Apple and Google do. DP is subject to sequential composition: if a dataset collected at time t_1 is DP-protected with ϵ_1 and a dataset collected at time t_2 on a non-disjoint group of respondents is DP-protected with ϵ_2, the dataset obtained by composing the two collected datasets is DP-protected only with $\epsilon_1 + \epsilon_2$. Therefore, to enforce a certain ϵ after n data collections on the same set of individuals, each collection should be DP-protected with ϵ/n, thereby very substantially reducing the utility of the collected data. Strictly speaking, it is impossible to collect DP-protected data from a community of respondents an indefinite number of times with a meaningful privacy guarantee. As a remedy, Apple made the simplification that sequential composition only applies to the data collected on an individual during the same day but not in different days.[9] Google took a different way out: they use sequential composition only for values that have not changed from the previous collection.[4] Both fixes are severely flawed: the data of an individual collected across consecutive days or that may have changed still refer to the same individual, rather than to disjoint individuals. Ignoring this and conducting a systematic data collection for long periods increases the effective ϵ and thus reduces the effective level of protection by several orders of magnitude. This issue is significantly more privacy-harming than the previously mentioned large values of ϵ declared by Apple and Google.

Machine learning (ML) has also seen applications of DP. In Abadi et al.,[1] DP is used to ensure deep learning models do not expose private information contained in the datasets they have been trained on. Such a privacy guarantee is interesting to facilitate crowdsourcing of representative training data from individual respondents. The paper describes the impact of sequential composition

Fundamental misunderstandings and flawed implementations pervade the application of differential privacy to data releases, data collection, and machine learning.

over ML training epochs (which can be viewed as continuous data collection) on the effective ϵ: after 350 epochs, a very large $\epsilon = 20$ is attained. To obtain usable results without rising to such a large ϵ, the authors use the (ϵ,δ) relaxation of DP that keeps ϵ between 2 and 8. Employing relaxations of DP to avoid the "bad press" of large ϵ while keeping data usable is a common workaround in recent literature.[1,15] However, DP relaxations are not a free lunch: relaxed DP is not DP anymore. For example, with (ϵ,δ)-DP, "δ values on the order of $1/|D|$ (where $|D|$ is the size of the dataset) are very dangerous: they permit preserving privacy by publishing the complete records of a small number of database participants."[5] The value of δ employed in Abadi et al.[1] and Triastcyn and Faltings[15] is in fact on the order of $1/|D|$, thereby incurring severe risk of disclosure. Nonetheless, despite the use of relaxations and large ϵ and δ, the impact of DP on data utility remains significant: in Abadi et al.[1] the deep learning algorithm without privacy protection achieves 86% accuracy on the CIFAR-10 dataset, but it falls down to 73% for $\epsilon = 8$ and to 67% for $\epsilon = 2$.

Very recently, DP has also been proposed for a decentralized form of ML called federated learning. Federated learning allows a model manager to learn a ML model based on data that are privately stored by a set of clients: in each epoch, the model manager sends the current model to the clients, who return to the manager a model update based on their respective private datasets. This does not require the clients to surrender their private data to the model manager and saves computation to the latter. However, model updates might leak information on the clients' private data unless properly protected. To prevent such a leakage, DP is applied to model updates.[15,16] However, in addition to distorting model updates, using DP raises the following issues:

- Since model updates are protected in each epoch, and in successive epochs they are computed on the same (or, at least, not completely disjoint) client data, sequential composition applies. This means that the effective ϵ grows with the number of epochs, and the effective protection decreases exponentially. Therefore, reasonably useful models can only be obtained for meaninglessly large ϵ (such as 50–100[16]).
- In the original definition of DP, a dataset where each record contains the answer of a different respondent is assumed. Then DP ensures the record contributed by any single respondent is unnoticeable from the released DP-protected output. This protects the privacy of any single respondent. However, when DP is used to protect the model update submitted by a client, all records in the client's dataset belong to the client. Making any single record unnoticeable is not sufficient to protect the client's privacy when all records in the client's private dataset are about the client, as it happens for example, if the client's private data contain her health-related or fitness measurements. Thus, the DP guarantee loses its significance in this case.

Conclusion

Differential privacy is a neat privacy definition that can co-exist with certain well-defined data uses in the context of interactive queries. However, DP is neither a silver bullet for all privacy problems nor a replacement for all previous privacy models.[3] In fact, extreme care should be exercised when trying to extend its use beyond the setting it was designed for. As we have highlighted, fundamental misunderstandings and blatantly flawed implementations pervade the application of DP to data releases, data collection, and machine learning. These misconceptions have serious consequences in terms of poor privacy or poor utility and they are driven by the insistence to twist DP in ways that contradict its own core idea: to make the data of any single individual unnoticeable.

References

1. Abadi, M. et al. Deep learning with differential privacy. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security-CCS'16* (2016), 308–318.
2. Bambauer, J., Muralidhar, K., Sarathy, R. Fool's gold: An illustrated critique of differential privacy. *Vanderbilt Journal of Entertainment & Technology Law 16*(, 4 (2014), 701–755.
3. Clifton, C., Tassa, T. On syntactic anonymity and differential privacy. *Transactions on Data Privacy 6*, 2 (2013), 161–183.
4. Cyphers, B. Differential privacy, part 3: Extraordinary claims require extraordinary scrutiny. Accessnow (Nov. 30, 2017); https://bit.ly/3oqOTku
5. Dwork, C., Roth, A. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science 9*, 3–4, (2014).
6. Francis, P. Dear differential privacy, put up or shut up. Technical Report MPI-SWS-2020-005 (Jan. 2020); https://bit.ly/3whnpkc
7. Fredrikson, M. et al. Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Security Symposium* (2014), 17–32.
8. Garfinkel, S., Abowd, J.M., Martindale, C. Understanding database reconstruction attacks on public data. *Commun. ACM 62*, 3 (Mar. 2019), 46–53.
9. Greenberg, A. How one of Apple's key privacy safeguards falls short. *Wired* (Sept. 15, 2017); https://bit.ly/2RsaLjr
10. Kifer, D.and Machanavajjhala, A. No free lunch in data privacy. In *Proceedings of the SIGMOD Conference 2011* (2011), 193—204.
11. Kifer, D. et al. Guidelines for implementing and auditing differentially private systems. (Feb. 10, 2020); https://bit.ly/2RwMHvH
12. Mervis, J. Researchers finally get access to data on Facebook's role in political discourse. *Science* (Feb. 13, 2020); https://bit.ly/3ynS7Kj
13. Ruggles, S. et al. Differential privacy and Census data: implications for social and economic research. *AEA Papers and Proceedings, 109* (2019), 403–408.
14. Santos-Lozada, A.R., Howard, J.T. and Verdery, A.M. How differential privacy will affect our understanding of health disparities in the United States. In *Proceedings of the National Academy of Sciences 117*, 24) (2020), 13405–13412.
15. Triastcyn, A. and Faltings, B. Federated learning with Bayesian differential privacy. In *Proceedings of 2019 IEEE Intl. Conf. on Big Data* (2019), 2587–2596.
16. Wei, K. et al. Federated learning with differential privacy: algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security, 15* (2020), 3454–3469.

Josep Domingo-Ferrer (josep.domingo@urv.cat) is a Distinguished Professor of Computer Science and an ICREAAcadèmia Researcher at Universitat Rovira i Virgili, Tarragona, Catalonia, where he holds the UNESCO Chair in Data Privacy and leads CYBERCAT.

David Sánchez (david.sanchez@urv.cat) is an associate professor and an ICREA-Acadèmia Researcher at Universitat Rovira i Virgili, Tarragona, Catalonia.

Alberto Blanco-Justicia (alberto.blanco@urv.cat) is a senior postdoc at Universitat Rovira i Virgili, Tarragona, Catalonia.

The authors received support from the European Commission (projects H2020-871042 "SoBigData++" and H2020-101006879 "MobiDataLab"), the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and D. Sánchez, and grant 2017 SGR 705), the Spanish Government (projects RTI2018-095094-B-C21 "Consent" and TIN2016-80250-R "Sec-MCloud") and the Norwegian Research Council (project no. 308904 "CLEANUP"). The authors are associated with the UNESCO Chair in Data Privacy, but the views in this Viewpoint are the authors' own and not necessarily shared by UNESCO.

DOI:10.1145/3437254

Diomidis Spinellis

Viewpoint
Why Computing Students Should Contribute to Open Source Software Projects

Acquiring developer-prized practical skills, knowledge, and experiences.

Learning to program is—for many practical, historical, as well as some vacuous reasons—a rite of passage in probably all computer science, informatics, software engineering, and computer engineering courses. For many decades, this skill would reliably set computing graduates apart from their peers in other disciplines. In this Viewpoint, I argue that in the 21st century programming proficiency on its own is neither representative of the skills that the marketplace requires from computing graduates, nor does it offer the strong vocational qualifications it once did. Accordingly, I propose that computing students should be encouraged to contribute code to open source software projects through their curricular activities. I have been practicing and honing this approach for more than 15 years in a software engineering course where open source contributions are an assessed compulsory requirement.[2] Based on this experience, I explain why the ability to make such contributions is the modern generalization of coding skills acquisition, outline what students can learn from such activities, describe how an open source contribution exercise is embedded in the course, and conclude with practices that have underpinned the assignment's success.

IMAGE BY BAKHTIAR ZEIN

Contributing Is the New Coding

Programming skills nowadays are only a part of what a software developer should know. This is the case for two reasons. First, practices have advanced well beyond the chief programmer/surgeon model popularized by Fred Brooks in the 1970s,[1] to include work on orders of magnitude larger systems, advanced tooling, pervasive process automation, as well as sophisticated teamwork, workflows, and management. Second, industrial best practices have homogenized with those followed by large and successful open source software projects. Businesses have assimilated and contributed many open source development practices. This has made the corresponding knowledge and skills portable between volunteer projects and enterprise ones.

Consequently, instruction must move from a course's educational laboratory to an organizational setting. By contributing to open source projects, students acquire *in practice* a formidable range of skills, knowledge, and experiences, allowing them to work productively as modern well-rounded developers rather than as the lone-wolf coders portrayed by Hollywood. The most difficult skills to acquire in a traditional programming assignment are the following social and organizational skills.

- Developing a sense of context: understanding how development work is embedded within a project's scope, mission, team of co-developers, and new forms of leadership;
- Interacting with a project's global and diverse community;
- Negotiating feature requests, requirements, and implementation choices;
- Dealing with communication problems, such as absent responses, which are common in volunteer-run projects;
- Appreciating the software as a product through practices such as issue triaging and release planning; and
- Receiving, discussing, and addressing code review comments.

Corresponding learning outcomes associated with technology range from analysis and evaluation to application and creation, including the following:

- Navigating through a project's assets, such as software code, issues, documentation, and pull requests;
- Evaluating swiftly the product and process quality of software systems or components, as is often required in modern software reuse;
- Configuring, building, running, and debugging third-party code;
- Setting up and running software intensive systems with diverse software and hardware requirements. In the course I run, these have included mobile phones, car electronics, application servers, databases, containers, IoT equipment, and embedded devices;
- Choosing realistic contribution goals. (Initially students tend to wildly overestimate their ability to contribute to a project.) This is a key activity in agile development sprints;
- Reading third-party code to identify where their additions or fixes need to be made;
- Modifying a large third-party system by adding a new feature or fixing a bug;
- Writing tests that demonstrate a contribution is working as expected now and into the future;
- Working with software systems developed using multiple programming languages and tools; students are often surprised to find out that knowledge of an Integrated Development Environment (IDE) is by no means a passport for contributing to a project;
- Documenting their work, typically using a declarative markup language, for example Markdown or documentation generator code comments;
- Following sophisticated configuration management (version control) workflows, such as working on issue branches and rebasing code commits; and
- Passing pre-commit and continuous integration checks and tests.

We assess the students' performance based on their open source project work available online, their final written report, and their in-class presentations.

Both the social and technical learning outcomes are very relevant in the modern workplace—and they go well beyond the proposed ACM/IEEE curriculum for software engineering programs.[3] In parallel, the course's practices embrace many of the ACM/IEEE curriculum guidelines as cross-cutting concerns. These include: the exercising of personal skills, such as critical judgment, effective communication, and the recognition of one's limitations (Curriculum Guideline 8); developing skills for self-directed learning (CG 9); appreciating the multiple dimensions of software engineering problem solving (CG 10); using appropriate and up-to-date tools (CG 12); having a real-world basis (CG 14); and educating through a variety of teaching and learning approaches (CG 18).

Embedding Open Source Development in a Software Engineering Course

The compulsory open source software contribution assignment is part of a third-year course titled "Software Engineering in Practice." (The course received the Management School's Excellence in Teaching award in 2019.) We teach this course to about 20–50 students each year who follow the "Software and Data Analytics Technologies" specialization offered by the Athens University of Economics and Business Department of Management Science and Technology. The course is also a recommended elective for the university's Department of Informatics.

The course is delivered using a (light) flipped classroom approach[4] and is entirely assessed through coursework. The open source contribution assignment counts for 50% of the course's grade. Students can work alone or in pairs. Pairing aims to help students who may feel insecure on their own, though in such cases the pair must deliver more work than an individual, and the contributions must be made from separate GitHub accounts.

We assess the students' performance based on their open source project work available online (code commits and interactions), their final written report, and their in-class presentations. Three presentations take place approximately on week 4 (describing the selected project), week 8 (outlining the proposed contributions), and week 14 (summarizing the contributions' implementation). Getting a contribution accepted is not a prerequisite for passing the assignment, but it is positively assessed. Other assessed elements include the students' comprehension and documentation of the project they chose, their contribution's breadth, their implementation's quality, their code's integration with the project, their testing implementation, their collaboration with the project's development team, their oral presentations, the quality of their written report, and their use of the available tooling for activities such as version control, code reviews, issue management, and documentation.

Cheating (by copying a contribution from a project fork, or farming out work) could, in theory, be an issue; it is countered by having students present their work in class, and by knowing that their (public) contributions become part of their work portfolio and may be quizzed by future prospective employers.

The course benefits each year from one or two dedicated teaching assistants who run laboratory sessions on key tools, and are available during office hours to advise the students on difficulties they invariably face. The hard work they put into supporting the course, means that increasing the number of attending students would require a commensurate increase in teaching assistants.

Ensuring Successful Open Source Contributions

Students approach the course and its assignment with trepidation and complete it with jubilation. Ensuring students can make meaningful contributions to an open source project requires balancing their inexperience with the fast-paced sophistication of modern, open source software development.

Throughout the years I have given out the assignment, I have seen that contributing to open source projects has become easier. Projects are becoming more inclusive. Many projects have streamlined on-boarding and mentoring, teams are more diverse (including female leads), a published code of contact is common, responses are typically polite, and Windows builds are often supported (though some students adopt Linux to avoid glitches). Contributing has been simplified thanks to handholding in pull request workflows, widespread adoption of continuous integration, diverse code check bots, friendly code review processes, and the use of draft pull requests to allow incremental reviewing of work in progress.

The open source project environment the students dive into is very far apart from the one they typically experience in traditional academic assignments.

Still, the open source project environment the students dive into, is very far apart from the one they typically experience in traditional academic assignments. Therefore, a small-scale contribution is the only realistic goal. The key to making the course's assignment work, is to have what are, on first sight, very low ambitions for the students' contributions. To an undergraduate student, the barriers to open source contributions are often so high, that getting 20 lines of code integrated into a large project is a worthwhile achievement indeed. The advice we give our students for choosing a project can be summarized as follows.

- Choose a project with several active contributors, so that there is a community to guide you and respond to your questions.
- Choose a relatively popular project (some GitHub stars) demonstrating that it provides useful functionality and is developed in a relatively sound way. You want to avoid an abandoned thesis project uploaded on GitHub.
- Avoid very popular projects, so that your contributions will not get drowned in competition, noise, and bureaucracy. (Despite this, we have had students contributing to blockbuster projects, such as Tensorflow and Visual Studio Code.)
- Verify that you can build and run the project on your computer setup.
- Ensure the project regularly accepts pull requests from outsiders, so that yours will also have a chance.
- Try to contribute a trivial fix as a warm-up exercise and as a way to test your ability to follow the project's workflows.
- Look for project issues marked as "Good first issue," which indicate a project that is open to new contributors. (There are several online lists of projects with such issues.)

We leave the choice of the contribution entirely to the students: they can pick an open task from the project's issue database, or propose their own enhancement or fix. Students also often change tack after interacting with the project's core team. Although their freedom to choose their contribution may appear to make the assignment too easy, we have found that it makes it easy enough so that about half of the student contributions get integrated.

The most common problems faced by the students over their assignment are the inability to build the project (typically due to inexperience and platform incompatibilities) and a lack of communication by the project's team (students get needlessly anxious, thinking that their work must be integrated into the project). On the flip-side, the biggest delight felt by the students is when they find their code integrated into production software used worldwide. Invariably, in the course evaluation students comment favorably on the many practical skills and self-confidence they gain after they complete their open source software contribution assignment.

References

1. Brooks, F.P., Jr. *The Mythical Man-Month.* Addison-Wesley, Boston, MA, 1975, 32.
2. Spinellis, D. Future CS course already here. *Commun. ACM 49*, 8 (Aug. 2006), 13; https://bit.ly/3bYxSJs
3. The Joint Task Force on Computing Curricula. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. ACM. New York, NY; https://bit.ly/3vn04NP
4. Tucker, B. The flipped classroom. *Education Next 12*, 1 (Mar. 2012), 82–83.

Diomidis Spinellis (dds@aueb.gr) is a professor of software engineering in the Department of Management Science and Technology at the Athens University of Economics and Business, Greece, and a professor of software analytics in the Department of Software Technology at the Delft University of Technology, the Netherlands.

Many thanks to my colleagues Serge Demeyer, Michael Greenberg, and Angeliki Poulymenakou, as well as to the former course students Zoe Kotti and Christos Pappas for their detailed and constructive comments on earlier versions of this Viewpoint.

DOI:10.1145/3440753

Travis Breaux and Jennifer Moritz

Viewpoint
The 2021 Software Developer Shortage Is Coming

Companies must address the difficulty of hiring and retaining high-skilled employees from an increasingly smaller labor supply.

WHILE GLOBAL ECONOMIES of all scales have been severely impacted by the COVID-19 global pandemic, information technology (IT) companies have managed to survive much of the economic downturn. This is due in large part to their past success in distributed IT development, remote IT operations, and remote maintenance. IT companies have required their staff to essentially do at home what they had been doing in the office years before. Moreover, to meet the constraints of working from home, the demand for IT services has increased across many other market sectors, including retail, entertainment, education, and healthcare, leading to a more profitable IT market.

Tightening in the Upstream IT Labor Supply

While this news is largely positive for IT companies, the pandemic is also taking its toll on the upstream supply of IT labor, and software developers in particular. These effects could constrain hiring and innovation over the next two to three years. Due in part to travel embargos, limited access to educational loans, and delays in student visa processing, U.S. colleges and universities are observing significant declines in graduate-level enrollments in computer and information science this year. These declines are translating into one-year enrollment deferrals, meaning a temporary, but significant reduction in expected 2021 graduation rates.

Two years ago, in 2019, U.S. colleges and universities awarded more than 136,000 bachelor's, master's, and Ph.D. degrees in computer and information science (CIS).[6] Among which, 35,200 of these degrees were awarded to nonresident aliens, among whom 27,200 or 77% earned either a master's or doctoral degree, placing these graduates in a higher skilled technology category. Recently, the American Council on Education (ACE) reported a 43% decline in pandemic-era enrollments for international students based on recent surveys.[5] This decline will reduce CIS post-graduates by 11,700 students in 2021, and would disproportionately affect higher-skilled graduates with more than two years of industry experience. While this story is still playing out, we should already anticipate a tighter IT labor market, particularly among higher-skilled technology jobs.

Increased Competition for Highly Skilled IT Labor

Prior to the pandemic, the U.S. Bureau of Labor Statistics (BLS) estimated that approximately 1,469,200 full-time software developers were working in the U.S. in 2019, earning an average salary of $107,510. Over the next 10 years, BLS estimates the U.S. labor market will add

IMAGE BY DMITRY KOVALCHUK

316,000 software developer jobs, which is a phenomenal 22% growth rate, or an average of about 31,000 new jobs each year. By comparison, lesser-skilled computer programmer jobs would decline by 9% to fewer than 193,800 jobs by 2029, showing a continued shift in the U.S. labor market toward higher-skilled IT jobs that include the engineering practices of software design, software construction and maintenance. The 43% decline in new CIS graduates reported by the ACE would reduce BLS estimated job growth by 38% in 2021 by leaving thousands of jobs unfilled.

In our review of published 2018 data from the Student and Exchange Visitor Program (SEVP), 61% of the top 200 companies that hire graduates under Optional Practical Training (OPT) conduct business in management consulting, and information technology, including retail, finance, and healthcare;[9] see the sidebar here. This includes popular consulting companies, such as Deloitte, PriceWaterhouseCoopers, and KPMG, as well as popular technology companies, such as Amazon, Apple, Google, and Microsoft. We estimate that in 2018 there were 46,700 new workers authorized under STEM OPT who were potentially working in IT-related fields. If true, then a decline of 11,700 CIS graduates in 2021 could reduce the available OPT-authorized IT workforce by as much as 25%.

The Impact of Labor Shortages on IT and Innovation

When labor-supply shortages arise in IT, the impact can negatively affect both productivity and innovation. In a survey of electronic and mechanical engineering and IT firms in Northern Ireland, Bennett and McGuinness found that skill shortages in hard-to-fill and unfilled vacancies reduce firm-level performance.[1] Hard-to-fill vacancies include those requiring more experience and qualifications, such as IT positions that require at least two years of experience, which are reported to be 40% more difficult to fill than entry-level positions. Unfilled vacancies are measured by the raw number of open positions within the last 12 months, which is independent of how recruiters interpret "difficult to hire." In general, Bennett and McGuinness found that hard-to-fill vacancies reduced productivity levels by 65%, while unfilled vacancies reduced productivity levels by 75%. This productivity loss is amplified when the company has a large proportion of new graduates and inexperienced workers, or when the company has significant investments in R&D, which otherwise drives up productivity when labor shortages are minimal. Substantiating the impact of labor shortages on innovation, Horbach and Rammer analyzed three years of the German Innovation Survey and found that unfilled positions in high skill areas correlates with increased canceling of innovation projects.[4]

Unfilled positions lead to increased churn as skilled employees jockey for better jobs at more attractive companies. What makes a company attractive to a prospective employee is driven in part by the skills that a prospective hire feels they can develop on the new job. Based on an analysis of employment data from over 50,000 workers, Tambe et al. found that IT employees accept a "compensating differential," such as trading pay for other job attributes, when they can work with novel and emerging IT systems and develop their skills on the job.[7] Furthermore, their

Optional Practical Training

For nonresident aliens to work in the U.S. after graduation, they must first have a valid work authorization, which has historically included Optional Practical Training (OPT).[a] The OPT began in 1953 to authorize international students to work for a period of 6–18 months as determined by their school and training agency (18 Fed. Reg. at 3,529). This program has since evolved to meet the demands of modern educational and workforce requirements. Today, international students enrolled in bachelor's, master's and Ph.D. programs can apply for 12 months of OPT at each educational level.[2] With the Science, Technology, Engineering, and Mathematics (STEM) OPT extension, CIS graduates can apply for an additional 24 months, for a total of up to three years. After this period, CIS graduates must leave the U.S., if they have not obtained a new work authorization, such as an H1-B visa.

[a] see https://bit.ly/3fLBhwj

analysis shows employees value working on interesting IT systems above most other employer attributes, such as pay and other benefits. They also note that recent graduates who arrived with newer skills in a high-demand job market would have more job options, and thus be more competitive hires.

How Can Companies Improve Their Retention and Hiring?

Instead of waiting for positions to become hard-to-fill and unfilled—leading to more lost productivity and reduced innovation—companies can take two strategic steps to attract and retain high-skilled employees: establish a dual career path for managerial and technical staff, and invest in employee education and training.

In 1988, Ginzburg and Baroudi identified the need to define a non-managerial technical career ladder to complement the traditional management track for promotions.[3] Over the last seven years, we observed that over 50% of graduates of the Master of Software Engineering (MSE) program at Carnegie Mellon University (CMU) have entered senior positions. These graduates enter either a managerial track, such as Project and Program Manager or Staff Software Engineer, or they enter a technical track, such as Senior Software Engineer and Software Development Engineer II and above. The technical track encourages career growth without shifting one's responsibilities to primarily cover management activities. In April 2020, we looked at career paths of alumni who graduated from 2015–2017 recording 44 alumni who ascended the technical ladder within their companies, and 10 who ascended by changing employers. Companies that allow employees to move between tracks afford employees more flexibility to pursue skill development within the company versus looking outward for open positions elsewhere. For example, we observed CMU MSE alumni working for a few years as technical leaders, building critical service infrastructure, who later chose to develop their personnel and project management skills by moving to a management track. We also see the converse, though less often: after having managed a team for some time, an alumnus/alumna chooses to re-enter a product or project team as a technical leader in a non-management role. In our discussions with alumni, these shifts were not viewed as career compromises, but as skill development opportunities.

Unfilled positions lead to increased churn as skilled employees jockey for better jobs at more attractive companies.

Another way to attract and retain employees, and to support internal career advancement, is to engage employees in on-the-job education by providing online learning benefits in addition to pay. One consequence of the pandemic has been large moves by colleges and universities toward online learning. Coursework in artificial intelligence and machine learning, DevOps, and software architecture are now online and taught by the world's experts in these topics. Online learning opportunities come in different sizes, from one- to three-course certificates to part-time degree programs that students complete over two to three years. In many cases, employees dedicate 10–12 hours a week to online courses, while applying this advanced knowledge to their workplace projects. To build such benefits, companies should consider several issues, including: how to allow for flexible schedules that accommodate online classes; how much to compensate employees for course tuition, recognizing these newly learned skills will transfer to their workplace performance; and whether a certificate or a degree program is the right incentive for recruitment and retention. Certificates may be attractive to senior employees, whereas a graduate degree could entice recent hires looking for more experience and seek additional formal education aimed at advancing their careers internally.

By emphasizing career paths that foster skill development, companies can move to keep employees more engaged with new opportunities that demonstrate their intellect and entrepreneurship. Moreover, providing corporate support for online software engineering education can re-energize a culture of innovation. Together, these investments can help companies reduce their exposure to the effects of this pandemic, while the upstream supply of IT labor replenishes to meet growing demand.

Conclusion

It is evident a labor shortage is expected to arrive soon and that it will disproportionately affect highly skilled workers. With an estimated reduction of 11,700 CIS graduates in 2021[5] and the BLS estimated 22% growth in software developer jobs,[8] there could be thousands of positions left unfilled or hard-to-fill. The added difficulty of hiring skilled employees from a smaller labor supply has the potential to cost companies through lost productivity and reduced innovation. The remaining question is how will companies creatively hire and retain high-skilled employees amidst all the coming turmoil? Companies can get ahead of the shortfall with two strategic steps: establishing a dual career path that allows employees to grow, and investing in employee education and training.

References

1. Bennett, J. and McGuinness, S. Assessing the impact of skill shortages on the productivity performance of high-tech firms in Northern Ireland. *Applied Economics*, 41 (2009), 727–737.
2. Department of Homeland Security. F-1 Optional Practical Training. *Study in the States Website* (2020); https://bit.ly/33RBvMH
3. Ginzberg, M.J. and Baroudi, J.J. MIS careers: A theoretical perspective. *Commun. ACM 31*, 5 (May 1988), 586–594.
4. Horbach, J. and Rammer, C. Labor shortage and innovation. ZEW—Centre for European Economic Research Discussion Paper No. 20-009. (2020).
5. Mitchell, T. Letter to Secretary Blinken and Secretary Mayorkas, on behalf of the American Council on Education (Mar. 18, 2021).
6. National Center for Educational Statistics. Digest of Educational Statistics. Tables 322.30, 323.30, and 324.25, (2020); https://bit.ly/3opVMCL
7. Tambe, P., Ye, X., and Cappeli, P. Paying to program? Engineering brand and high-tech wages. *Management Science 66*, 7 (2020), 3010–3028.
8. U.S. Bureau of Labor Statistics. Computer and Information technology occupations. *Occupational Outlook Handbook.* (2019); https://bit.ly/2S8dlec
9. U.S. Immigration and Customs Enforcement. 2018 Top 200 Employers for OPT and STEM OPT Students. SEVP Digital Library, (2018).

Travis Breaux (breaux@cs.cmu.edu) is Associate Professor and Director of the Software Engineering Masters Program at Carnegie Mellon University, Pittsburgh, PA, USA.

Jennifer Moritz (jmoritz@andrew.cmu.edu) is Alumni and Corporate Relations Manager at Carnegie Mellon University, Pittsburgh, PA, USA.

DOI:10.1145/3446796

Article development led by acmqueue
queue.acm.org

While powerful, frameworks are not for everyone.

BY CHRIS NOKLEBERG AND BRAD HAWKES

Application Frameworks

SHARED LIBRARIES ENCOURAGE code reuse, promote consistency across teams, and ultimately improve product velocity and quality. But application developers are still left to choose the right libraries, figure out how to correctly configure them, and wire everything together. By preinstalling and preconfiguring libraries, application frameworks provide a simplified developer experience and even greater consistency, albeit at the cost of some flexibility.

By owning the entire application life cycle, frameworks go beyond a mere collection of libraries. Guaranteed framework behavior can scale development—for example, by avoiding the need for in-depth security or privacy code reviews of every application. The cross-team and cross-language consistency provided by frameworks is also a necessary foundation for higher-level automation and smart systems.

This article offers an overview of the central aspects of frameworks, then dives deeper into their benefits, the trade-offs they entail, and the most important features we recommend implementing. Finally, we present a practical application of frameworks at Google: how developing a microservices platform allowed Google to break up its monolithic code base, and how frameworks enabled that change.

A framework is, in many ways, similar to a shared library and has similar benefits. For Google, two technical principles help to distinguish a framework from a library: inversion of control and extensibility. While seemingly modest, the many benefits of frameworks discussed in this article are mainly derived from these principles.

Inversion of control. In an application built from scratch, the engineer dictates the flow of the program—this is *normal control flow*. In a framework-based application, the framework controls the flow and will call into the

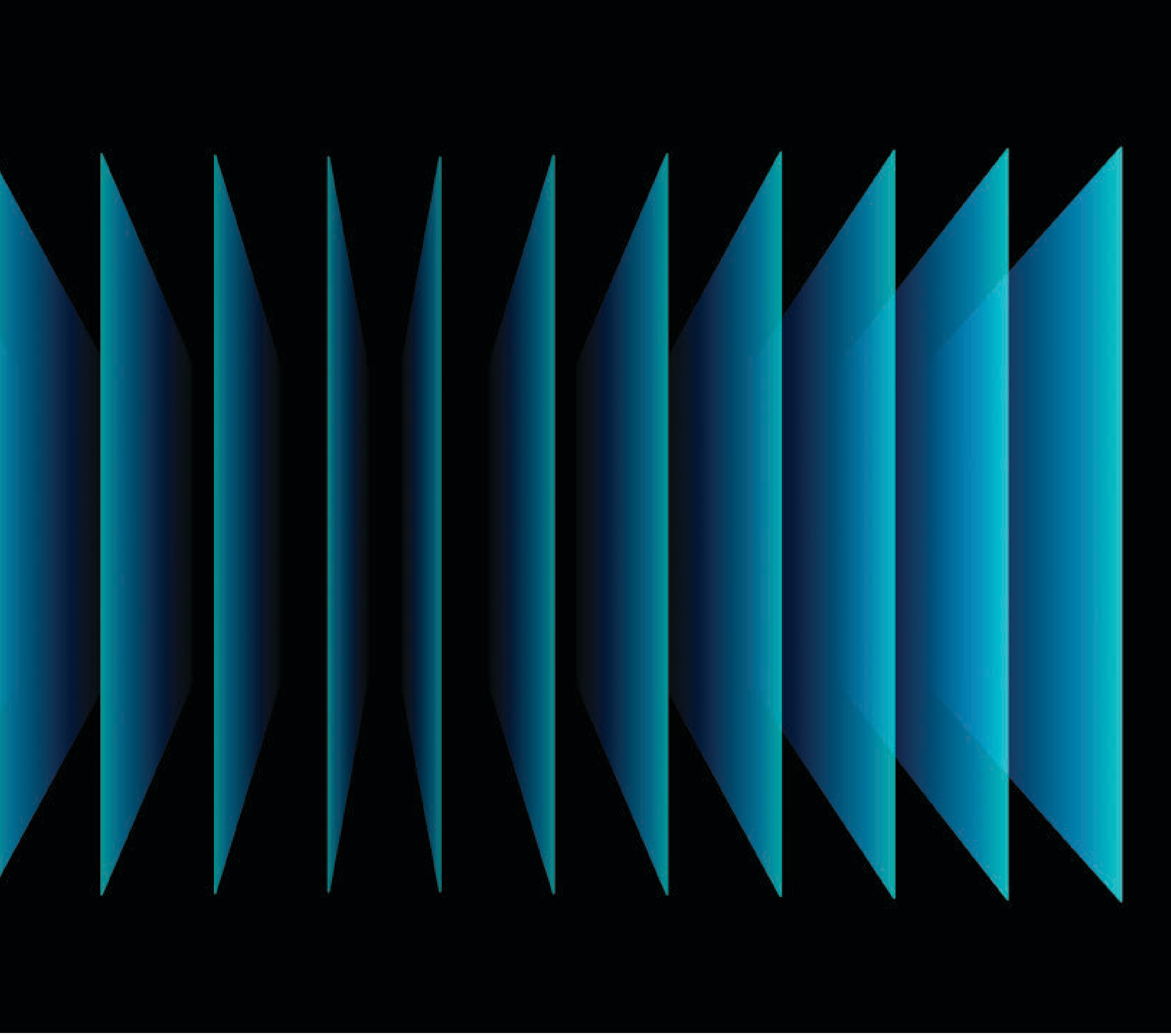


IMAGE BY KORKENG/SHUTTERSTOCK

user code—this is *inverted control flow*. Inverted control flow is sometimes referred to as the Hollywood principle: "Don't call us; we'll call you." The framework control flow is well defined and standard across all applications. Ideally, applications implement only the application-specific logic, while the framework handles all the other minutiae of building something like a microservice.

Extensibility is the second key differentiator from a library and goes hand in hand with inversion of control. Because a framework's control flow is owned by the framework, the only mechanism to alter its behavior is via the extension points it exposes. For example, a server framework might have an extension point that allows an application to run some code after every request. This behavior also implies that nonextensible parts of a framework are fixed and cannot be changed by applications.

Benefits of Frameworks

Frameworks have multiple benefits beyond the functionality that shared libraries provide, and they are advantageous to a variety of stakeholders in different ways.

Developers. Developers, who ultimately decide whether or not to use an available framework, are their most obvious beneficiaries. Primary developer benefits include increased productivity, simplicity, and conformity to best practices. Developers can write less code by leveraging built-in framework features, and the code they do write can be simpler because the framework handles boilerplate code. A framework provides a well-lit path for best practices by providing sensible defaults and eliminating pointless and time-consuming decision-making.

Production teams. In addition to improving developer productivity, frameworks benefit product teams by freeing up team resources that would otherwise be spent building redundant infrastructure. Product teams can then focus on what makes their product special.

Product teams also benefit when frameworks isolate them from changes in the underlying infrastructure. While not possible in all cases, the additional abstractions provided by a framework mean that some infrastructure migrations can be treated as implementation details, which are handled entirely by whomever maintains the framework.

Product launches at Google often require signoff from multiple teams. For example, a launch coordination engineer is responsible for reviewing launches for production safety and effectiveness, while an information security engineer will check an application's design for common security vulnerabilities. A framework can simplify the launch review process when the teams performing reviews are familiar with the frameworks and can rely on their behavioral guarantees. After launch, standardization will also make the system easier to manage.

The company. At the company level, common frameworks can increase developer mobility by reducing how long it takes for developers to get up to speed on a new application. If a company has a sufficiently large community of developers, investing in high-quality documentation and training programs is worthwhile; this in turn helps attract documentation and code contributions from the community itself. Widespread usage of a framework also means that a relatively small investment in improvements to the framework can have a large impact.

Over time, centralization in framework architectures can allow widescale reaction to changing landscapes. For example, if you rely on a consistent microservice/remote procedure call (RPC) framework and bandwidth becomes more expensive relative to CPU, then the framework defaults can centrally adjust compression parameters based on that cost trade-off.

Trade-Offs for Frameworks

While frameworks come with the multiple benefits just described, they also entail certain trade-offs.

Opinionated frameworks can hinder innovation. Frameworks often have to make choices about which types of technologies to support. While supporting every conceivable technology is not practical, there are clear benefits when frameworks are *opinionated*—that is, when they encourage the use of some technologies or design patterns over others.

> **While applications can take advantage of the framework extension points, most of the application code takes the form of actions, which embody the application-specific business logic.**

Opinionated frameworks can greatly simplify the job of developers approaching their new system with a blank slate. When developers have many ways to accomplish the same task, they can easily get buried in the details of decisions that have negligible impacts on the overall system. For these developers, accepting the preferred technologies of an opinionated framework allows them to focus on the business of building their system. Having a common and consistent preference also benefits the entire company, even if that answer is less than perfect.

Of course, you may have to deal with a long tail of applications and teams, and some product requirements or team preferences may not be well suited for existing frameworks. Framework maintainers are put in the position of deciding what is and isn't a best practice, and whether an unconventional use case is "valid" or not, which can be uncomfortable for everyone involved.

Another important consideration is that even if something is clearly a best practice today, technology evolves quickly, and there's a risk that frameworks will not keep pace with innovation. Experimenting with alternative application designs may be more expensive because developers need to either learn framework implementation details or rely on assistance from framework maintainers.

Universality can lead to unnecessary abstractions. Many framework benefits, such as common control surfaces (explained later), are realized only when a critical mass of applications use the same framework. Such a framework must be generic enough to support the vast majority of use cases, which in practical terms means having a rich request life cycle and all the extensibility hooks that any application would need. These requirements necessarily add some layers of indirection between the application and underlying libraries, which can add both cognitive and CPU overhead. For application developers, more layers in the software stack can complicate debugging.

Another potential drawback of frameworks is that they are yet one more thing engineers have to learn. Newly hired Googlers are frequently overwhelmed by the number of technologies they need to learn just to get a "Hello, world" example working. A full-featured framework might make the situation worse, not better.

Google has mitigated these issues somewhat by trying to make the core of each framework as simple and performant as possible and leaving other features as optional modules. Google also tries to provide framework-aware tools that can leverage the inherent structure of the frameworks to simplify debugging. Ultimately, however, frameworks have a cost that you must acknowledge, and you need to make sure that any given framework provides enough benefits to justify this cost. Different programming-language frameworks may also have differing sets of trade-offs, creating another decision point and cost/benefit scenario for developers.

Important Framework Features

As already discussed, inversion of control and extensibility are fundamental aspects of frameworks. Beyond those basic parameters, frameworks should account for several other features.

Standardized application life cycle. To reiterate, inversion of control means that a framework owns and standardizes an application's overall life cycle, but what benefit does this structure actually buy? The scenario of avoiding cascading failure provides one example.

Cascading failure is a well-known cause of system outages, including many at Google. It can occur when part of a distributed system fails, which then increases the probability that other parts will fail. For more information on the causes of cascading failures and how to avoid them, see the chapter on "Addressing Cascading Failures" in *Site Reliability Engineering*.[1]

Server frameworks at Google have a number of built-in protections against cascading failure. Two of the most important principles are:

- *Keep serving.* If a server can answer requests successfully, it should do so. If it can successfully serve some kinds of requests but not other kinds, it should continue running and answer the requests that it can serve.
- *Start up quickly.* The server should start up as quickly as possible. Faster startup means faster recovery from crashes. The server should avoid waiting serially for initializations involving RPCs to external systems to complete.

The Google production environment gives each server a configurable amount of time to become "healthy" (start responding to requests). If the time expires, the system assumes that an unrecoverable error occurred and terminates the server process.

There is one common antipattern that occurs naturally in the absence of a framework: A library creates its own RPC connection and then waits for that connection to be ready. As a server code base grows over time, you can end up with literally dozens of such libraries in the transitive dependencies. The result is server initialization code, which, if unrolled effectively, looks like Figure 1.

Under normal circumstances, this code will work fine, which is especially problematic because there is no indication of a lurking problem. That problem shows itself when one of the associated back-end services slows down or goes down altogether—now the primary server's startup is delayed. If the startup is sufficiently delayed, it will be killed before it ever gets a chance to start handling requests, which can contribute to a cascading failure.

One possible improvement is to create the RPC stubs first, as in Figure 2, and then wait for them all in parallel. In this scenario, you need to wait only for the *max* of the stub initialization times rather than the *sum*.

While still not perfect, even this limited refactoring demonstrates that you need *some* coordination between the libraries creating the RPC stubs—they must hand off the responsibility of waiting for the stub to something outside of the library. In Google's case, that responsibility is owned by the server framework, which also has the following features:

- Waiting for all stubs to be ready *in parallel*, by polling readiness periodically (< 1 sec). Once a configurable timeout has elapsed, the server can continue with initialization *even if not all back ends are ready*.
- Emitting human- and machine-readable logging for debugging and integration with standard monitoring and alerting systems.
- Plugging in arbitrary resources, not just RPC stubs, through a generic mechanism. Technically, only a function returning a Boolean (for "Am I ready?") and a name is necessary for logging purposes. These hooks are typically used by the common libraries that deal with resources (for example, a file API); application developers often get

Figure 1. Server initialization code.

```
// In library 1
// newStub creates a stub which will asynchronously connect to a backend.
FooService.Stub fooStub = FooService.newStub(...);
// waitUntilReady blocks until the stub is successfully connected.
waitUntilReady(fooStub);
// In library 2
BarService.Stub barStub = BarService.newStub(...);
waitUntilReady(barStub);
// In library 3
BazService.Stub bazStub = BazService.newStub(...);
waitUntilReady(bazStub);
```

Figure 2. RPC stubs.

```
// Make sure to call newStub for all stubs first, before we wait for any
// of them.
FooService.Stub fooStub = FooService.newStub(...); // In library 1
BarService.Stub barStub = BarService.newStub(...); // In library 2
BazService.Stub bazStub = BazService.newStub(...); // In library 3
// Wait, now that we have started the async connection process for all
// stubs. The order in which we wait for the stubs is irrelevant.
waitUntilReady(fooStub);
waitUntilReady(barStub);
waitUntilReady(bazStub);
```

the behavior automatically just by using the library.

- Providing a centralized way to configure certain back ends as "critical," which alters their startup and runtime behavior.

These features would (rightly) be considered overkill for any individual library, but implementing them makes sense if you can do so in a central place from which all back end-using libraries can benefit. Just as shared libraries are a way to share code among applications, in this case the framework is a way for libraries themselves to share functionality.

Site reliability engineers (SREs) are much happier to support framework-based servers because of features such as these, and they often encourage their developer counterparts to choose framework-based solutions. Frameworks provide a baseline level of production regularity that is difficult—if not impossible—to achieve when just gluing together a bunch of disconnected libraries.

Standardized request life cycle. While details vary depending on the type of application, many frameworks support additional life cycles beyond an overall application life cycle. For Google server frameworks, the most important unit of work is a request. Following a similar inversion-of-control model, the goal of the request life cycle is to divide the responsibilities for different aspects of the request into separate extensible pieces of code. This allows application developers to concentrate on writing the actual business logic that makes their application unique.

Here's an example of one such real-world framework and its component pieces, as illustrated in Figure 3:

- Processors—intercept incoming and outgoing payloads. Mostly used for logging but have some capabilities for short-circuiting a request (for example, enforcing invariants across an entire application).
- Action—application business logic that takes the request and returns a response object, possibly with side effects.
- Exception handler—converts an uncaught exception into a response object.
- Response handler—serializes a response object to the client.

While applications can take advantage of the framework extension points, most of the application code takes the form of actions, which embody the application-specific business logic.

This separation of concerns has been helpful in the realm of Web security, for example. Google develops many Web applications, so it has a strong desire to guard against all of the various Web security vulnerabilities, such as XSS (cross-site scripting). XSS vulnerabilities are often caused by application code returning a string response containing insufficiently sanitized or escaped data. The traditional approach to fixing these bugs is simply to add in the missing escaping data and hope that tests and code reviews will prevent similar problems in the future (spoiler: they will not).

Fundamentally, this approach won't work because the underlying APIs that the applications are coding against are inherently prone to bugs, like XSS, because they accept strings or similarly unstructured/untyped data. For example, the Java Servlet API gives applications a raw writer, to which you can pass arbitrary characters. This approach puts too much of a burden on developers to do the right thing; instead, the security team at Google has focused on designing inherently safe APIs, such as:

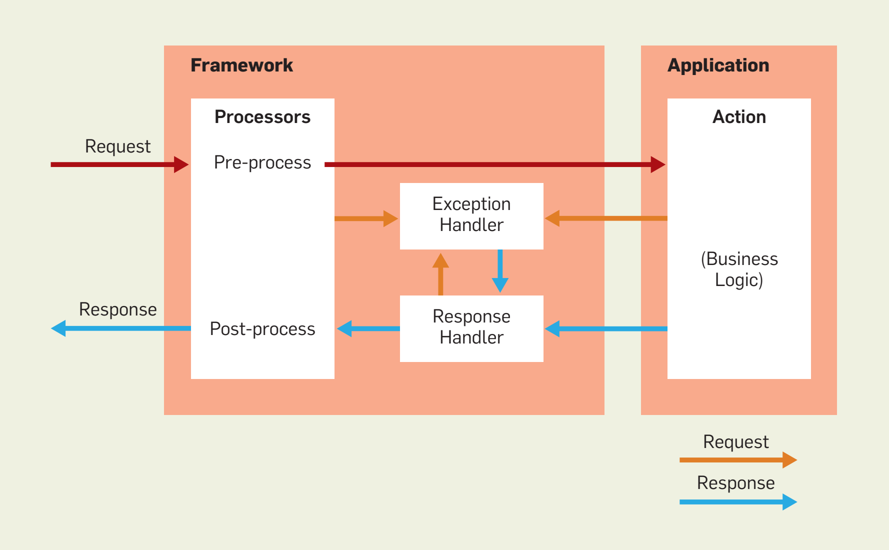
- HTML template systems with contextual-aware escaping.
- SQL injection-resistant database APIs.
- "Safe HTML" wrapper types that carry contracts stipulating that their value is safe to use in various contexts.

The request life cycle of Google's server frameworks complements the use of these APIs because the application code never deals with raw strings or bytes. Instead, the code returns high-level response objects with types such as SafeHtmlResponse that can be constructed only in ways that are guaranteed to be well formed. Turning those response objects into bytes on the wire is the responsibility of a *response handler*, which is typically a built-in part of the framework. Google sometimes needs custom response handlers, but all usages must be reviewed by the security team—a requirement that is enforced at the build level.

The net impact is that Google has reduced the number of XSS vulnerabilities to virtually zero in applications using these frameworks. As you can imagine, Google's security team strongly encourages the use of frameworks and has made many framework contributions to improve the security story for all framework users. A standard framework-based server can effectively skip many of the security or privacy reviews that a bespoke server would require to launch, since the framework is trusted to guarantee certain behaviors.

Of course, the benefits of a structured and extensible request life cycle go well beyond separating business logic from response serialization. The most basic benefit is that it keeps each component small and easy to reason about, which helps long-term code health. Other infrastructure teams within Google can easily extend frame-

Figure 3. An example framework and its component pieces.



work functionality without working directly with the framework team. Finally, applications can introduce their own cross-cutting features without touching each action. In some cases, these features are domain-specific, but other features end up being generally applicable and are eventually "upstreamed" into the framework itself.

Common control surfaces. What we call *control surfaces* include all of the non-application-specific inputs and outputs of a binary when viewed as a black box. These include operational controls, monitoring, logging, and configuration.

Uniformity across servers simplifies troubleshooting. Regardless of which server they're troubleshooting, developers and SREs all know what information is available and where to look for it. If something about a server needs to be tweaked, everyone knows which knobs are available and how to change them.

Beyond making it easier for humans to operate servers, having common control surfaces across servers also makes shared automation viable. For example, if all servers export errors in a standard way, changing the release pipeline to perform automatic canarying becomes possible: You can first roll out a new binary to a few servers and look for a spike in errors before performing a wider rollout. You can read more about the benefits of a common control surface from the SRE's perspective in the chapter on the "Evolving SRE Engagement Model."[1]

Frameworks provide a great opportunity to enforce a level of uniformity across application control surfaces. While, generally, only a few people care about the exact composition of control surfaces, there is tremendous value to the company in having a single, consistent answer. Consistency means that you can easily share and scale automation across multiple binaries. By simplifying integration with the surrounding ecosystem, you can reap the benefits of *having* a standard many times over, independent of the merits of the standard itself.

One challenge of implementing a common control surface is that framework maintainers are often the first to discover inconsistencies across programming-language libraries. For example, all languages had an existing notion of a command-line argument whose value was a duration of time. On the positive side, the syntax was somewhat compatible among languages, at least for the most basic examples, such as "1h30m." Once we dug into the details, however, a different picture emerged, as shown in Figure 4.

These days, library owners have a greater awareness of the value of cross-language consistency and the need to take such consistency into consideration. On the framework side, Google also uses test harnesses to run the same suite of tests against servers written in each programming language to ensure consistency going forward.

Modularity. For better or worse, there is no central software engineering authority at Google. Although most developers work against a single code repository, engineering practices still vary significantly among teams. The choice of technologies for any given project typically rests with the tech lead of the project, with few top-down mandates. Understandably, people tend to choose technologies with which they have prior experience. As a result, in order for a new technology to gain significant adoption, it must have either an obvious value or a low barrier to entry; typically, it must have both.

For Google's server frameworks, core life-cycle management and request dispatching are the only strictly required features. All other functionality is bundled into optional, independent "modules" that implement their functionality using the various life-cycle hooks exposed by the framework, as discussed earlier. Application developers can pick and choose which modules to add to their server, and in many cases even major features can be added via a one-liner:

```
install(new LoadSheddingModule());
```

The actual list of available standard modules numbers in the hundreds, including features such as authentication, experiments, and logging.

The ability to incrementally add framework features to a server was a big factor in framework adoption at Google. It allowed for "Hello, world" examples and prototype servers to be small and easy to understand, while still making it simple to scale up to a more full-featured server when appropriate.

The independence of the modules also allows for easy substitution of an application-specific module for a standard framework module if you have special requirements. Because standard framework modules use the same extensibility APIs as application-specific modules, upstreaming a useful feature into the framework is usually a trivial matter of moving code. This allows a framework to be an ever-growing collection of best practices, once they have proven their real-world value.

The high degree of encapsulation exhibited here means that framework maintainers can radically change the implementation of a module without touching any application code. This is especially useful when a back-end system is deprecated or requires API changes (which is distressingly common). Google frameworks have insulated many application developers from needing to perform complex or costly migrations, and for many teams this is one of the most compelling benefits of frameworks today.

One role of a framework maintainer is to ensure that modules correctly collaborate with each other. Maintainers can also select default module lists or provide recommendations and constraints about which modules should be used for different situations. One challenge is striking the right balance with regard to granularity:

Figure 4. Inconsistencies across programming language libraries.

	C++	Java
Days ("d")		✓
Hours ("h")	✓	✓
Minutes ("m")	✓	✓
Seconds ("s")	✓	✓
Milliseconds ("ms")	✓	
Microseconds ("us")	✓	
Nanoseconds ("ns")	✓	
Units out of order	✓	
Repeated units	✓	
Fractional values	✓	
Unitless value		✓
Mixed case		✓

While developers tend to prefer fine-grained modules for flexibility, it's harder for framework maintainers to ensure that all combinations will work well together.

Microservices. The standardization provided by widespread use of frameworks leads to opportunities for higher-level tools and automation. This allowed Google to create a microservices platform and break up the monolithic servers.

Before microservices: The monolith. The existence of shared libraries and frameworks has greatly simplified the actual act of writing production-quality code within Google. Writing code, however, is just one part of deploying an application at Google. Other critical ingredients include integration testing, launch reviews for aspects such as security and privacy, acquiring production resources, performing releases, collecting and saving logs, experimentation, and debugging and resolving outages.

Historically, handling all of these items was an expensive process that all server owners had to complete, regardless of server size. As a result, instead of deploying new servers, smaller teams adding a new service would look for an *existing* server to which they could add their code. This way, the team could just focus on writing their business logic and get everything else "for free." Of course, once enough teams took this approach, it became clear that piggybacking onto existing server functionality was not actually free. This incentive structure resulted in a tragedy of the commons many times at Google: Well-supported servers continued to grow and grow until they became huge and unmaintainable monoliths.

Monoliths have many negative consequences. On the developer productivity front, you must deal with slow builds, slow server startup, and a high likelihood that your presubmit tests will break when you try to submit your change. For example, one important Google Search-related C++ binary became so large that it was impossible even to link, given technical limits at the time (12GB RAM).

When it comes to releases, it's difficult to push them to monoliths on schedule. As a monolith grows, so too does the number of contributing developers, which results in more blocking bugs. A delayed release may make achieving the next release even more difficult, creating a vicious cycle.

In production, monoliths create a dangerous shared fate between ostensibly unrelated services, as well as a greater chance of bugs caused by unexpected interactions. Scaling services independently of one another is impossible, which makes resource provisioning more difficult.

Moving away from a server-oriented world. Although it eventually became clear that the monolith situation at Google was unsustainable, there was no good alternative. Simply mandating that people stop adding to a monolith would have had equally bad consequences. Instead, Google needed to eliminate the toil of productionizing and running a new server. That would allow the decision of which *services* should make up a *server* to be based purely on production reasons, rather than developer convenience, shown in Figure 5.

Working backward from the goal that developers should just focus on the business logic of their application-specific service, and that *everything else* should be automated as much as possible, some requirements eventually became clear:

- Developers should be declaring and implementing service APIs, not writing main methods: orchestrating how a binary is actually run is the role of the microservice platform.
- All of the metadata needed for automation, including production configuration, should live in a declarative format alongside the code for the service.
- Resources and dependencies between services should be explicit and declarative. Ideally, you should be able to visualize the entire production topology just from looking at the metadata for the universe of services.
- Services should be isolated from each other, so that arbitrary services can be co-assembled into a server.

Figure 5. Breaking up a monolithic server into smaller servers.

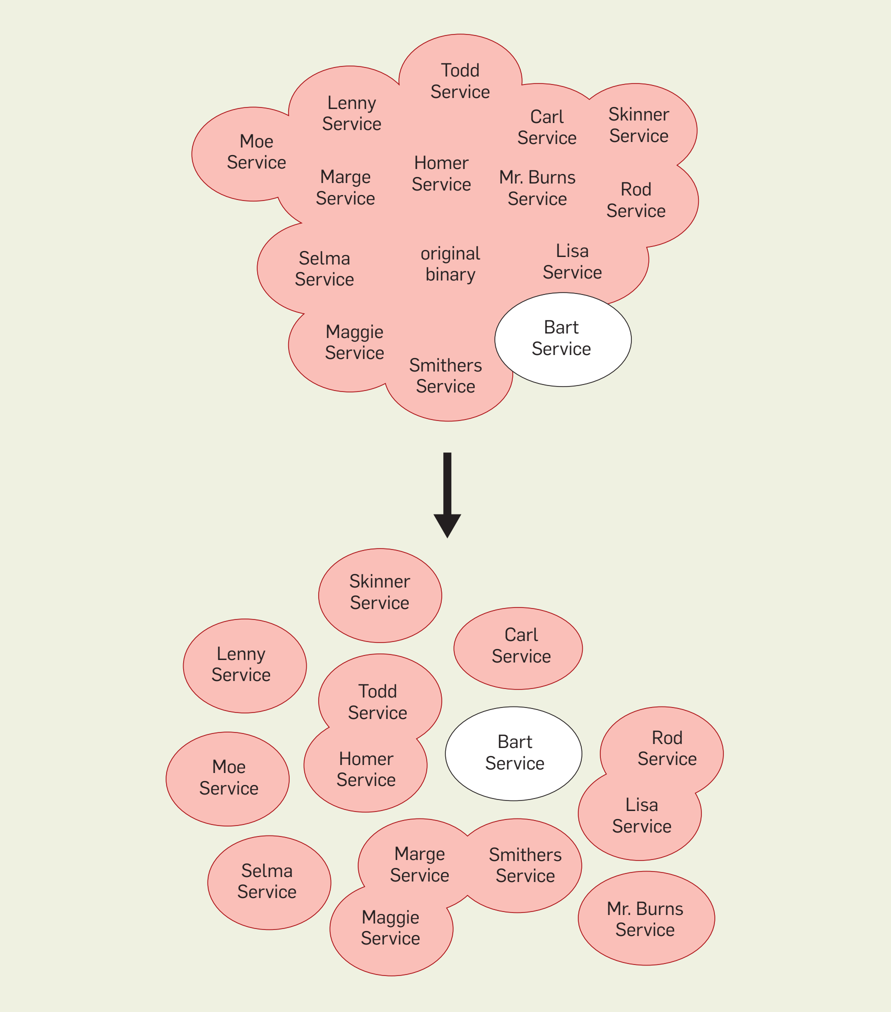
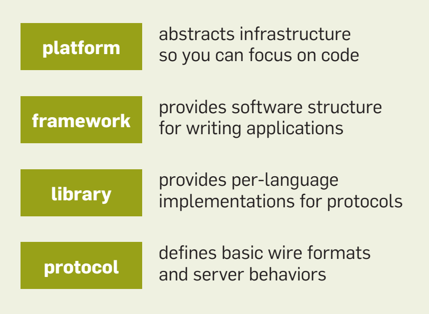


Figure 6. Development stack of Google's microservices platform.


platform
abstracts infrastructure so you can focus on code
framework
provides software structure for writing applications
library
provides per-language implementations for protocols
protocol
defines basic wire formats and server behaviors

Among other requirements, this means avoiding global state and side effects.

When these requirements are satisfied, virtually all formerly manual processes can be automated. For example, testing infrastructure can use the metadata to wire up a portion of the service-dependency graph when running integration tests.

A microservice platform using these principles was developed at Google, initially for the purpose of breaking up a particularly large monolithic server that had seen rapid growth as a result of intense feature development. Once the platform proved beneficial, it was organically adopted by other teams within Google and was eventually spun out into a separate, officially supported project.

Today the platform is the de facto standard for new server development, in part because it appeals to both small teams and large organizations. Because of the high level of automation, small teams can now easily turn up a Google-quality production service in a matter of days, whereas before a turnup following best practices might have taken months. For large organizations, the consistency across teams reduces support costs, and the shared platform means that staffing org-specific infrastructure teams is often unnecessary.

Another benefit of moving to microservices has been encouraging developers to think more about the proper division of work among services, which has led to more rational system architectures. Using technologies such as gRPC and protocol buffers as the boundary between separate systems forces you to consider the APIs in a way that doesn't necessarily happen when you're only using function calls in the same process. RPC systems are also language agnostic, so each microservice owner can independently decide which language to use.

One remaining challenge, and a ripe area for future work, is providing higher-level tools to manage an ever-growing number of microservices. For example, monitoring consoles that were written in the previous era may have assumed a relatively small number of unique binaries, and this will require a new user interface to accommodate the much greater number of binaries that arises when people fully adopt microservices.

Relationship with frameworks. Frameworks are a critical component of making Google's microservices platform work for a few reasons:

- The inversion of control inherent in the framework's life cycle naturally lends itself to a model where application developers just hand off their service implementation to the platform.
- Common control services (across both servers and languages) are required for many platform features, including release management, monitoring, and logging.
- Modularity means that both the platform and application code can provide independent modules, which when combined together, work in a sane way to form a complete server.

Figure 6 shows the full development stack for Google's full microservices platform.

As discussed before, frameworks can offer a greater level of encapsulation than libraries, which simplifies writing applications and provides isolation from underlying library churn. In a similar way, the microservices platform goes beyond just code to encapsulate other artifacts, such as production configuration. This allows for a corresponding higher level of simplification and isolation from churn. For example, platform maintainers can (if necessary) automatically apply an emergency code fix or configuration change that rebuilds all affected binaries and pushes them to production in a uniform way—previously impossible.

Using a microservices platform, however, does present some challenges. One of the biggest of these is that enforcing all of the invariants required to make the microservices platform function properly can be onerous and may even affect how applications are coded. To provide one example, Google's Java servers share certain thread pools. Combined with the requirement that all services must be isolated from each other, this implies that a blocking thread-per-request model cannot be allowed—it would be too easy for a blocking service to use up all the threads and starve another service. For that reason, servers are mandated to be async only, a solution that not all teams are happy with.

Another challenge is that adding more hops between microservices may add latency to the overall request. In some cases, this latency can be mitigated by architectural improvements that happen as part of a microservices rewrite. For its microservices platform, Google has also ensured that requests between services that happen to be colocated in the same server use an optimized in-process transport.

Conclusion

While frameworks can be a powerful tool, they have some disadvantages and may not make sense for all organizations. Framework maintainers need to provide standardization and well-defined behavior while not being overly prescriptive. When frameworks strike the right balance, however, they can offer large developer productivity gains. The consistency provided by widespread use of frameworks is a boon for other teams, such as SRE and security that have a vested interest in the quality of applications. Additionally, the structure of frameworks provides a foundation for building higher-level abstractions, such as microservices platforms, which unlock new opportunities for system architecture and automation. At Google, such frameworks and platforms have seen broad organic adoption and have had a significant positive impact.

Reference

1. Beyer, B. et al. *Site Reliability Engineering.* O'Reilly Media, Inc. (Apr. 2016).

Brad Hawkes is a senior software engineer working on core infrastructure at Google. He works on the server framework on the Java Virtual Machine, which is used in thousands of servers across Google.

Chris Nokleberg is a principal software engineer and tech lead of server frameworks at Google. He started developing frameworks as a tech lead on Google Docs almost 10 years ago; he is now helping large teams adopt Google's microservices platform and standardizing developer best practices across the company.

DOI:10.1145/3442518

Article development led by acmqueue
queue.acm.org

A discussion with Theo Schlossnagle, Justin Sheehy, and Chris McCubbin.

Always-on Time-Series Database:

Keeping Up Where There's No Way to Catch Up

IN ALL LIKELIHOOD, you have never given so much as a thought to what it might take to produce your own database. And you will probably never find yourself in a situation where you need to do anything of the sort.

But, if only as a thought exercise, consider this for a moment: What if, as a core business requirement, you found you needed to provide for the capture of data from disconnected operations, such that updates might be made by different parties at the same time—or in overlapping time—without conflicts? And what if your service called for you to receive massive volumes of data almost continuously throughout the day, such that you couldn't really afford to interrupt data ingest at any point for fear of finding yourself so far behind present state that there would be almost no way to catch up? Given all that, are there any commercially available databases out there you could use to meet those requirements?

Right. So, where would that leave you? And what would you do then? We wanted to explore these questions with **Theo Schlossnagle**, who did, in fact, build his own time-series database. As the founder and CTO of Circonus, an organization that performs telemetry analysis on an already large and exponentially growing number of IoT (Internet of Things) devices, Schlossnagle had good reason to make that investment.

Justin Sheehy, the chief architect of global performance and operations for Akamai, asks Schlossnagle about the thinking behind that effort and some of the key decisions made in the course of building the database, as well as what has been learned along the way. On behalf of ACM, **Chris McCubbin**, a senior applied scientist with Amazon Web Services, contributes to the discussion.

JUSTIN SHEEHY: As someone who once made the dubious decision to write my own database, I know it *can* prove to be the right thing to do, but—for most companies—I don't think it turns out that way. This isn't just a business question, but one that also has some interesting engineering dimensions to it. So, Theo, why did you feel the need to write your own time-series database?

THEO SCHLOSSNAGLE: There were a number of reasons. For one, almost all the data that flows into our telemetry-analysis platform comes in the form of numbers over time. We've witnessed more than exponential growth in the volume and breadth of that data. In fact, by our estimate, we've seen an increase by a factor of about $1\text{x}10^{12}$ over the past decade. Obviously, compute platforms haven't kept pace with that. Nor have storage costs dropped by a factor of $1\text{x}10^{12}$. Which is to say the rate of data growth we've experienced has been way out of line with the economic

IMAGE BY TIMOFEEV VLADIMIR

realities of what it takes to store and analyze all that data.

So, the leading reason we decided to create our own database had to do with simple economics. Basically, in the end, you can work around any problem but money. It seemed that by restricting the problem space, we could have a cheaper, faster solution that would end up being more maintainable over time.

SHEEHY: Did you consider any open source databases? If so, did you find any that seemed almost adequate for your purposes, only to reject them for some interesting reason? Also, I'm curious whether you came upon any innovations while looking around that you found intriguing … or, for that matter, anything you really wanted to avoid?

SCHLOSSNAGLE: I've been very influenced by DynamoDB and Cassandra and some other consistent hashing databases like Riak. As inspiring as I've found those designs to be, I've also become very frustrated by how their approach to consistent hashing tends to limit what you can do with constrained datasets.

What we wanted was a topology that looked similar to consistent hashing databases like DynamoDB or Riak or Cassandra, but we also wanted to make some minor adjustments, and we wanted all of the data types to be CRDTs [conflict-free replicated data types]. We ended up building a CRDT-exclusive database. That radically changes what is possible, specifically around how you make progress writing to the database.

There are a few other nuances. For one thing, most consistent hashing systems use the concept of vBuckets in the rings where, say, you have 15 hosts and 64 virtual buckets that data falls into, with the hosts ultimately negotiating to determine which of them owns which bucket.

With our system, we wanted to remove that sort of gross granularity. So, we actually use SHA-256 as our hashing scheme, and we employ 2^{256} vBuckets. As a consequence, where each data point falls is driven by where the node and the ring fall instead of by vBucket ownership.

This allows us to break into what we call a "two-sided ring." In a typical consistent hashing ring, you have a set of hosts, and each of those has multiple representations all around the ring. What we did instead was to allow the ring to be split in half, with the first 180 degrees of the ring—the first pi of the ring—containing half of the nodes, and the other pi containing the remaining nodes.

Then, to assign data to the nodes, we used what we call a "skip walk." Basically, that flips pi radians back and forth across the ring, thus guaranteeing that you alternate from one side of

THEO SCHLOSSNAGLE

There is an incredibly strong impetus to build time-series databases for always-on data ingest since otherwise—in the event of disruptions of service or catastrophic failures—you will find, upon resumption of service, your state will be so far behind present that there will simply be no way to catch up.

the ring to the other, which turns out to be pretty advantageous when it comes to putting half of your ring onto one AZ [availability zone] and the other half onto another—or into one region or another, or into one cloud or another.

An interesting aspect of using CRDTs—rather than requiring consensus to make progress—is that it lets you, say, put half of your ring on an oil rig and the other half in the Azure cloud, and then have everything synchronize correctly whenever the VSAT [very small aperture terminal] link is working as it should. That way you can have all the same features and functionality and guarantees even when they're disconnected.

SHEEHY: I've also done some work with CRDTs and find them interesting in that you have data types that can be updated by multiple parties—possibly on multiple computers, at the same time or in overlapping time—without conflicts. Since updates are automatically resolved, you don't need isolation in the traditional database sense to ensure that only one party at a time is able to perform a transaction on a given data structure. In fact, this can happen arbitrarily, and it still will all sort out.

Also, there are different ways to solve this, whether through commutativity or convergent operations—bearing in mind that a lot of research has been done on these over the past 10-plus years. So, you can have some of the benefits of consensus without forbidding more than one party to act on the same data at any one time. Which is why I consider this to be an exciting area of research and implementation.

CHRIS MCCUBBIN: How does this apply to the data types and operations that are best suited for time-series databases? Has that even been the focus of CRDT research to date?

SCHLOSSNAGLE: Much of the CRDT research so far has focused on disconnected operations, and, certainly, that isn't the first thing that comes to mind when you think about time-series databases. But the real advantage of the CRDT approach is that, if you can limit your entire operation set to CRDTs, you can forego consensus algorithms such as Paxos, Multi-Paxos, Fast Paxos, Raft, Extended Virtual Synchrony, and anything else along those lines. Which is not to disparage any of those algorithms. It's just that they come with a lot of unnecessary baggage once the ability to make progress without quorum can be realized.

There are a couple of reasons why this makes CRDTs incredibly appealing for time-series databases. One is that most time-series databases—especially those that work at the volume ours does—take data from machines, not people. The networks that connect those machines are generally wide and what I'd describe as "always on and operational." This means that, if you have any sort of interruption of service on the ingest side of your database, every moment you're down is a moment where it's going to become all the more difficult to recover state.

So, if I have an outage where I lose quorum in my database for an hour, it's not like I'll be able just to pick up right away once service resumes. First, I'll need to process the last hour of data, since the burden on the ingest side of the database continues to accumulate over time, regardless of the system's availability. Which is to say, there's an incredibly strong impetus to build time-series databases for always-on data ingest since otherwise—in the event of disruptions of service or catastrophic failures—you will find, upon resumption of service, your state will be so far behind present that there will simply be no way to catch up.

SHEEHY: It sounds like, for thoughtful reasons, you traded one very hard problem for another—by which I mean you got out from under the issues related to consensus algorithms. I've learned, however, that many so-called CRDT implementations don't actually live up to that billing. I'm curious about how you got to where you could feel confident your data-structure implementations truly qualified as CRDTs.

SCHLOSSNAGLE: It's certainly the case that a lot of CRDTs are really complicated, especially in those instances where they represent some sort of complex interrelated state. A classic example would be the CRDTs used for document editing, string interjection, and that sort of thing. But the vast majority of machine-generated data is of the write-once, delete-never, update-never, append-only variety. That's the type of data yielded by the idempotent transactions that occur when a device measures what something looked like at one particular point in time. It's this element of idempotency in machine-generated data that really lends itself to the use of simplistic CRDTs.

In our case, conflict resolution is the primary goal since, for time-series data, there can be only one measurement that corresponds to a particular timestamp from one specific sensor. To make sure of that, we use a pretty simplistic architecture to ensure that the largest absolute value for any measurement will win. If, for some reason, a device should supply us with two samples for the same point in time, our system will converge on the largest absolute value. We also have a generational counter that we use out of band. This is all just to provide for simplistic conflict resolution.

With all that said, in the course of a year, when we might have a million trillion samples coming in, we'll generally end up with zero instances where conflict resolution is required simply because that's not the sort of data machines generate.

As might be expected, Schlossnagle and his team didn't start from scratch when it came to designing and developing their time-series database. Instead, they looked to see what frameworks might be used and which libraries could be borrowed from.

Up front, they determined what was most crucial and which trade-offs they would be willing to make. For example, they knew they would need to treat forward and backward compatibility as a fundamental requirement since they couldn't afford to have anything disrupt data ingestion.

They also understood that the actual matter of writing data to raw devices would be one of the hardest things to get right. So, they designed their database such that there are only a few places where it actually writes to disk itself. Instead, a number of existing embedded database technologies are leveraged, with optimized paths within the system having been engineered to take advantage of each of those database technologies while also working around their respective weaknesses.

SHEEHY: It's clear you viewed CRDTs as a way to address one of your foremost design constraints: the need to provide for always-on data ingest. Were there any other constraints on the system that impacted your design?

SCHLOSSNAGLE: We've learned firsthand that whenever you have a system that runs across multiple clusters, you don't want to have certain nodes that are more critical than all the others since that can lead to operational problems. A classic problem here is something like NameNodes in Hadoop's infrastructure or, basically, any sort of special metadata node that needs to be more available than all the other nodes. That's something we definitely wanted to avoid if only to eliminate single points of failure. That very much informed our design.

Also, while we focused on the economies of scale since we were taking in some really high-volume telemetry data, one challenge we didn't think about early enough, I'd say, was how we might later manage to find things among all that data. That is, if you have a million trillion samples coming into your system over the course of a year, how are you going to find something in all that? How are you even going to be able to navigate all that data?

For example, in the IoT realm, if you're taking in measurement data from 10 billion sensors, how are you then going to isolate the data that was obtained from certain sensors based on a metadata search across a distributed system? I don't think that would normally pose a particularly hard computing challenge, but because it's not something we designed for up front, it certainly led to a lot of pain and suffering during our implementation.

SHEEHY: You rarely hear people talk about how they needed to change their approach based on what they learned that was at odds with their initial assumptions or exposed something that hadn't been provided for—like the very thing you're talking about here. If I understand you correctly, you initially didn't include exploratory querying for some data you considered to be of lesser importance, only to realize later it actually was significant.

SCHLOSSNAGLE: That's a fair characterization. To be more specific, let's just say that, with any telemetry-based time-series system, you're going to have a string of measurements attached to some sensor. Say you're measuring CPU usage on a server. For that case, you would have some unique identifier for a CPU linked to some number of measurements—whether taken every second, every minute, or every tenth of a second—that express how that CPU is actually being used.

We found you can have as many as 100 million or even a billion of these strings within a single system, meaning it can be very difficult to find one particular string and explore it. As a standalone computer science problem, that wouldn't be all that difficult to solve.

What complicates matters is that the metadata around these strings keeps changing over time. A great example has to do with container technology. Let's say you attach your CPU-usage data to a container you're running under Kubernetes, and then you decide to do 40 launches a day over the course of a year. This means that, where you previously had 100 million streams, you now have 14,600 times as many [365 days x 40 launches]—so, you've got a real cardinality challenge on your hands.

SHEEHY: Given your always-on, always-ingesting system and the obvious need to protect all the data you're storing for your customers, I'm curious about how you deal with version upgrades. When you're planning to change some deeply ingrained design element in your system, it's not like you can count on a clean restart. I've dealt with this concern myself a few times, so I know how much it can affect your engineering choices.

SCHLOSSNAGLE: I think you'll find a lot of parallels throughout database computing in general. In our case, because we know the challenges of perfect forward compatibility, we try to make sure each upgrade between versions is just as seamless as possible so we don't end up needing to rebuild the whole system. But the even more important concern is that we really can't afford to have any disruption of service on ingestion, which means we need to treat forward and backward compatibility as an absolutely fundamental requirement. Of course, you could say much the same thing about any database that stores a lot of data and has a wide user base.

Postgres is a great example of a database that has really struggled with this challenge in the sense that, whenever you make an on-disk table format change for a 30TB database, you can count on some sort of prolonged outage unless you can perform some serious magic through replication and that sort of thing. Still, I think Postgres over the past few years has become much better at this as it has come to realize just how large databases are getting to be and how long these outages can last whenever people make changes that break forward compatibility.

SHEEHY: And it's not only storage you need to worry about with forward and backward compatibility. These same concerns apply to all the nodes you're running in your system since you can't have all of them change versions at exactly the same moment. Also, in my experience, the compatibility issue proves to be quite a bit more difficult from an engineering perspective, since then you're more or less living it all the time—not just when you need to upgrade your storage.

SCHLOSSNAGLE: This is definitely about more than just on-disk formats and capabilities. Protocol compatibility also needs to be taken into account, specifically with regard to replication and querying and that sort of thing. There are some frameworks such as Google Protobuf [Protocol Buffers] and gRPC that include some advances that provide for this. We use FlatBuffers, which, ironically, also happens to come from Google. All of these frameworks help future-proof for compatibility. So, you can serialize data and add new fields, but people who have been around for a long time will still be able to read the data without knowing a thing about all those new fields. And, just so, new people will be able to read the old data even though it's missing those fields. This definitely helps ease many of the implementation concerns. But the design concerns remain, so I think you need to approach it that way. In fact, I'd love to learn about any best practices emerging now in industry that address how to design for this forward-compatibility challenge.

Another reservation I have is that these frameworks, for all the advantages they offer, tend to be very language specific. If there's a tool you can use to good effect in, say, Erlang, it's unlikely to be of much help to anyone who works in Go, just as a tool written for Rust isn't necessarily going to do much for people who have to work in a C environment. When you consider that there are more than just a handful of commonly used production-system languages out there, it becomes easy to see how this can make things pretty tricky.

SHEEHY: I completely agree, but let's get back to what in particular drove the development of *your* time-series database. I'm especially interested in learning about any preexisting things you managed to leverage. We already talked a little about FlatBuffers, but I imagine there were some other pieces of software, or even hardware, you were able to take advantage of.

SCHLOSSNAGLE: Let me first say that FlatBuffers provides an especially good example in the sense that it provides for the reuse of a ready-made solution for serialization and de-serialization, which has got to be one of the least glamorous tasks for any engineer. What's even more important, though, is that, by providing a nice toolkit around all that, FlatBuffers also delivers important backward-compatibility and endian guarantees for the network, which are invaluable. This also applies to Protobuf and Cap'n Proto.

I will say, though, that we're not equally enamored of every one of these frameworks. In particular, Avro and Thrift, with their blatant insistence on ignoring unsigned types, have proved to be quite difficult to use in practice. We ended up deciding to focus only on those serialization/de-serialization solutions that actually understand common systems types.

Beyond this, we rely on a large number of libraries. In fact, most of the data structures we use come from open source libraries. Concurrency Kit is a great example that provides a set of basic data structures and primitives, which really helps in producing non-actor-based, high-concurrency, high-performance systems.

Then there's the matter of storing things on disk, which is always an interesting challenge. One reason people say you should never write your own database is because it is difficult to write something to disk while making sure it's safe and actually located where you think it is. We designed our system such that we have only a few places where we write to disk ourselves. Most everywhere else, we rely on existing embedded database systems that, over time, we've learned to make as pluggable as possible. Today we use four internal embedded database technologies altogether, with the two most popular being LMDB [Lightning Memory-mapped Database] and RocksDB, a Facebook derivative of LevelDB.

SHEEHY: When you mention storing things directly on disk, I think of all that has been said about how the common choice to sit on top of a file system comes with lots of conflicts that can keep you from designing your database correctly. Yet, I know you made a conscious decision to go with one specific type of file-system technology. What drove that choice, and how do you feel about it now?

SCHLOSSNAGLE: There absolutely is a performance penalty to be paid when you're operating on top of a file system. Most of that relates to baggage that doesn't help when you're running a database. With that said, there still are some significant data-integrity issues to be solved whenever you're looking at writing data to raw devices. The bottom line is: I can write something to a raw device. I can sync it there. I can read it back to make sure it's correct. But then I can go back to read it later, and it will no longer be correct. Bit rot is real, especially when you're working with large-scale systems. If you're writing an exabyte of data out, I can guarantee it's not all coming back. There are questions about how you deal with that.

Our choice to use ZFS was really about delivering value in a timely manner. That is, ZFS gave us growable storage volumes; the ability to add more disks seamlessly during operation; built-in compression; some safety guarantees around snapshot protection; checksumming; and provisions for the autorecovery of data. The ability to get all of that in one fell swoop made it worthwhile to take the performance penalty of using a file system.

The other part of this, of course, is that we would have had to build much of that ourselves, anyway. Could we have built that to achieve better performance? Probably, since we could have dispensed with a lot of unnecessary baggage, such as Posix compliance, which is something ZFS provides for. But that probably also would have required six or seven years of product development. Instead, we were able to get what we needed right out of the gate. It came at the price of a performance penalty, which we were willing to pay.

SHEEHY: Another consideration, which I'm sure you took into account, is that ZFS has a complicated history in the public eye, with many people having serious doubts about its legal status. Did you run into any difficulty with your customers around your decision to go with ZFS?

SCHLOSSNAGLE: Success is always defined in the court of public opinion. So, yes, I'd say the ZFS gamble was a risky proposition from the very start. Over time it proved to be a safe market choice due to the adoption of OpenZFS for Linux. Still, I have a feeling that if ZFS had not been made easily available on Linux, we would have needed to replatform owing to a widespread reticence to deploy ZFS.

In 2016, we had some serious discussions about whether we could move forward with ZFS, given that the majority of our customers deploy on Linux. We hung in there and delayed that decision long enough for OpenZFS on Linux to come through and legitimize our choice. But there was a time when we were close to abandoning ZFS.

SHEEHY: I fully understand your decision, but I continue to be perplexed about why you chose to go with four embedded database technologies. I assume they don't sit on top of the file system; you instead give them direct device access, right?

SCHLOSSNAGLE: No, our embedded databases also sit atop ZFS.

SHEEHY: How do four embedded database technologies prove useful to you?

SCHLOSSNAGLE: At root, the answer is: A single generalized technology rarely fits a problem perfectly; there's always compromise. So, why might I hesitate to use LMDB? It turns out that writing to LMDB can be significantly slower at scale than writing to the end of a log file, which is how LSM [log-structured merge]-style databases like Rocks work. But then, reading from a database like RocksDB is significantly slower than reading from a B+ tree database like LMDB. You must take all these tradeoffs and concessions into account when you're building a large-scale database system that spans years' worth of data and encompasses a whole range of access patterns and user requirements.

Ultimately, we chose to optimize various paths within our system so we could take advantage of the strengths of each of these database technologies while working around their weaknesses. For example, as you might imagine, we write all the data that comes in from the outside world into an LSM architecture (RocksDB) since that doesn't present us with any inherent performance constraints. You just write the data to a file, and, as long as you can sort things fast enough, you can keep up. Still, given that these databases can grow substantially over time, you need to keep an eye on that. I mean, if you have a 30-TB RocksDB database, you're going to be in a world of hurt.

We have a number of techniques to stay on top of this. Many of them have to do with time-sharding the data. We'll have a Rocks database that represents this week's data. Then, as the week closes up, we'll open a new database. Beyond that, after the previous week's database has remained unmodified for a while, we will ETL [extract, transform, load] it into another format in LMDB that better services read queries—meaning we *re*-optimize it.

In the end, the Rocks database we use to handle ingest is a key-value store, but those values then are stored in a very column-oriented manner. We also glue all that together so you can read from the write side and occasionally write to the read side. In the end, using and blending these two different techniques allows us to optimize for our predicted workloads.

Given the volumes of data the Circonus system needs to handle and process, optimization is critical. Indeed, the database contains thousands of tunable parameters to allow for that. Making the best use of those capabilities, however, requires extraordinary visibility into all aspects of system performance. Toward that end, Schlossnagle says HDR [high dynamic range] log-linear quantized histograms are used to "track everything" over time and even conduct experiments to find out how

> “
> **JUSTIN SHEEHY**
> **Whenever you're building a database for others to use, there is a tension between just how configurable or tunable you want to make it.**
> ”

performance might be optimized by changing certain isolated tunings.

Still, does the team have any regrets about how the system was built? Just a few.

SHEEHY: Whenever you're building a database for others to use, there is a tension between just how configurable or tunable you want to make it. At one end of that spectrum is the option of making almost every variable as user accessible as possible. The other extreme is to make everything as turnkey as possible, such that everything works pretty well and it's hard for users to accidentally break things. Obviously, the spectrum isn't nearly as linear as that, but there's this tension just the same. I'm curious to learn about the approach you took to sort that out. Even more, I'd like to hear about what you learned in that process and what adjustments you then found you needed to make.

SCHLOSSNAGLE: My own experience, having done it both ways, suggests there's no right answer. I will say, though, that this notion of autotuning, self-configuring software that always works is pretty much a pipe dream unless your use case happens to be really simple. Even if you do choose to let every single configuration parameter or setting be tunable, it's practically impossible to make it such that all those tuning combinations will be valid. You really could wreck your system if you're not careful.

And yet, we probably have between 5,000 and 10,000 tunable parameters inside the system that we can configure online. In fact, the vast majority of those are only internally documented. By way of Tier 2/Tier 3 support, we are able to investigate systems, speculate as to what might be causing some particular problem, and then try to hotpatch it. The feedback from that then informs the default-handling parameters for the software from that time on.

We also have some self-adjusting systems—mostly around concurrency control, throttling, backoffs, and that sort of thing—which more or less just measure use patterns and then self-tune accordingly. These are limited to those cases where we have extensive real-world experience, have seen the patterns before, and so have the wherewithal to build suitable models.

SHEEHY: I have to say that having up to 10,000 levers does seem to give you a *lot* of power. But how do your support folks figure out which one to touch? That is, what did you do to provide the live inspection capabilities people could use to really *understand* the system while it's running?

SCHLOSSNAGLE: That's something I can talk a lot about since one of the really interesting parts of our technology has to do with our use of high-definition histograms. We have an open source implementation of HDR log-linear quantized histograms, called circllhist, that's both very fast and memory efficient. With the help of that, we're able to track the performance of function boundaries for every single IOP [input/output operation], database operation, time in queue, and the amount of time required to hand off to another core in the system. This is something we use to track *everything* on the system, including every single RocksDB or LMDB `get()` or `put()`. The latency of each one of those is tracked in terms of nanoseconds. Within a little Web console, we're able to review any of those histograms and watch them change over time. Also, since this happens to be a time-series database, we can then put that data back in the database and connect our tooling to it to get time-series graphs for the 99.9th percentile of latencies for writing numeric data, for example.

Once the performance characteristics of the part you're looking to troubleshoot or optimize have been captured, you have what you need to perform controlled experiments where you can change one tuning at a time. This gives you a way to gather direct feedback by changing just one parameter and then tracking how that changes the performance of the system, while also looking for any unanticipated regressions in other parts of the system. I should add this all comes as part of an open source framework [https://github.com/circonus-labs/libmtev].

SHEEHY: It sounds like this has really paid off for you and that this whole undertaking has yielded some impressive returns. But I wonder, if you were just starting to build your time-series database today, is there anything you would do in a substantially different way?

SCHLOSSNAGLE: Absolutely! There are lots of things we would approach differently. The system we're talking about here is nine years old now, so plenty of innovation has been introduced since then that we could leverage. Also, under the heading of "Hindsight is 20/20," I wish I'd selected some different data structures that would have transitioned better from the in-memory data world to the on-disk data one—particularly in-memory indexes and in-memory caches where, at a certain volume, you actually do want to take a cache-style approach, but you also want it to be on disk for availability reasons.

And you really want to be able to treat that as semipermanent. Just think in terms of a 130-gig in-memory adaptive radix index, for example. Well, it turns out that building a 130-gig part is nontrivial. It would be nice if I could have those data structures map seamlessly to on-disk data structures. Of course, those are data structures that would have had little practical purpose in 2011 since they would have been too slow without technology such as NVMe [Non-Volatile Memory Express] supporting them. Still, making those data structures memory-independent—pointer invariant—would have been a really good investment. In fact, we're in the middle of that now.

Probably the biggest change I would make at this point—looking back over all the bugs that have surfaced in our product over time—is that I wouldn't write it in C and C++. Instead, if Rust had been around at the time, that's what I would have used. It would have been pretty fantastic to write the system that way since Rust, by introducing the borrow checker and ownership models of memory, has essentially managed to design away most of the issues that have caused faults in our software.

But now, I'd have to say that ship has already sailed; retooling our platform on Rust and reeducating our team at this point would be an intractable proposition. Still, I continue to see this as a missed opportunity because I think a Rust-based system would have served us better for many of the use cases we've encountered over the past few years. **C**

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

turing lecture

DOI:10.1145/3448250

How can neural networks learn the rich internal representations required for difficult tasks such as recognizing objects or understanding language?

BY YOSHUA BENGIO, YANN LECUN, AND GEOFFREY HINTON

Deep Learning for AI

TURING LECTURE

Yoshua Bengio, Yann LeCun, and Geoffrey Hinton are recipients of the 2018 ACM A.M. Turing Award for breakthroughs that have made deep neural networks a critical component of computing.

RESEARCH ON ARTIFICIAL neural networks was motivated by the observation that human intelligence emerges from highly parallel networks of relatively simple, non-linear neurons that learn by adjusting the strengths of their connections. This observation leads to a central computational question: How is it possible for networks of this general kind to learn the complicated internal representations that are required for difficult tasks such as recognizing objects or understanding language? Deep learning seeks to answer this question by using many layers of activity vectors as representations and learning the connection strengths that give rise to these vectors by following the stochastic gradient of an objective function that measures how well the network is performing. It is very surprising that such a conceptually simple approach has proved to be so effective when applied to large training sets using huge amounts of computation and it appears that a key ingredient is depth: shallow networks simply do not work as well.

We reviewed the basic concepts and some of the breakthrough achievements of deep learning several years ago.[63] Here we briefly describe the origins of deep learning, describe a few of the more recent advances, and discuss some of the future challenges. These challenges include learning with little or no external supervision, coping with test examples that come from a different distribution than the training examples, and using the deep learning approach for tasks that humans solve by using a deliberate sequence of steps which we attend to consciously—tasks that Kahneman[56] calls *system 2* tasks as opposed to *system 1* tasks like object recognition or immediate natural language understanding, which generally feel effortless.

From Hand-Coded Symbolic Expressions to Learned Distributed Representations

There are two quite different paradigms for AI. Put simply, the logic-inspired paradigm views sequential reasoning as the essence of intelligence and aims to implement reasoning in computers using hand-designed rules of inference that operate on hand-designed symbolic expressions that formalize knowledge. The brain-inspired paradigm views learning representations from data as the essence of intelligence and aims to implement learning by hand-designing or evolving rules for modifying the connec-

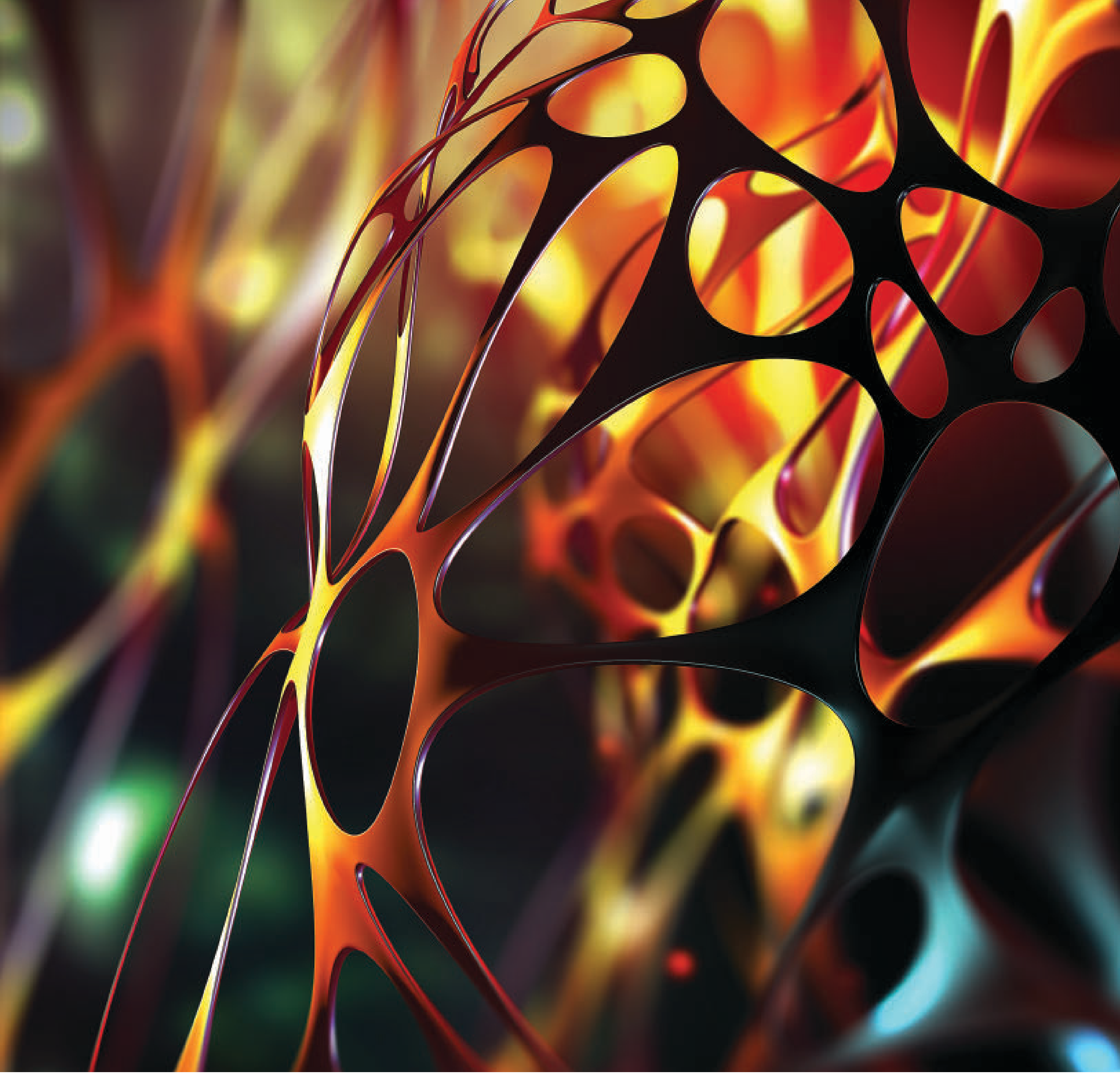


IMAGE BY YURCHANKA SIARHEI

tion strengths in simulated networks of artificial neurons.

In the logic-inspired paradigm, a symbol has no meaningful internal structure: Its meaning resides in its relationships to other symbols which can be represented by a set of symbolic expressions or by a relational graph. By contrast, in the brain-inspired paradigm the external symbols that are used for communication are converted into internal vectors of neural activity and these vectors have a rich similarity structure. Activity vectors can be used to model the structure inherent in a set of symbol strings by learning appropriate activity vectors for each symbol and learning non-linear transformations that allow the activity vectors that correspond to missing elements of a symbol string to be filled in. This was first demonstrated in Rumelhart et al.[74] on toy data and then by Bengio et al.[14] on real sentences. A very impressive recent demonstration is BERT,[22] which also exploits self-attention to dynamically connect groups of units, as described later.

The main advantage of using vectors of neural activity to represent concepts and weight matrices to capture relationships between concepts is that this leads to automatic generalization. If Tuesday and Thursday are represented by very similar vectors, they will have very similar causal effects on other vectors of neural activity. This facilitates analogical reasoning and suggests that immediate, intuitive analogical reasoning is our primary mode of reasoning, with logical sequential reasoning being a much later development,[56] which we will discuss.

The Rise of Deep Learning

Deep learning re-energized neural network research in the early 2000s by introducing a few elements which made it easy to train deeper networks. The emergence of GPUs and the availability of large datasets were key enablers of deep learning and they were greatly enhanced by the development of open source, flexible software platforms with automatic differentiation such as Theano,[16] Torch,[25] Caffe,[55] TensorFlow,[1] and PyTorch.[71] This made it easy to train complicated deep nets and to reuse the latest models and their building blocks. But the composition of more layers is what allowed more complex non-linearities and achieved surprisingly good results in perception tasks, as summarized here.

Why depth? Although the intuition that deeper neural networks could be more powerful pre-dated modern deep learning techniques,[82] it was a series of advances in both architecture and training procedures,[15,35,48] which ushered in the remarkable advances which are associated with the rise of deep learning. But why might deeper networks generalize better for the kinds of input-output relationships we are interested in modeling? It is important to realize that it is not simply a question of having more parameters, since deep networks often generalize better than shallow networks with the same number of parameters.[15] The practice confirms this. The most popular class of convolutional net architecture for computer vision is the ResNet family[43] of which the most common representative, ResNet-50 has 50 layers. Other ingredients not mentioned in this article but which turned out to be very useful include image deformations, dropout,[51] and batch normalization.[53]

We believe that deep networks excel because they exploit a particular form of compositionality in which features in one layer are combined in many different ways to create more abstract features in the next layer.

For tasks like perception, this kind of compositionality works very well and there is strong evidence that it is used by biological perceptual systems.[83]

Unsupervised pre-training. When the number of labeled training examples is small compared with the complexity of the neural network required to perform the task, it makes sense to start by using some other source of information to create layers of feature detectors and then to fine-tune these feature detectors using the limited supply of labels. In transfer learning, the source of information is another supervised learning task that has plentiful labels. But it is also possible to create layers of feature detectors without using any labels at all by stacking auto-encoders.[15,50,59]

First, we learn a layer of feature detectors whose activities allow us to reconstruct the input. Then we learn a second layer of feature detectors whose activities allow us to reconstruct the activities of the first layer of feature detectors. After learning several hidden layers in this way, we then try to predict the label from the activities in the last hidden layer and we backpropagate the errors through all of the layers in order to fine-tune the feature detectors that were initially discovered without using the precious information in the labels. The pre-training may well extract all sorts of structure that is irrelevant to the final classification but, in the regime where computation is cheap and labeled data is expensive, this is fine so long as the pre-training transforms the input into a representation that makes classification easier.

In addition to improving generalization, unsupervised pre-training initializes the weights in such a way that it is easy to fine-tune a deep neural network with backpropagation. The effect of pre-training on *optimization* was historically important for overcoming the accepted wisdom that deep nets were hard to train, but it is much less relevant now that people use rectified linear units (see next section) and residual connections.[43] However, the effect of pre-training on *generalization* has proved to be very important. It makes it possible to train very large models by leveraging large quantities of unlabeled data, for example, in natural language processing, for which huge corpora are available.[26,32] The general principle of pre-training and fine-tuning has turned out to be an important tool in the deep learning toolbox, for example, when it comes to transfer learning or even as an ingredient of modern meta-learning.[33]

The mysterious success of rectified linear units. The early successes of deep networks involved unsupervised pre-training of layers of units that used the logistic sigmoid nonlinearity or the closely related hyperbolic tangent. Rectified linear units had long been hypothesized in neuroscience[29] and already used in some variants of RBMs[70] and convolutional neural networks.[54] It was an unexpected and pleasant surprise to discover[35] that rectifying nonlinearities (now called ReLUs, with many modern variants) made it easy to train deep networks by backprop and stochastic gradient descent, without the need for layerwise pre-training. This was one of the technical advances that enabled deep learning to outperform previous methods for object recognition,[60] as outlined here.

Breakthroughs in speech and object recognition. An acoustic model converts a representation of the sound wave into a probability distribution over fragments of phonemes. Heroic efforts by Robinson[72] using transputers and by Morgan et al.[69] using DSP chips had already shown that, with sufficient processing power, neural networks were competitive with the state of the art for acoustic modeling. In 2009, two graduate students[68] using Nvidia GPUs showed that pre-trained deep neural nets could slightly outperform the SOTA on the TIMIT dataset. This result reignited the interest of several leading speech groups in neural networks. In 2010, essentially the same deep network was shown to beat the SOTA for large vocabulary speech recognition without requiring speaker-dependent training[28,46] and by 2012, Google had engineered a production version that significantly improved voice search on Android. This was an early demonstration of the disruptive power of deep learning.

At about the same time, deep learning scored a dramatic victory in the 2012 ImageNet competition, almost halving the error rate for recognizing a thousand different classes of object in natural images.[60] The keys to this victory were the major effort by Fei-Fei Li and her collaborators in collecting more than a million labeled images[31] for the training set and the very efficient use of multiple GPUs by Alex Krizhevsky. Current hardware, including GPUs, encourages the use of large mini-batches in order to amortize the cost of fetching a weight from memory across many uses of that weight. Pure online stochastic gradient descent which uses each weight once converges faster and future hardware may just use weights in place rather than fetching them from memory.

The deep convolutional neural net contained a few novelties such as the use of ReLUs to make learning faster and the use of dropout to prevent overfitting, but it was basically just a feed-forward convolutional neural net of the kind that Yann LeCun and his collaborators had been developing for many years.[64,65] The response of the computer vision community to this breakthrough was admirable. Given this incontrovertible evidence of the superiority of convolutional neural nets, the community rapidly abandoned previous hand-engineered approaches and switched to deep learning.

We believe that deep networks excel because they exploit a particular form of compositionality in which features in one layer are combined in many different ways to create more abstract features in the next layer.

Recent Advances

Here we selectively touch on some of the more recent advances in deep learning, clearly leaving out many important subjects, such as deep reinforcement learning, graph neural networks and meta-learning.

Soft attention and the transformer architecture. A significant development in deep learning, especially when it comes to sequential processing, is the use of multiplicative interactions, particularly in the form of soft attention.[7,32,39,78] This is a transformative addition to the neural net toolbox, in that it changes neural nets from purely vector transformation machines into architectures which can dynamically choose which inputs they operate on, and can store information in differentiable associative memories. A key property of such architectures is that they can effectively operate on different kinds of data structures including sets and graphs.

Soft attention can be used by modules in a layer to dynamically select which vectors from the previous layer they will combine to compute their outputs. This can serve to make the output independent of the order in which the inputs are presented (treating them as a set) or to use relationships between different inputs (treating them as a graph).

The transformer architecture,[85] which has become the dominant architecture in many applications, stacks many layers of "self-attention" modules. Each module in a layer uses a scalar product to compute the match between its query vector and the key vectors of other modules in that layer. The matches are normalized to sum to 1, and the resulting scalar coefficients are then used to form a convex combination of the value vectors produced by the other modules in the previous layer. The resulting vector forms an input for a module of the next stage of computation. Modules can be made multi-headed so that each module computes several different query, key and value vectors, thus making it possible for each module to have several distinct inputs, each selected from the previous stage modules in a different way. The order and number of modules does not matter in this operation, making it possible to operate on sets of vectors rather than single vectors as in traditional neural networks. For instance, a language translation system, when producing a word in the output sentence, can choose to pay attention to the corresponding group of words in the input sentence, independently of their position in the text. While multiplicative gating is an old idea for such things as coordinate transforms[44] and powerful forms of recurrent networks,[52] its recent forms have made it mainstream. Another way to think about attention mechanisms is that they make it possible to dynamically route information through appropriately selected modules and combine these modules in potentially novel ways for improved out-of-distribution generalization.[38]

Transformers have produced dramatic performance improvements that have revolutionized natural language processing,[27,32] and they are now being used routinely in industry. These systems are all pre-trained in a self-supervised manner to predict missing words in a segment of text.

Perhaps more surprisingly, transformers have been used successfully to solve integral and differential equations symbolically.[62] A very promising recent trend uses transformers on top of convolutional nets for object detection and localization in images with state-of-the-art performance.[19] The transformer performs post-processing and object-based reasoning in a differentiable manner, enabling the system to be trained end-to-end.

Unsupervised and self-supervised learning. Supervised learning, while successful in a wide variety of tasks, typically requires a large amount of human-labeled data. Similarly, when reinforcement learning is based only on rewards, it requires a very large number of interactions. These learning methods tend to produce task-specific, specialized systems that are often brittle outside of the narrow domain they have been trained on. Reducing the number of human-labeled samples or interactions with the world that are required to learn a task and increasing the out-of-domain robustness is of crucial importance for applications such as low-resource language translation, medical image analysis, autonomous driving, and content filtering.

Humans and animals seem to be able to learn massive amounts of background knowledge about the world, largely by observation, in a task-independent manner. This knowledge underpins common sense and allows humans to learn complex tasks, such as driving, with just a few hours of practice. A key question for the future of AI is how do humans learn so much from observation alone?

A key question for the future of AI is how do humans learn so much from observation alone?

In supervised learning, a label for one of N categories conveys, on average, at most $log_2(N)$ bits of information about the world. In model-free reinforcement learning, a reward similarly conveys only a few bits of information. In contrast, audio, images and video are high-bandwidth modalities that implicitly convey large amounts of information about the structure of the world. This motivates a form of prediction or reconstruction called self-supervised learning which is training to "fill in the blanks" by predicting masked or corrupted portions of the data. Self-supervised learning has been very successful for training transformers to extract vectors that capture the context-dependent meaning of a word or word fragment and these vectors work very well for downstream tasks.

For text, the transformer is trained to predict missing words from a discrete set of possibilities. But in high-dimensional continuous domains such as video, the set of plausible continuations of a particular video segment is large and complex and representing the distribution of plausible continuations properly is essentially an unsolved problem.

Contrastive learning. One way to approach this problem is through latent variable models that assign an energy (that is, a badness) to examples of a video and a possible continuation.[a]

Given an input video X and a proposed continuation Y, we want a model to indicate whether Y is compatible with X by using an energy function $E(X, Y)$ which takes low values when X and Y are compatible, and higher values otherwise.

$E(X, Y)$ can be computed by a deep neural net which, for a given X, is trained in a contrastive way to give a low energy to values Y that are compatible with X (such as examples of (X, Y) pairs from a training set), and high energy to other values of Y that are incompatible with X. For a given X, inference consists in finding one $\check{Y}$ that minimizes $E(X, Y)$ or perhaps sampling from the Ys that have low values of $E(X, Y)$. This energy-based approach to representing the way Y depends on X makes it possible to model a diverse, multi-modal set of plausible continuations.

The key difficulty with contrastive learning is to pick good "negative" samples: suitable points Y whose energy will be pushed up. When the set of possible negative examples is not too large, we can just consider them all. This is what a softmax does, so in this case contrastive learning reduces to standard supervised or self- supervised learning over a finite discrete set of symbols. But in a real-valued high-dimensional space, there are far too many ways a vector $\hat{Y}$ could be different from Y and to improve the model we need to focus on those Ys that should have high energy but currently have low energy. Early methods to pick negative samples were based on Monte-Carlo methods, such as contrastive divergence for restricted Boltzmann machines[48] and noise-contrastive estimation.[41]

Generative Adversarial Networks (GANs)[36] train a generative neural net to produce contrastive samples by apply-

[a] As Gibbs pointed out, if energies are defined so that they add for independent systems, they must correspond to negative log probabilities in any probabilistic interpretation.

ing a neural network to latent samples from a known distribution (for example, a Gaussian). The generator trains itself to produce outputs $\hat{Y}$ to which the model gives low energy $E(\hat{Y})$. The generator can do so using backpropagation to get the gradient of $E(\hat{Y})$ with respect to $\hat{Y}$. The generator and the model are trained simultaneously, with the model attempting to give low energy to training samples, and high energy to generated contrastive samples.

GANs are somewhat tricky to optimize, but adversarial training ideas have proved extremely fertile, producing impressive results in image synthesis, and opening up many new applications in content creation and domain adaptation[34] as well as domain or style transfer.[87]

Making representations agree using contrastive learning. Contrastive learning provides a way to discover good feature vectors without having to reconstruct or generate pixels. The idea is to learn a feed-forward neural network that produces very similar output vectors when given two different crops of the same image[10] or two different views of the same object[17] but dissimilar output vectors for crops from different images or views of different objects. The squared distance between the two output vectors can be treated as an energy, which is pushed down for compatible pairs and pushed up for incompatible pairs.[24,80]

A series of recent papers that use convolutional nets for extracting representations that agree have produced promising results in visual feature learning. The positive pairs are composed of different versions of the same image that are distorted through cropping, scaling, rotation, color shift, blurring, and so on. The negative pairs are similarly distorted versions of different images which may be cleverly picked from the dataset through a process called hard negative mining or may simply be all of the distorted versions of other images in a minibatch. The hidden activity vector of one of the higher-level layers of the network is subsequently used as input to a linear classifier trained in a supervised manner. This Siamese net approach has yielded excellent results on standard image recognition benchmarks.[6,21,22,43,67] Very recently, two Siamese net approaches have managed to eschew the need for contrastive samples. The first one, dubbed SwAV, quantizes the output of one network to train the other network,[20] the second one, dubbed BYOL, smoothes the weight trajectory of one of the two networks, which is apparently enough to prevent a collapse.[40]

Variational auto-encoders. A popular recent self-supervised learning method is the Variational Auto-Encoder (VAE).[58] This consists of an encoder network that maps the image into a latent code space and a decoder network that generates an image from a latent code. The VAE limits the information capacity of the latent code by adding Gaussian noise to the output of the encoder before it is passed to the decoder. This is akin to packing small noisy spheres into a larger sphere of minimum radius. The information capacity is limited by how many noisy spheres fit inside the containing sphere. The noisy spheres repel each other because a good reconstruction error requires a small overlap between codes that correspond to different samples. Mathematically, the system minimizes a free energy obtained through marginalization of the latent code over the noise distribution. However, minimizing this free energy with respect to the parameters is intractable, and one has to rely on variational approximation methods from statistical physics that minimize an upper bound of the free energy.

The Future of Deep Learning

The performance of deep learning systems can often be dramatically improved by simply scaling them up. With a lot more data and a lot more computation, they generally work a lot better. The language model GPT-3[18] with 175 billion parameters (which is still tiny compared with the number of synapses in the human brain) generates noticeably better text than GPT-2 with only 1.5 billion parameters. The chatbots Meena[2] and BlenderBot[73] also keep improving as they get bigger. Enormous effort is now going into scaling up and it will improve existing systems a lot, but there are fundamental deficiencies of current deep learning that cannot be overcome by scaling alone, as discussed here.

Comparing human learning abilities with current AI suggests several directions for improvement:

1. Supervised learning requires too much labeled data and model-free reinforcement learning requires far too many trials. Humans seem to be able to generalize well with far less experience.
2. Current systems are not as robust to changes in distribution as humans, who can quickly adapt to such changes with very few examples.
3. Current deep learning is most successful at perception tasks and generally what are called system 1 tasks. Using deep learning for system 2 tasks that require a deliberate sequence of steps is an exciting area that is still in its infancy.

What needs to be improved. From the early days, theoreticians of machine learning have focused on the iid assumption, which states that the test cases are expected to come from the same distribution as the training examples. Unfortunately, this is not a realistic assumption in the real world: just consider the non-stationarities due to actions of various agents changing the world, or the gradually expanding mental horizon of a learning agent which always has more to learn and discover. As a practical consequence, the performance of today's best AI systems tends to take a hit when they go from the lab to the field.

Our desire to achieve greater robustness when confronted with changes in distribution (called out-of-distribution generalization) is a special case of the more general objective of reducing sample complexity (the number of examples needed to generalize well) when faced with a new task—as in transfer learning and lifelong learning[81]—or simply with a change in distribution or in the relationship between states of the world and rewards. Current supervised learning systems require many more examples than humans (when having to learn a new task) and the situation is even worse for model-free reinforcement learning[23] since each rewarded trial provides less information about the task than each labeled example. It has already been noted[61,76] that humans can generalize in a way that is different and more powerful than ordinary iid generalization: we can correctly interpret novel combinations of existing concepts, even if those combinations are extremely unlikely under our training distribution, so long as they respect high-level syntactic and semantic patterns we have already learned. Recent studies help us clarify how different neural net architectures fare in terms of this systematic generalization ability.[8,9] How can we design future machine learning systems with these abilities to generalize better or adapt faster out-of-distribution?

From homogeneous layers to groups of neurons that represent entities. Evidence from neuroscience suggests that groups of nearby neurons (forming what is called a hyper-column) are tightly connected and might represent a kind of higher-level vector-valued unit able to send not just a scalar quantity but rather a set of coordinated values. This idea is at the heart of the capsules architectures,[47,59] and it is also inherent in the use of soft-attention mechanisms, where each element in the set is associated with a vector, from which one can read a key vector and a value vector (and sometimes also a query vector). One way to think about these vector-level units is as representing the detection of an object along with its attributes (like pose information, in capsules). Recent papers in computer vision are exploring extensions of convolutional neural networks in which the top level of the hierarchy represents a set of candidate objects detected in the input image, and operations on these candidates is performed with transformer-like architectures.[19,84,86] Neural networks that assign intrinsic frames of reference to objects and their parts and recognize objects by using the geometric relationships between parts should be far less vulnerable to directed adversarial attacks,[79] which rely on the large difference between the information used by people and that used by neural nets to recognize objects.

Multiple time scales of adaption. Most neural nets only have two timescales: the weights adapt slowly over many examples and the activities adapt rapidly changing with each new input. Adding an overlay of rapidly adapting and rapidly, decaying "fast weights"[49] introduces interesting new computational abilities. In particular, it creates a high-capacity, short-term memory,[4] which allows a neural net to perform true recursion in which the same neurons can be reused in a recursive call because their activity vector in the higher-level call can be reconstructed later using the information in the fast weights. Multiple time scales of adaption also arise in learning to learn, or meta-learning.[12,33,75]

Higher-level cognition. When thinking about a new challenge, such as driving in a city with unusual traffic rules, or even imagining driving a vehicle on the moon, we can take advantage of pieces of knowledge and generic skills we have already mastered and recombine them dynamically in new ways. This form of systematic generalization allows humans to generalize fairly well in contexts that are very unlikely under their training distribution. We can then further improve with practice, fine-tuning and compiling these new skills so they do not need conscious attention anymore. How could we endow neural networks with the ability to adapt quickly to new settings by mostly reusing already known pieces of knowledge, thus avoiding interference with known skills? Initial steps in that direction include Transformers[32] and Recurrent Independent Mechanisms.[38]

It seems that our implicit (system 1) processing abilities allow us to guess potentially good or dangerous futures, when planning or reasoning. This raises the question of how system 1 networks could guide search and planning at the higher (system 2) level, maybe in the spirit of the value functions which guide Monte-Carlo tree search for AlphaGo.[77]

Machine learning research relies on inductive biases or priors in order to encourage learning in directions which are compatible with some assumptions about the world. The nature of system 2 processing and cognitive neuroscience theories for them[5,30] suggests several such inductive biases and architectures,[11,45] which may be exploited to design novel deep learning systems. How do we design deep learning architectures and training frameworks which incorporate such inductive biases?

The ability of young children to perform causal discovery[37] suggests this may be a basic property of the human brain, and recent work suggests that optimizing out-of-distribution generalization under interventional changes can be used to train neural networks to discover causal dependencies or causal variables.[3,13,57,66] How should we structure and train neural nets so they can capture these underlying causal properties of the world?

How are the directions suggested by these open questions related to the symbolic AI research program from the 20th century? Clearly, this symbolic AI program aimed at achieving system 2 abilities, such as reasoning, being able to factorize knowledge into pieces which can easily recombined in a sequence of computational steps, and being able to manipulate abstract variables, types, and instances. We would like to design neural networks which can do all these things while working with real-valued vectors so as to preserve the strengths of deep learning which include efficient large-scale learning using differentiable computation and gradient-based adaptation, grounding of high-level concepts in low-level perception and action, handling uncertain data, and using distributed representations.

References

1. Abadi, M. et al. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symp. Operating Systems Design and Implementation*, 2016, 265–283.
2. Adiwardana, D., Luong, M., So, D., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. Towards a human-like open-domain chatbot 2020; *arXiv preprint arXiv:2001.09977*.
3. Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization, 2019; *arXiv preprint arXiv:1907.02893*.
4. Ba, J., Hinton, G., Mnih, V., Leibo, J., and Ionescu, C. Using fast weights to attend to the recent past. *Advances in Neural Information Processing Systems*, 2016, 4331–4339.
5. Baars, B. *A Cognitive Theory of Consciousness.* Cambridge University Press, Cambridge, MA, 1993.
6. Bachman, P., Hjelm, R., and Buchwalter, W. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems*, 2019, 15535–15545.
7. Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2014; *arXiv:1409.0473*.
8. Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T., Vries, H., and Courville, A. Systematic generalization: What is required and can it be learned? 2018; *arXiv:1811.12889*.
9. Bahdanau, D., de Vries, H., O'Donnell, T., Murty, S., Beaudoin, P., Bengio, Y., and Courville, A. Closure: Assessing systematic generalization of clever models, 2019; *arXiv:1912.05783*.
10. Becker, S. and Hinton, G. Self-organizing neural network that discovers surfaces in random dot stereograms. *Nature 355*, 6356 (1992), 161–163.
11. Bengio, Y. The consciousness prior, 2017; *arXiv:1709.08568*.
12. Bengio, Y., Bengio, S., and Cloutier, J. Learning a synaptic learning rule. In *Proceedings of the IEEE 1991 Seattle Intern. Joint Conf. Neural Networks 2*.
13. Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. A meta-transfer objective for learning to disentangle causal mechanisms. In *Proceedings of ICLR'2020; arXiv:1901.10912*.
14. Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *NIPS'2000*, 2001, 932–938.
15. Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In

Proceedings of NIPS'2006, 2007.

16. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. Theano: A CPU and GPU math expression compiler. In *Proceedings of SciPy*, 2010.
17. Bromley, J., Guyon, I., LeCun, Y., Säkinger, E., and Shah, R. Signature verification using a "Siamese" time delay neural network. *Advances in Neural Information Processing Systems*, 1994, 737–744.
18. Brown, T. et al. Language models are few-shot learners, 2020; *arXiv:2005.14165*.
19. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *Procedings of ECCV'2020*; *arXiv:2005.12872*.
20. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments, 2020;. *arXiv:2006.09882*.
21. Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020; *arXiv:2002.05709*.
22. Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning, 2020; *arXiv:2003.04297*.
23. Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T., and Bengio, Y. Babyai: First steps towards grounded language learning with a human in the loop. In Proceedings in ICLR'2019; *arXiv:1810.08272*.
24. Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conf. Computer Vision and Pattern Recognition 1*, 539–546.
25. Collobert, R., Kavukcuoglu, K., and Farabet, C. Torch7: A matlab-like environment for machine learning. In *Proceedings of NIPS Worskshop BigLearn*, 2011.
26. Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML'2008*.
27. Conneau, A. and Lample, G. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems 32*, 2019. H. Wallach et al., eds. 7059–7069. Curran Associates, Inc.; http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf.
28. Dahl, G., Yu, D., Deng, L., and Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio, Speech, and Language Processing 20*, 1 (2011), 30–42.
29. Dayan, P. and Abbott, L. *Theoretical Neuroscience*. The MIT Press, 2001.
30. Dehaene, S., Lau, H., and Kouider, S. What is consciousness, and could machines have it? *Science 358*, 6362 (2017, 486–492.
31. Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of 2009 IEEE Conf. Computer Vision and Pattern Recognition*, 248–255.
32. Devlin, J., Chang, M., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of ACL'2019*; a*rXiv:1810.04805*.
33. Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks, 2017; a*rXiv:1703.03400*.
34. Ganin, Y and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *Proceedings of Intern. Conf. Machine Learning*, 2015, 1180–1189.
35. Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of AISTATS'2011*.
36. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014, 2672–26804.
37. Gopnik, A., Glymour, C., Sobel, D., Schulz, L., Kushnir, T., and Danks, D. A theory of causal learning in children: causal maps and bayes nets. *Psychological Review 111*, 1 (2004).
38. Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. Recurrent independent mechanisms, 2019; *arXiv:1909.10893*.
39. Graves, A. Generating sequences with recurrent neural networks, 2013; *arXiv:1308.0850*.
40. Grill, J-B. et al. Bootstrap your own latent: A new approach to self-supervised learning, 2020; *aeXiv:2006.07733*.
41. Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th Intern. Conf. Artificial Intelligence and Statistics*, 2010, 297–304.
42. He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of CVPR'2020*, June 2020.
43. He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of CVPR'2016, 770–778*.
44. Hinton, G. A parallel computation that assigns canonical object-based frames of reference. In *Proceedings of the 7th Intern. Joint Conf. Artificial Intelligence 2*, 1981, 683–685.
45. Hinton, G. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence 46*, 1-2 (1990), 47–75.
46. Hinton, G. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing 29*, 6 (2012), 82–97.
47. Hinton, G., Krizhevsky, A., and Wang, S. Transforming auto-encoders. In *Proceedings of Intern. Conf. Artificial Neural Networks*. Springer, 2011, 44–51.
48. Hinton, G., Osindero, S., and Teh, Y-W. A fast-learning algorithm for deep belief nets. *Neural Computation 18* (2006), 1527–1554.
49. Hinton, G. and Plaut, D. Using fast weights to deblur old memories. In *Proceedings of the 9th Annual Conf. Cognitive Science Society*, 1987, 177–186.
50. Hinton, G. and Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science 313* (July 2006), 504–507.
51. Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Improving neural networks by preventing co-adaptation of feature detectors. In *Proceedings of NeurIPS'2012*; arXiv:1207.0580.
52. Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation 9*, 8 (1997), 1735–1780.
53. Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.
54. Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. What is the best multi-stage architecture for object recognition? In *Proceedings of ICCV'09*, 2009.
55. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM Intern. Conf. Multimedia*, 2014, 675–678.
56. Kahneman, D. *Thinking, Fast and Slow*. Macmillan, 2011.
57. Ke, N., Bilaniuk, O., Goyal, A., Bauer, S., Larochelle, H., Pal, C., and Bengio, Y. Learning neural causal models from unknown interventions, 2019; *arXiv:1910.01075*.
58. Kingma, D. and Welling, M. Auto-encoding variational bayes. In *Proceedings of the Intern. Conf. Learning Representations*, 2014.
59. Kosiorek, A., Sabour, S., Teh, Y., and Hinton, G. Stacked capsule autoencoders. *Advances in Neural Information Processing Systems*, 2019, 15512–15522.
60. Krizhevsky, A., Sutskever, I., and Hinton, G. ImageNet classification with deep convolutional neural networks. In *Proceedings of NIPS'2012*.
61. Lake, B., Ullman, T., Tenenbaum, J., and Gershman, S. Building machines that learn and think like people. *Behavioral and Brain Sciences 40* (2017).
62. Lample, G. and Charton, F. Deep learning for symbolic mathematics. In *Proceedings of ICLR'2020*; *arXiv:1912.01412*.
63. LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature 521*, 7553 (2015), 436–444.
64. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*, 4 (1989), 541–551.
65. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.
66. Lopez-Paz, D., Nishihara, R., Chintala, S., Scholkopf, B., and Bottou, L. Discovering causal signals in images. In *Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition*, 2017, 6979–6987.
67. Misra, I. and Maaten, L. Self-supervised learning of pretext-invariant representations. In *Proceedings of CVPR'2020*, June 2020; *arXiv:1912.01991*.
68. Mohamed, A., Dahl, G., and Hinton, G. Deep belief networks for phone recognition. In *Proceedings of NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*. (Vancouver, Canada, 2009).
69. Morgan, N., Beck, J., Allman, E., and Beer, J. Rap: A ring array processor for multilayer perceptron applications. In *Proceedings of the IEEE Intern. Conf. Acoustics, Speech, and Signal Processing*, 1990, 1005–1008.
70. Nair, V. and Hinton, G. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the ICML'2010*.
71. Paszke, A., et al. Automatic differentiation in pytorch. 2017.
72. Robinson, A. An application of recurrent nets to phone probability estimation. *IEEE Trans. Neural Networks 5*, 2 (1994), 298–305.
73. Roller, S., et al. Recipes for building an open domain chatbot, 2020; *arXiv:2004.13637*.
74. Rumelhart, D., Hinton, G., and Williams, R. Learning representations by back-propagating errors. *Nature 323* (1986), 533–536.
75. Schmidhuber, J. Evolutionary principles in self-referential learning. Diploma thesis, Institut f. Informatik, Tech.Univ. Munich, 1987.
76. Shepard, R. Toward a universal law of generalization for psychological science. *Science 237*, 4820 (1987), 1317–1323.
77. Silver, D., et al. Mastering the game of go with deep neural networks and tree search. *Nature 529*, 7587 (2016), 484.
78. Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. End-to-end memory networks. *Advances in Neural Information Processing Systems 28*, 2015, 2440–2448. C. Cortes et al., eds. Curran Associates, Inc.; http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf.
79. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *Proceedings of ICLR'2014*; *arXiv:1312.6199*.
80. Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. Web-scale training for face identification. In *Proceedings of CVPR'2015*, 2746–2754.
81. Thrun, S. Is learning the n-th thing any easier than learning the first? In *Proceedings of NIPS'1995*. MIT Press, Cambridge, MA, 640–646.
82. Utgoff, P. and Stracuzzi, D. Many-layered learning. *Neural Computation 14* (2002), 2497–2539, 2002.
83. Van Essen, D. and Maunsell, J. Hierarchical organization and functional streams in the visual cortex. *Trends in Neurosciences 6* (1983), 370–375.
84. van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions, 2018; *arXiv:1802.10353*.
85. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, T., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017, 5998–6008.
86. Zambaldi, V., et al. Relational deep reinforcement learning, 2018; arXiv:1806.01830.
87. Zhu, J-Y., Park, T., Isola, P., and Efros, A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the 2017 IEEE Intern. Conf. on Computer Vision*, 2223–2232.

Yoshua Bengio is a professor in the Department of Computer Science and Operational Research at the Université de Montréal. He is also the founder and scientific director of Mila, the Quebec Artificial Intelligence Institute, and the co-director of CIFAR's Learning in Machines & Brains program.

Yann LeCun is VP and Chief AI Scientist at Facebook and Silver Professor at New York University affiliated with the Courant Institute of Mathematical Sciences and the Center for Data Science, New York, NY, USA.

Geoffrey Hinton is the Chief Scientific Advisor of the Vector Institute, Toronto, Vice President and Engineering Fellow at Google, and Emeritus Distinguished Professor of Computer Science at the University of Toronto, Canada.

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

DOI:10.1145/3431280

BRAN KNOWLES
Lancaster University,
Lancashire, England, U.K.

VICKI L. HANSON
Association for Computing Machinery,
New York, NY, USA.

YVONNE ROGERS
University College
London, England, U.K.

ANNE MARIE PIPER
University of California,
Irvine, CA, USA.

JENNY WAYCOTT
University of Melbourne,
Australia.

NIGEL DAVIES
Lancaster University,
Lancashire, England, U.K.

ALOHA HUFANA AMBE
Queensland University of Technology,
Brisbane, Australia.

ROBIN N. BREWER
University of Michigan,
Ann Arbor, MI, USA.

DEBALEENA CHATTOPADHYAY
University of Illinois at Chicago,
IL, USA.

MARIANNE DEE
University of Dundee,
Scotland, U.K.

DAVID FROHLICH
University of Surrey,
Guildford, England, U.K.

MARISELA GUTIERREZ-LOPEZ,
University of Bristol,
England, U.K.

BEN JELEN
Indiana University,
Bloomington, IN, USA.

AMANDA LAZAR
University of Maryland at College Park,
MD, USA.

RADOSLAW NIELEK
Polish-Japanese Academy
of Information Technology,
Warsaw, Poland.

BELÉN BARROS PENA
Northumbria University,
Newcastle upon Tyne, U.K.

ABI ROPER
City University of London,
England, U.K.

MARK SCHLAGER
Google Inc.,
San Carlos, CA, USA.

BRITTA SCHULTE
Bauhaus University,
Weimar, Germany.

IRENE YE YUAN
University of Minnesota,
Minneapolis, MN, USA.

Including older adults as full stakeholders in digital society.

The Harm in Conflating Aging With Accessibility

"The quest for youth—so futile. Age and wisdom have their graces too." —Jean Luc Picard

IT IS AN increasingly global phenomenon that societies promote the notion of youth as the preferred state.[a] In stark contrast to the "wise elder" of ages past, today old age is assumed to be marked by loss of physical and cognitive ability, diminished relevance, and as we are sadly seeing with the COVID-19 pandemic, devalued humanity.[18] In many ways, it is not surprising that such stereotypes are reflected in our technologies: tech companies compete for territory in an already overcrowded youth market; whereas older adults,[b] if considered users at all, are offered little more than fall alarms, activity monitors, and senior-friendly (often lower functionality) versions of existing tools. Meanwhile, there is a growing trend

[a] *The New Yorker.* Why ageism never gets old; https://www.newyorker.com/magazine/2017/11/ 20/why-ageism-never-gets-old

[b] Whether "older adult" is even a meaningful category of user is something we question in this article.

IMAGE BY MRMOHOCK/SHUTTERSTOCK

of workers aging out of the tech industry as early as their mid-40s,[17] reflecting the higher value placed on the perspectives of those who represent the default target demographic.

ACM has produced a Code of Ethics and Professional Conduct,[1] which affirms the importance of computing technologies being accessible as well as meeting the social needs of a diverse population of users. In light of such principles, it is ethically problematic that individuals toward the far (and particularly farthest) end of the age spectrum are clearly the lesser beneficiaries of digital technologies.[3] There are competing views on why this is the case. On the one hand, older adults are more likely than younger adults to have multiple health related constraints which can present difficulties in using standard (or, shall we say, poorly designed) technologies. But differences in technology adoption rates between young and old may more accurately reflect technologies' lack of appeal to older adults than their inaccessibility. After all, healthy older adults have been shown to reject digital technologies when they are perceived to be in conflict with "what matters" in their lives and to society at large.[12]

It is our contention that usability concerns have for too long overshadowed questions about the usefulness and acceptability of digital technologies for older adults. In this article, we confront the uneasy relationship be-

» key insights

- **The narrow focus on accessibility concerns harms older adults by excluding them from wider benefits of technologies.**
- **Technologies must be usable by older adults who may have age-related limitations; but they also must be useful and acceptable to older adults, many of whom do not have age-related limitations.**
- **Older adults are important stakeholders in digital society, and listening to their objections to technologies will help us design our society in ways that work better for everyone.**
- **The key to designing technologies that older adults want to use is to think differently: to reject a mindset that aging is disabling, to understand that aging is a process, and to recognize more positive aspects of that process.**

tween accessibility and aging research—specifically, the assumption that the two fall under the same umbrella *despite the fact that aging is neither an illness nor a disability*. Our point is not that the phenomenon of disability represents a comparatively simple challenge for designers, as assistive devices and accessibility adaptations are inadequate for users with disabilities for many of the same reasons we highlight in this article.[20] Instead, we argue that while accessibility research is important as one aspect of ensuring individuals are not unfairly discriminated against, Human-Computer Interaction (HCI) and Aging research should be seen as a separate entity. As a basis from which more inclusive HCI and Aging research may spring, we eschew notions of "old age" in favor of the alternative framing of aging as a largely positive process to which all people are subject.

What's the Harm?

We begin by laying out the ways in which conflating aging with accessibility inadvertently harms older adults. Our argument is essentially Foucauldian:[8] that assuming a natural connection between aging and accessibility works to reify "older adult" as a category of user/subject in ways that disproportionately benefit younger technology users. In other words, understanding older adult users through the lens of accessibility—in which disability is the most salient characteristic in technology use—has implications for the kinds of design opportunities we identify and prioritize for this group, and in turn for how older adults see themselves as belonging (or rather not belonging) within digital society. Here, we break this down into four related problems stemming from the casual association between aging and accessibility research.

Focusing on age-related limitations perpetuates negative stereotypes of aging and promotes ageism. That negative narratives of aging abound is not the fault of digital technology, though it is important that researchers and designers are cognizant of the ways in which negative societal attitudes toward aging are reflected in and reinforced by technology. Fundamentally, this bias affects what behaviors are visible or invisible to designers: it is easier to see what older adults can't do, rather than what they can do, and thus design aims to compensate for age-related deficit[22,24] rather than designing to support late-life development and enrichment.[5]

Research has shown that attitudes toward aging concretely impact the way people age. People who hold more negative attitudes are found to be more likely to show biomarkers for Alzheimer's disease in their brains and as a result experience greater cognitive changes.[14] This suggests there is a strong psycho-social component to aging, and as such, technology has a role to play in shaping how we conceive of, and in turn, experience old age. Consider the following:

- The lack of interest by the tech sector in designing for the older adult market contributes to implicit ageist messaging that older adults are not worthy of investment, technologically or otherwise, and any harms that may occur to older adults through this omission is an unavoidable externality of a system that economically rewards other priorities.
- Technologies specially designed for "old people" that seem to prove ageist stereotypes can be actively stigmatizing for users, and therefore are rarely adopted unless one has no other choice.[23,26]
- If one internalizes the stereotype that old people are incapable of using technology, any poorly designed technology that presents usability challenges can make someone who otherwise doesn't consider themselves old *feel* old—a phenomenon known as situated elderliness.[4]
- Similarly, notions of "aging successfully" involve older adults being able to keep up with technological change, putting pressure on older adults to adopt and master new technologies lest they reveal their old age.[25,26]

In each of these ways, technology works to define what it means to be old: it *subjectifies* older people and turns aging itself into a *problem*. Through the experience of digital technologies, one is forced to either identify as an "older adult" or deliberately refuse the term. It is interesting that older adults can use these stereotypes to their advantage, insofar as claiming that one is "too old" frees the person from having to adopt technologies they otherwise don't want to adopt.[12] But resisting technologies by conceding the stereotype, unfortunately, only reinforces this subjectification.

Designing for potential physical and/or cognitive differences of older adults to the exclusion of other contextual factors limits the scope for technology to meaningfully relate to and positively contribute to the older adult experience. As people reach advanced old age, their individual personalities and identities can sometimes become hidden; the world only sees them as "old."[9] It is an insult to older adults—and undermines the identity building work that is so important to wellbeing in older adulthood—that digital technologies do not represent or accommodate their individuality as users.

Not only are older adults as multifaceted as people of any other age, they also differ from younger adults in a variety of important ways that are too often overlooked when blinded by potential differences in ability (see sidebar: "What Makes Older Adults Interesting?"). One cannot construct a well-defined design problem when ignoring these factors, and digital technologies are therefore less likely to appeal to and work for older adults.

Assuming older adults lack the ability to use digital technologies makes it harder to conceive of meaningful contributions they might make as co-designers of technology futures. Simply put, designers cannot take seriously the opinions of those they infantalize. Technologies are almost always designed *for* older adults, rather than by or with older adults. This can lead to situations where older adults do not find such digital technologies relevant to their lives.

It can be more difficult to reach older adult populations for research (perhaps the research community ought to ask why this is the case), but studies that have sought to engage older adults in co-design have demonstrated they can be productively engaged in helping envision the future.[2,10,22] Older adults have much to contribute to their communities, other generations and research, if enabled. They are a storehouse of personal experience and historical knowledge, and researchers as well as designers can learn a lot from their frank insights and vision for society.

Construing older adults as differently abled contributes to the "other-

ing" of older adult users. In addition to limiting the scope of innovation for older adult end users (as described previously), a preoccupation with physical deterioration and limitations associated with aging can obscure key design issues underlying older adults' difficulties with or objections to technologies. Younger adults also struggle to use poorly designed technologies and are frustrated with negative consequences of these technologies, but often they have greater incentive to learn to use such technologies, indeed often less choice whether to adopt them.[12] When older adults choose not to adopt these same technologies, assuming their non-use is due to a lack in ability (physical or cognitive) or digital literacy is a way of de-legitimizing these objections, while also concealing that the objections likely pertain to other users.

The categorical separation of older adults is how they come to be thought of as a problematic "other" or a divergent user group. It is part of an apparatus that encourages the treatment of older adults as peripheral to digital society. And if older adults are considered essentially problematic or difficult to accommodate, this feeds a vicious cycle: older adults are less likely to use technologies that do not account for their needs and wants; then not using these tools means they lose confidence in their technical abilities, and so are even less likely to use digital technologies. Ultimately, this ends up justifying the decision not to invest in the older adult market, focusing instead on delivering technologies that appeal to younger users.

What Makes Older Adults Interesting?

Older adults are not a well-defined category of user,[21] in part because there is no set age that makes someone "older." HCI and Aging research has largely failed to make clear what is different or interesting about older adults beyond their likeliness to experience usability issues related to age-related physical and/or cognitive decline. We have debunked this already as the sole reason for focusing on older adults; and yet there are several contextual factors that make older adults uniquely interesting to research and design for.

Life experiences. **Older adults have lived through more/different historical events and cultural shifts, which shape their view of the world, even if in different ways to one another.**

Technology biographies. **As part of their life experiences, older people have learned a variety of technologies across their lifespan, not all of which have been digital. These inform the way they approach their interactions with new technologies and can contribute to discomfort with novel forms of interaction, particularly if introduced to them post-retirement. Most importantly, however, they shape their understandings of what makes for "good" or "bad" technologies.**

Societal expectations. **Whether an older person individually ascribes to positive or negative views of aging, they will be aware of and react to/against these narratives in ways that affect their use of technology. Older adults can be ageist against themselves and their peers, too,[15] just like younger people.[6] Lack of expectation for their proficiency or comfort with technology, however, allows older adults to voice criticisms of technology that others either take for granted or must suffer through as a necessary means of accomplishing everyday or work-related tasks.**

Changing family structures. **Often older adults have to navigate multigenerational bonds and caring responsibilities (for example, for spouses, grandchildren, friends, their own elderly parents), putting particular constraints on their time and energy.**

Stage of life. **While a luxury not all older adults are guaranteed, retirement can precipitate a number of dramatic changes in one's social life, create new opportunities, and stimulate rapid identity building. As one perceives the end of life to be near (either due to advanced age or ill health), people seek more meaningful, emotionally fulfilling relationships (see socioemotional selectivity theory), thus giving them a new perspective on what might be important and not important when engaging with technologies.**

Taken together, older adults offer a perspective that can deepen understanding of the effects of digital technologies, so that we, as designers, can better understand the trade-offs entailed by our design decisions. Also, actively engaging with older adults helps to mitigate designers' own latent ageism—something one must do as a deliberate practice—resulting in technologies more likely to enrich the lives of those who are fortunate enough to arrive at older adulthood.

Thinking Differently, Designing Differently

Having identified the problems with the current view of aging in HCI, there are clear alternatives. We offer the following recommendations as an antidote to the harms identified earlier, directly mirroring each in turn:

Seek design inspiration in narratives of positive aging. It is worth pointing out that conflating aging with accessibility is not just a way of making aging more tractable as a design problem. It is really a *mindset*—a mindset that views the old as infirm, incompetent and in need of help. This view has its origins in the medical model of disability, whereas we are adopting a more social, positive and ultimately empowering model.

The first step in challenging this mindset is to consciously attend to the more positive aspects of the aging experience. Research and design could focus on the relative freedoms that older adults enjoy compared with those busy with child rearing or work life, and the space this opens up for being able to engage with questions about "what matters." Thereby, retirement becomes not an end but a new beginning, a chance to re-invent oneself or renew interests in hobbies, to travel, or to volunteer in the community. Designers could look to older adults as "elders"—those experienced in the art of living whose advice society should actively seek as we design our world.

Making this adjustment has two effects, therefore:

- Considering older adults as teachers and custodians of culture—roles they have held or still hold in many societies—enables one to benefit from older adults' (and indeed, older members of the design team's) wealth of experience. Designing technologies with this in mind can overcome naive and shortsighted development and instead can lead to technologies that substantially contribute to a life worth living.
- Researchers and designers can readily conceive of older adulthood as a site for exciting innovation. How radical it would be to put older adults at the frontline of our most innovative technologies, creating and benefiting from technologies that others also want to use.

Construct user types on the basis of shared contextual factors. People are different, and nothing can ever really work for everyone; but divergent perspectives ought to be accommodated within a digital society. Diversity nourishes insight and innovation; it helps society to become more empathetic, and design more compassionate technologies. Enfolding older adult perspectives and enriching the diversity of user types can only improve one's ability to design good technologies, hence the importance of working to understand what motivates these individuals.

But we caution that "older adult" is not a user type that is meaningful enough to inspire good design. This group, such as it is, is not monolithic, and the tendency to treat it as such perpetuates harmful stereotypes that they, as users, rarely conform to. In addition, treating older adults as a distinct user type isolates the transferable insights that the HCI community might gain from their perspectives and experiences.

To combat this, it is critical that research and design specify user types untethered from age. Some issues may be more salient for younger or older adults, but there are few issues that are so specific to chronological age that the user type of interest could not be represented by either an older or younger adult. And yet, while HCI research would benefit from more age-diverse samples, there may be especial cause to turn to older adults to better understand the impacts of technology within a wider historical and social context.

Empower older adults to envision and shape the future. It would be paternalistic to ignore that health is very important to older adults, even if one disagrees with some of the reasons why health has become such a big focus. But it is paternalistic, too, to assume that because someone is older, we designers know what is best for them—that it is in their best interest to adopt assistive technologies, for example. Paternalism is antithetical to the kind of listening stance that underpins good design work. It makes it more difficult to hear what older adults are really saying, or even to ask the right questions to begin with.

The ACM Code of Ethics and Professional conduct states, "*A computing professional should …* 1.1. Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing."[1] With this in mind, it is key to conceive of older adults as stakeholders not only in the particular subset of technologies specifically designed with them in mind, but also in the technologies that directly and indirectly shape the wider society in which they are, clearly, stakeholders. There is a casual ageism that assumes that individuals who may be closer to death are not worth consulting about technologies of the future. In our experience interviewing older adults, they take a great interest in how technologies may affect the lives of their grandchildren, for example, and make certain decisions about their own technology use to try to bring about more positive future.[11]

Diversity nourishes insight and innovation; it helps society to become more empathetic, and design more compassionate technologies.

There is a methodological solution here, not unrelated to our previous recommendation, which is to involve older adults as study participants and/or co-designers as a rule, not as the exception. Merely considering what individuals across the age spectrum might want is not sufficient for inclusive design. What makes design truly inclusive is ensuring that all stakeholders have a voice in the design process and are respected and engaged with as equals. We note that some efforts are being made to include older adults in design and discussions about technology (our reference list includes a number of good examples),[c] but there is still a long way to go.

Design for everyone "growing old." In sustainability discourse, there is a phenomenon known as NIMBY-ism: Not-In-My-Back-Yard. It describes a person not minding others having to live near wind turbines or nuclear power plants, but personally not wanting to live near them. The equivalent in the HCI and Aging field is the ". . . but they need it" argument: "I don't want any of this, but it would be great for my aging father." Technology carries so much stigma, and people don't want to be that "older adult" who is being designed for. It is telling that while most people over the age of 65 do not self-identify as "old"[d] nor see themselves as needing senior-friendly or assistive technologies, these are still designed "for their own good," ignoring their objections.

[c] See https://www.techenhancedlife.com/.

[d] Often young-old adults (65–75) still have aging parents they are taking care of themselves, hence they don't see themselves as an older adult.

The surveillance tools designed to monitor older adults is one example: few would tolerate such overt violations of privacy, but for their children's and society's peace of mind it is widely accepted that it is a good safety technology.

These tools can and do provide some benefit to some older adults. But we argue that in the zeal to "help" older adults one must remain aware of what might make technology unacceptable for them as for other populations. Making older adults suffer through bad technology does harm to them.[7,16,19] If there are consequences to a technology that the (almost always younger) person designing it is unwilling to accept, why should they assume those consequences are acceptable to older adults?

One way of countering this impulse is to move away from designing for "older adults" (with all of the cultural stereotypes this entails) to designing for *the experience of aging*. This shifts the focus from a population in which age demarcations may be cultural, contextual, and, at times, arbitrarily imposed to a focus on the experiences, transitions, and changes that people experience over the lifespan. This stance also helps designers recognize their own journey toward older adulthood, motivating them to design the kinds of technologies that make this life stage enjoyable once they get there.

Conclusion

To help ensure older adults are not disenfranchised by the digital technologies that permeate society, the HCI community will need to move beyond a focus on accessibility as the core design requirement for older adults and consider the myriad other factors that make learning and using digital technologies less appealing for this demographic. Ultimately, by listening to and learning from older adult perspectives, the computing sector is better positioned for designing technologies that not only benefit the current generation of older adults but will ultimately enhance all people's experience of aging.

Acknowledgments

We thank the participants at the CHI 2019 workshop on "HCI and Aging: Beyond Accessibility."[13] The full submissions of workshop participants, on which we based this article, are available at http://mobileage.scc-brutha.lancs.ac.uk/index.php/participants-2/

This work was supported in part by a number of research grants: EU Horizon2020 No. 693319; EPSRC CASE ACCEP/G501327/1; NSF grant IIS-1551574; a Future Fellowship grant from the Australian Research Council (FT170100420); and IBM Open Collaborative Research on Aging in the Workplace.

References

1. ACM Code of Professional Ethics and Conduct. (2018); https://www.acm.org/code-of-ethics
2. Baker, S., Waycott, J., Carrasco, R., Hoang, T. and Vetere, F. Exploring the design of social VR experiences with older adults. In *Proceedings of the 2019 on Designing Interactive Systems Conf.*
3. Berkowsky, R.W., Sharit, J. and Czaja., S.J. Factors predicting decisions about technology adoption among older adults. *Innovation in Aging 1*, 3 (2018).
4. Brandt, E., Binder, T., Malmborg, L. and Sokoler, T. Communities of everyday practice and situated elderliness as an approach to co-design for senior interaction. In *Proceedings of the 22nd Conf. Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction.* ACM, 2010, 400–403.
5. Brewer, R. and Piper, A.M. Tell it like it really is: A case of online content creation and sharing among older adult bloggers. In *Proceedings of the 2016 CHI Conf. Human Factors in Computing Systems.* ACM, 5529–5542.
6. Diaz M., Johnson, I., Lazar, A., Piper, A.M. and Gergle, D. Addressing age-related bias in sentiment analysis. In *Proceedings of the 2018 CHI Conf. Human Factors in Computing Systems.* Article 412, 14 pages.
7. Dotolo, D., Petros, R. and Berridge, C. A hard pill to swallow: Ethical problems of digital medication. *Social work 63*, 4 (2018), 370–372.
8. Foucault, M. The subject and power. *Critical inquiry 8*, 4 (1982), 777–795.
9. Hanson, V.L., Cavender, A. and Trewin, S. Writing about accessibility. *Interactions 22*, 6 (Oct. 2015), 62–65.
10. Harrington, C.N., Wilcox, L., Connelly, K., Rogers, W. and Sanford, J. Designing health and fitness apps with older adults: Examining the value of experience-based co-design. In *Proceedings of the 12th EAI Intern. Conf. Pervasive Computing Technologies for Healthcare.* ACM, 2018, 15–24.
11. Knowles, B. and Hanson, V.L. Older adults' deployment of 'distrust.' *ACM Trans. Computer-Human Interaction 25*, 4 (2018), 21.
12. Knowles, B. and Hanson, V.L. The wisdom of older technology (non-) users. *Commun. ACM 61*, 3 (2018), 72–77.
13. Knowles, B. and Hanson, V.L. Rogers, Y., Piper, A.M., Waycott, J. and Davies, N. HCI and Aging: Beyond Accessibility. In *Extended Abstracts of the 2019 CHI Conf. Human Factors in Computing Systems.*
14. Levy, B.R., Ferrucci, L., Zonderman, A.B., Slade, M.D., Troncoso, J. and Resnick, S.M. A culture–brain link: Negative age stereotypes predict Alzheimer's disease biomarkers. *Psychology and Aging 31*, 1 (2016), 82.
15. Levy, B.R. and Leifheit-Limson, E. The stereotype-matching effect: Greater influence on functioning when age stereotypes correspond to outcomes. *Psychology and Aging 24*, 1 (2009), 230.
16. Lindsay, S., Brittain, K., Jackson, D., Ladha, C., Ladha, K. and Olivier, P. Empathy, participatory design and people with dementia. In *Proceedings of the SIGCHI Conf. Human Factors in Computing Systems.* ACM, 2012, 521–530.
17. *The Medium.* The Truth About Aging in the Tech Industry. (2017); https://medium.com/s/story/aging-in-the-tech-industry-6a0e116bdf09
18. Neutill, R. Why Are So Many People Ready To Let The Elderly Die? (2020); https://www.refinery29.com/en-us/2020/03/9602550/ elder-abuse-neglect-coronavirus-old-people-dying
19. Neven, L. By any means? Questioning the link between gerontechnological innovation and older people's wish to live at home. *Technological Forecasting and Social Change 93* (2015), 32–43.
20. Pullin, G., Treviranus, J., Patel, R. and Higginbotham, J. Designing interaction, voice, and inclusion in AAC research. *Augmentative and Alternative Commun. 33*, 3 (2017), 139–148.
21. Righi, V., Sayago, S. and Blat, J. When we talk about older people in HCI, who are we talking about? Towards a 'turn to community' in the design of technologies for a growing ageing population. *Intern. J. Human-Computer Studies 108* (2017), 15–31.
22. Rogers, Y., Paay, J., Brereton, M., Vaisutis, K.L., Marsden, G., and Vetere, F. Never too old: Engaging retired people inventing the future with MaKey MaKey. In *Proceedings of the 2014 SIGCHI Conf. Human Factors in Computing Systems.* ACM, 3913–3922.
23. Vines, J., Lindsay, S., Pritchard, G.W., Lie, M., Greathead, D. Olivier, P. and Katie Brittain, K. Making family care work: dependence, privacy and remote home monitoring telecare systems. In *Proceedings of the 2013 ACM Intern. Joint Conf. Pervasive and Ubiquitous Computing.* ACM, 607–616.
24. Vines, J., Pritchard, G., Wright, P., Olivier, P. and Brittain, K. An age-old problem: Examining the discourses of ageing in HCI and strategies for future research. *ACM Trans. Computer-Human Interaction 22*, 1 (2015), 2.
25. Wang, S., Bolling, K., Mao, W., Reichstadt, J., Jeste, D., Kim, H-C and Nebeker, C. Technology to support aging in place: Older adults' perspectives. *Healthcare 7* (2019), 60. Multidisciplinary Digital Publishing Institute.
26. Waycott, J., Vetere, F., Pedell, S., Morgans, A., Ozanne, E. and Kulik, L. Not for me: Older adults choosing not to participate in a social isolation intervention. In *Proceedings of the 2016 CHI Conf. Human Factors in Computing Systems.* ACM, 745–757.

For inquiries regarding this article, please contact **Bran Knowles** (b.h.knowles1@lancaster.ac.uk), Senior Lecturer in the Data Science Institute at Lancaster University, England, U.K..

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/aging-accessibility

DOI:10.1145/3419405

PDIs are emerging as alternative sociotechnical infrastructures to enhance flexible work arrangments.

BY MOHAMMAD HOSSEIN JARRAHI, GEMMA NEWLANDS, BRIAN BUTLER, SAIPH SAVAGE, CHRISTOPH LUTZ, MICHAEL DUNN, AND STEVE SAWYER

Flexible Work and Personal Digital Infrastructures

FLEXIBLE, CONTINGENT, OR 'agile,' working arrangements provide workers with greater autonomy over when, where, or how to fulfill their responsibilities. In search of increased productivity and reduced absenteeism, organizations have increasingly turned to flexible work arrangements. Although access to flexible work arrangements is more prevalent among high-skilled workers, in the form of flextime or co-working, the past decade has also witnessed growth of independent contractors, digital nomadism, digitally enabled crowdwork, online freelancing, and on-demand platform labor.[3]

Flexible work arrangements reduce commutes and can enable workers with care-responsibilities to stay in the workforce. Younger workers also see flexibility as a top priority when considering career opportunities.[2]

Flexible working arrangements can also be mutually beneficial, enabling organizations to scale dynamically. Specific skill sets can be immediately accessed by turning to freelancers to fill organizational gaps. A growing number of organizations and workers rely on short-term and project-based relationships, using online platforms such as Upwork or Fiverr to connect.

However, flexible work arrangements often come entwined with precarity cloaked in emancipatory narratives.[5] Fixed salaries and benefits have given way to hourly rates and quantified ratings. Flexible workers often face unpredictability and uncertainty as they carry more risk and responsibility, and are burdened with a great portion of administrative costs (that is, overhead) associated with organizational support systems.[6] Flexible workers at Google, for instance, outnumber full time workers but face far more unpredictability.[11]

Current formulations consider organizations as relatively fixed 'containers', which encapsulate the work performed and the information and communications technology (ICT) systems used to perform it.[12] However, flexible work arrangements take place outside of organizational containers. In this new sociotechnical dynamic, flexible workers interact with a diversity of digital tools that defy centralized, top-down standardization or governance.

We capture this diversity of digital tools through the concept of Personal

» key insights

- **As flexible work arrangements are accepted, work structures, performance expectations, and employee-employer relationships change, presenting both benefits and risks for workers.**
- **Personal digital infrastructures (PDIs) allow workers to realize the benefits of flexible work while avoiding the risks of precarious work.**
- **To avoid the significant negative impacts of increased flexibility, PDI management and design must become more responsive to the needs of flexible workers.**

IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

Digital Infrastructures (PDIs), which denote an individualized assemblage of tools and technologies, such as personal laptops, smartphones, cloud services, and applications brought together by workers to perform their work tasks. Yet, flexible workers constantly reconfigure their PDIs as the technology landscape, client-relationship, and task requirements shift.

For flexible work arrangements to be mutually beneficial, PDI integration in ICT systems for work is increasingly necessary, beyond a narrow focus on enterprise systems supporting standard work.

Our collective research on flexible work arrangements indicates that PDIs present non-trivial challenges, but a more effective design of ICT systems for work can facilitate the integration of these bottom-up infrastructures. The nuanced understanding of PDIs presented here highlights their interplay with flexible work arrangements across key dimensions (spatial, temporal, organizational, and technological) and suggests key priorities for technology and platform developers.

Methods

These findings are based on 170 semi-structured interviews with flexible workers conducted between 2015 and 2019. This number included 11 digital nomads, 37 remote knowledge workers, 51 online freelancers (for example, Upwork and Fivver), and 71 other types of on-demand workers (ride hailing, food delivery, and task work). Participants' average age was 35; 104 were male and 66 were female; 107 resided in the U.S. and 63 in Europe (Norway, Sweden, U.K., and Netherlands), including diverse nationalities and immigrant workers, particularly from India. Interviews were conducted online or in person (by phone or on Skype/WhatsApp/GotoMeeting). Our analysis highlighted a large diversity of tools and technologies for work used by the participants. Examples included digital labor platforms; personal laptops; mobile devices, such as cellphones or tablets; and applications such as Asana, Google Drive, Google Maps, and Zapier, reflecting varying needs.

Flexible Work Dimensions

Even though flexible work environments are becoming more common, our findings reveal a general mismatch between the dynamic requirements of flexible work arrangements and the current technological landscape. Workers often have to go to great lengths to configure PDIs to fit their needs. Designing a more effective PDI necessitates a more nuanced understanding of the requirements of flexible work arrangements.

Not all flexible work arrangements are flexible in the same way. As a useful framework for understanding the intersection of technologies and flexible work, we propose that flexible work arrangements diverge from standard work arrangements along three key dimensions: Organizational attachment (the extent to which workers are under the control of the organization); temporal attachment (the extent to which workers are employed long-term by one organization); and physical attachment (the extent to which workers are in physical proximity to the organization).[1]

Our collective research suggests that flexible work also diverges along a fourth dimension: technological flexibility, referring to the extent to which workers are able to self-curate their own PDI to support their work. These flexibility dimensions are not mutually exclusive and flexible workers often operate across multiple dimensions. The accompanying table summarizes the four dimensions, documenting the role of the current technological landscape in enabling and constraining different dimensions of flexible work environments.

The table is also helpful for understanding the nature of flexibility, since flexible work arrangements render workers less dependent on organizations. However, labels such as remote working or flex timing do not fully capture the complexity of flexible work[4] and hence are a poor basis for the design of PDIs, potentially leading to confusing, even abusive, employer-employee relationships.

Each dimension of work flexibility presents workers with both opportunities for and challenges to their autonomy, efficiency, and effectiveness. It is these opportunities that PDI technologies must magnify and mitigate.

Spatial flexibility refers to the extent to which workers can detach themselves from specific locations and workspaces.

Opportunities. The ubiquity of networked infrastructures allows flexible workers to be increasingly mobile. Modern norms of digital communication have created an environment where workers can be reached just as quickly half a world away as they can be in the next office. Our research participants who worked primarily online could 'get to work' from wherever they were, though sometimes bound by geo-restrictions in terms of which tasks they could fulfill. Spatial flexibility was characterized by being able to work from home, but participants

Different dimensions of flexible work environments.

Dimension of flexibility	Definition	Common examples of work arrangements presenting flexibility dimension	Examples of supporting digital technologies	Examples of technological constraints
Spatial flexibility	The extent to which workers can detach themselves from specific locations and workspaces	Nomadic work	▸ Portable computational equipment ▸ Non geo-restricted access to systems ▸ Adequately reliable and affordable Internet connectivity ▸ Access to charging stations and/or long battery life	▸ Fixed computational equipment ▸ Geo-restricted access to systems ▸ Lack of access to reliable or affordable Internet connectivity ▸ Lack of access to charging stations and/or low battery life
Temporal flexibility	The extent to which workers can detach themselves from specific work schedules	Temporary work; Part-time work; Flextime	▸ Complex time- and task-management systems ▸ Personal cloud services (such as Google drive) ▸ Asynchronous communication platforms and norms	▸ Blurring of work-life boundaries ▸ Digital distractions ▸ Inflexible time- and task-management systems
Organizational flexibility	The extent to which workers can detach themselves from organizations' administrative control	Gig work; Contract work; Freelance work	▸ Digital labor platforms ▸ Bespoke employment/engagement contract ▸ Digital accounting mechanisms ▸ Community-developed add-ons and plug-ins (for example, scripts)	▸ Policies restricting the external use of enterprise systems ▸ Technical management norms
Technological flexibility	The extent to which workers can self-curate the infrastructure that supports their work	All types of flexible work arrangements	▸ Ownership of personal IT (for example, personal devices and cloud) ▸ Systems that operate across platforms and devices	▸ Lack of interoperability of enterprise applications, task management software, and file formats

would still utilize extra-domestic spaces, such as co-working spaces, hotel rooms, and coffee shops. More pervasive cellular network coverage also contributed to the possibility of working remotely or on the move. Several participants had embraced this opportunity, becoming 'nomadic,' traveling long distances, and even setting themselves up wherever a stable Internet connection was available. Self-identifying global digital nomads are the best examples of high spatial freedom unleashed by digital connectivity.

Challenges. Spatial flexibility is challenging as workers have to constantly navigate and prepare for the unpredictability of new and changing work environments. For example, spatial flexibility is often stymied by the lack of an adequate or reliable Internet connection or charging station. Although the stereotype of the coffeeshop nomad holds true, workers face potentially high and unforeseen overhead costs in negotiating continued access to workspaces and essential information infrastructure. One participant noted: "The biggest kind of uncertainty is that I can't guarantee there will be a strong connection when I do go to coffee shops." The cost of co-working spaces can eat up much of the financial profit gained from remote work. As a result, for high-intensity digital work, such as semantic sequencing or video editing, most profits go to workers with stable and highly ergonomic home-office set-ups rather than those working remotely and using mobile devices.[9]

Temporal flexibility refers to the extent to which workers can detach themselves from specific work schedules.

Opportunities. Temporal flexibility ranges from workers setting their 'working hours' more flexibly within a defined set of parameters (that is, flextime), to workers having complete freedom in choosing when and how long to work (that is, creative freelance work). In both cases, digital task and time management systems aided this temporal flexibility among our participants, who employed a variety of tools in parallel to manage fluid temporal work rhythms. Communication platforms such as Slack afforded temporal flexibility in communicating with peers, asynchronously and across time zones. Personal cloud services, such as Google Drive or Dropbox, also provided the flexibility to create, access, and manipulate information across time. Those affiliated with larger organizations also used independent cloud services as the shared repository of asynchronous collaboration, in tandem with enterprise resources.

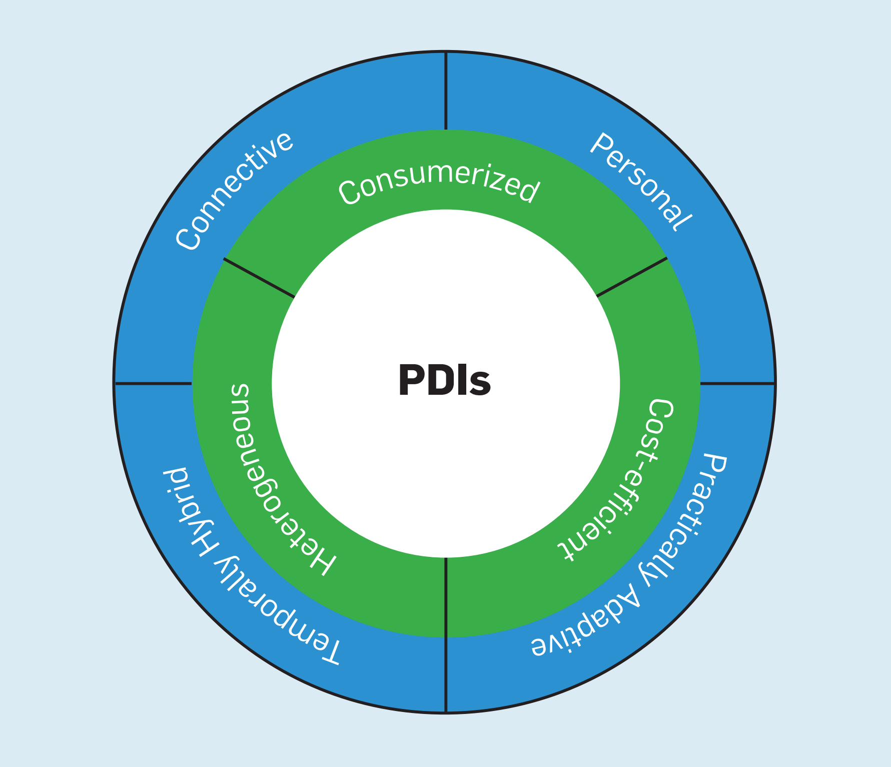
Our findings reveal a general mismatch between the dynamic requirements of flexible work arrangements and the current technological landscape.

Challenges. Temporal flexibility requires high trust levels from an organization and/or high autonomy for workers. Temporal flexibility is also relatively incompatible with traditional micro-management styles or with time-sensitive tasks. One of the most prominent challenges that our participants faced was the conflation of personal and work times. A participant puts this succinctly: "When you're in your office, people assume you're in the office, you're available. You're out of the office, maybe not as available. And now technology's made [us] available 24/7. People think, 'Wait a minute, I sent you an email. Why didn't you respond?'" Mobile technologies have rendered the boundary between the two spheres even less distinct; being able to work at any time increases pressure to be 'always on' and always available to respond to messages. Communicative affordances, such as read receipts or the 'seen' function in messaging, mean that workers face a pressure to respond immediately.

Organizational flexibility refers to the extent to which workers can detach themselves from organizations' administrative control.

Opportunities. Along with changing norms of work, such as project-centrism, flexible workers can find and execute projects on a global scale by using digital labor marketplaces facilitated by online platforms. Online labor platforms provide key facilitators of organizational flexibility through mechanisms such as digital escrow, digital accounting software, and digital contracting. Through manual selection or more complicated algorithmic matchmaking mechanisms, these platforms lower transaction costs for the service recipient (increasingly an enterprise) and the worker. More open platforms, such as Upwork, enable workers to pick and choose clients based on their own preferences. Several of our research participants left stan-

Figure 1. Technological and social layers of personal digital infrastructures (PDIs).



dard work arrangements and became dedicated freelancers because they saw the diversity of projects and tasks offered by online freelancing platforms as a source of learning and raising social capital. Without a formal employment contract, workers can engage in multiple projects and organizations can decide which contracts they want to take. For instance, one of our U.S.-based participants forwent full-time employment, a steady revenue stream, and health insurance benefits (as a cancer survivor) because he found working on a large number of system administration projects from different organizations to be a more fulfilling learning experience than being attached to a single firm with less diverse technical challenges.

Challenges. On the other hand, 'gig economy' platforms, such as Uber Eats and Deliveroo, provide algorithmic matching mechanisms between clients and workers but offer limited choice over which micro-contracts to take and limited organizational information (such as total length of delivery) to enable workers to choose their tasks more profitably. Indeed, even though flexible workers may enjoy a higher administrative flexibility, they may find their work to be fraught with different restrictive policies or requirements set by the organizations. For example, a participant described his work laptop as "locked down" as he "can't use any type of Google platform, can't use Skype, and can't use any open source, because it's [considered to have] security issues" by the employer. Along the same line, several participants noted how installing applications on the work laptop, or a smartphone, is not possible without going through a tedious bureaucratic process. Therefore, workers in these settings can usually be subjected to the restricted work systems imposed from above. Without careful design, the same systems that enable greater flexibility can also be used to increase technical managerial control and restrict worker autonomy.[7]

Technological flexibility refers to the extent to which workers can self-curate the infrastructures that support their work. Technological flexibility represents the convergence of multiple technological paradigms, such as consumerization; the proliferation of smart mobile devices; and the platform economy.

Opportunities. Our research makes it clear that flexible work is largely enabled by technological flexibility. Personal digital tools have penetrated the workplace, and many of our participants enjoyed a high level of flexibility in bringing their own devices to work, a trend which is captured through Bring Your Own Device (BYOD) programs and IT individualization. The ability to select their own work tools, rather than being chosen and imposed on by the employer, helps workers tailor PDIs to their own needs and dynamic work environment.

Challenges. On the flip side, the diversity of tools used by flexible workers can result in a lack of interoperability between various platforms, systems, and file types. Whereas Apple MacBooks are preferred tools among workers with creative and design tasks, their lack of interoperability with Windows-based systems and software creates many problems. Even small details, such as missing fonts or graphic packages, can create adversarial scenarios, lost income, and client dissatisfaction for these workers. Several of the research participants who use Gmail, for instance, do not want to integrate Google Drive for cloud storage. A participant noted, "I use Google Drive, mostly because Google kinda forces you to use Google Drive." Another lamented, "I'm having trouble with making all the technologies work. Google wants to take over. It wants Google Calendar to be your calendar." Since platform organizations impose their dominance, cross-platform coordination becomes difficult for workers who wish to take advantage of technological flexibility.

Characteristics of PDIs

PDIs are strategies employed by flexible workers to realize the opportunities and mitigate the risks that come with flexible work arrangements. Workers, whether flexible or not, will selectively use some digital tools and devices more than others, configuring these sociotechnical systems to support largely individual, creative, operationally resilient, and problem-driven work.[10] Next, we discuss the characteristics of PDIs which enable them to play this complex, enabling role for flexible work.

PDI technological layer: Heterogeneous, consumerized, and cost-efficient. PDIs build on fundamental technical characteristics of heterogeneity, low-fixed cost, and technology consumerization to create the conditions for realizing the benefits of flexible work arrangements (as illustrated in Figure 1). To achieve technological flexibility, PDIs are heterogeneous and involve ensembles of personal, consumer-based devices; end-user tools; digital platforms; and ubiquitous infrastructures (for example, local Wi-Fi networks). Yet, such digital technologies are often not owned by an organization, even if the worker is affiliated with a larger organization. Rather, our research participants built on what is available in the consumer technology market to remain versatile and retain control over their work parameters.

The cost of purchasing and maintaining such devices, however, falls on the worker and can generate problems in instances of interoperability and reduced device security. Our participants were cognizant of these costs and had to find strategies to keep them under control. As a freelance journalist, one participant managed to use Dropbox for free (beyond the normal capacity of free accounts): "through absolute pure stinginess to avoid paying for Dropbox, I do everything they offer to keep bumping up my limit, and the latest thing was if you store your photos on Dropbox, we'll give you an extra 3 gigs. So I said, 'Sure.'"

PDI social layer. Building on the foundational technologies, PDIs are connective, adaptive, and temporally hybrid sociotechnical systems, reflecting and reinforcing multiple dimensions of flexibility.

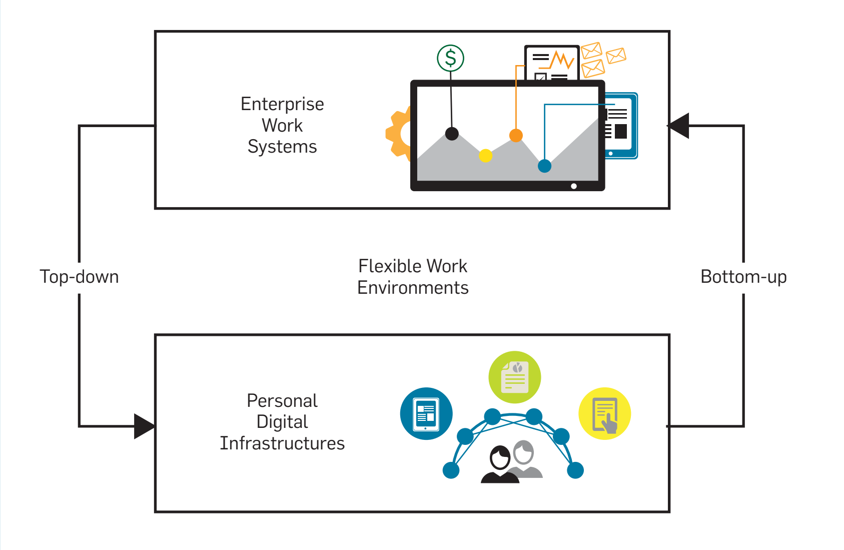
Connective. While the heterogeneity of digital technologies enables workers to adapt to the diverse needs of flexible work environments, this diversity simultaneously creates a key challenge: lack of interconnection and interoperability among various technologies and competing consumer-based ecosystems (for example, Microsoft vs. Apple). Therefore, to effectively support work practices that often stretch multiple tools, PDIs connect various ecosystems and enterprise information systems, often through gateway practices (activities that bring together competing systems), VPNs, or integration tools such as Zapier or IFTTT. Some of this work is done manually, requiring extra work on the part of the workers. For example, one participant receives new legal cases from clients through an organizationally sanctioned document management service called NetDocuments. However, to use his preferred cloud-based document management system (Box.com), he downloads and manually uploads each case separately.

Personal. Flexible workers often enjoy higher organizational flexibility and therefore assemble collections of digital resources from personal, public, and corporate elements based primarily on personal preferences. PDIs reflect personal and specific work situations, and the worker is personally responsible for making these collections of technology function. As such, our participants may dedicate great effort to maintaining PDIs and must rely on personal learning and development rather than receiving dedicated training.[3] This often requires a great deal of experimentation and situated learning. A Web developer in our sample, for instance, figured out through trial and error that he could leverage inflight Wi-Fi without having to pay for it. He is able to develop applications while his computer can still communicate with the client company's development server.

Temporally hybrid. Due to temporal flexibility, PDIs often span the personal and professional lives of flexible workers. Using personal devices and tools for work may often result in the intrusion of work contacts and projects into personal IT systems. For most participants, it was difficult to demarcate between tools and services that define and support personal and work-related uses. For instance, participants used social media messaging systems, such as WhatsApp, to communicate with friends, family, clients, and former clients. This has changed the temporal rhythm of work and further blurred the line between personal time and work. In response to these challenges, some participants have adopted specific strategies and tools to impose boundaries. They may use time management tools and offline working hours to demarcate between work and personal life, or to avoid digital distractions. For example, some had to clarify to their collaborators and clients that they would not reply to email messages after a certain time, even though they are on a flexible work schedule. A participant clearly communicated to work-related contacts: "I never check my email after 6 PM" or another has told clients when she is traveling, "I'm not available. I won't be responding."

Practically adaptive. PDIs are organizationally adaptive. While operating in a liminal space between different organizations and projects, participants often must accommodate the differing, sometimes contrasting,

Figure 2. Flexible work environment shaped at the interplay between work systems and PDIs.



technological requirements of multiple client organizations, projects, and collaborators. These workers are often cognizant of organizational constraints as they directly impact their technology practices, and they make sure their PDIs also connect with others to support collaborative information sharing, serving not just as individual resources but also collective infrastructures. A couple of participants highlighted restricted access to clients' enterprise information based only on specific IP addresses (in clients' offices or predetermined locations). They, however, often used workarounds, such as emailing the documents to themselves so they had the flexibility to work on information resources outside of the designated locations.

PDIs are also locally adaptive. Flexible workers may work from different places or even on the move. Therefore, an awareness of local infrastructures enables workers to ensure digital connectivity, which is a central element of digital work. Spatial mobility may sometimes require physical effort and planning for technological use across different spaces. For example, a highly mobile worker in our sample leveraged Wi-Fi analyzer applications to gauge the available networks in a neighborhood and assess their relative signal strength before choosing a public place to work. Others would carry assemblages of devices, such as external batteries, a power splitter, or USB-powered firewall (that provides a secure use of Wi-Fi), to create reliable mobile digital offices in different locations.

Implications

In what follows, we detail ways that organizational system design and management can help redress the precarity of flexible work by reinforcing benefits of flexible work and ameliorating its challenges. In doing so, we emphasize the need for better integration and facilitation of PDIs. This helps organizations, workers, and platforms navigate the consequences of flexibility and better support the PDIs that underlie effective and sustainable contingent work arrangements.

Integration of PDIs in work systems. Flexible workers, across different capacities, must dynamically relate their PDIs with traditional managerial structures and processes, such as allocating, evaluating, and coordinating work. Since PDIs are assembled by workers in a bottom-up fashion, replication of traditional expectations would curb key dimensions of flexibility. Rather, organizations must meet their flexible workers halfway, by tolerating and facilitating technological diversity. Organizations need to identify priority points where technological cohesion among the workforce is essential, such as enforcing universally readable file types or requiring certain smartphone operating systems. On-demand delivery platforms, for instance, require that workers' smartphones have functional GPS, sufficient mobile data and battery, and can support the latest worker-facing app updates.[8] On the other hand, organizations must also identify where technological cohesion is not essential but merely desirable, since our findings indicate that workers who are not provided with enough technological flexibility may resort to tedious workarounds or even sabotaging formal work systems.

Flexible work takes place in a hybrid space shaped by both work systems and PDIs (see Figure 2). Flexibility inexorably creates complexity, and a higher need for negotiation and transparency. In an optimal approach, employers and workers negotiate the top-down influence of work systems against the bottom-up force of PDIs. Through these negotiations, organizations can assure their goals are fulfilled and workers can meet their needs. PDIs enable workers to draw on systems that can be generative to diverse and flexible uses. Platform designers and managers of systems for work need to recognize PDIs as infrastructure of flexible work. As workers strive to bring in their own personal technologies, firms will seek to balance these uses against the need for an integrated and secure system that is the backbone of organizational processes and meets regulatory and compliance rules. Via negotiations, expectations of both parties should be made clear. For example, workers need to know the boundaries for flexibility and non-negotiable areas so they can act upon it in enacting PDIs.

Tools that facilitate a more cohesive

Tools that facilitate a more cohesive integration of PDIs into work systems help both workers and organizations control and manage projects while relating personal, flexible routines.

integration of PDIs into work systems help both workers and organizations control and manage projects while relating personal flexible routines. Integrative management platforms, for example, can help flexible workers smoothly navigate and work across personal data and enterprise resources. Such a platform provides versatile privacy configurations by dynamically learning what data should be shared with the organization for effectively managing work projects or kept on the worker's personal storage systems (locally or personal clouds) to provide the worker certain freedoms and autonomy.

Beyond the integration of flexible PDIs, enterprises must also facilitate flexible participation. Even though flexible workers occupy a dynamic relationship with enterprises and enterprise work systems, many of these systems are designed for standard, full-time employees, and evidence still points to the invisibility of this workforce.[11] The design of these systems must be mindful of a contingent, agile workforce that can scale up on demand and dynamically facilitate plug-and-play-type participation (for example, connecting with or disconnecting from certain enterprise resources). In addition, these systems must provide greater flexibility for remote, flexible access, something that has become even more paramount during the COVID-19 pandemic.

Facilitating construction and uses of PDIs by flexible workers. Flexible work is a largely independent pursuit and constructing complex PDIs is often done by each individual worker. However, community support and collective learning can complement centrally provisioned organizational support (for example, help desk support). Organizations can expand the scope of support toward a hybrid model by encouraging community-based support, which works in concert with formal IT support. Traditional firms, as well as digital on-demand platforms, can contribute to building collaborative structures through which workers help each other. Furthermore, the design and management of systems can promulgate community-based learning, which stands in contrast to the implicit design of many on-demand platforms that discourage workers' community-building activities. Flexible workers can greatly benefit from connecting with other workers who go through similar challenges (for example, securing the most profitable hits on Amazon Mechanical Turk or determining the most effective ways to present skills on Upwork profiles). One example of this type of community-based system is turkopticon,[a] a browser plug-in that enables MTurkers to share reviews of individual employers with each other.

Another key challenge to PDIs is spatial constraints. Workers often must go to great lengths to make PDIs locally adaptive. Spatial flexibility often requires workers to grapple with spatial constraints, such as a lack of access to information, centrally held tools, or the need to navigate multiple contextual barriers that stem from their work over unfamiliar territories. System design therefore needs to be mobility-sensitive and strive to mitigate these challenges in the creation and use of PDIs. One example would be a Firewall and VPN that provides secure Wi-Fi connections in public locations. Non-technological strategies that facilitate mobilizing the workforce can focus on providing local resources for more geographically mobile workers by, for example, partnering with local co-working spaces across different cities to ensure productive work environments and reliable infrastructural access.

Conclusion

PDIs are of growing importance to all workers, but especially those who must adopt and adapt practices to enable multi-axial modes of flexibility. The state of research and practice relative to the design and management of ICT for work largely focuses on one of the two extremes—either the organizationally embedded work technologies or the individually used consumer technologies. Addressing the needs of flexible workers and organizations using flexible work arrangements will necessarily require research, development, and deployment of PDIs to bring these two models together in a way that helps all parties realize the opportunities and challenges of flexibility. To avoid exploitative forms of precarious work, PDIs must provide adequate benefits for employees and employers while mitigating the risks. Yet, by helping organizations and workers navigate the conflicting consequences of flexibility, the design and management of digital infrastructure can support the emergence of new, effective, sustainable work arrangements and PDIs that undergird these arrangements. C

[a] https://turkopticon.ucsd.edu/

References

1. Ashford, S.J., George, E., and Blatt, R. Old assumptions, new work: The opportunities and challenges of research on nonstandard employment. *Acad. Manag. Ann. 1* (2007), 65–117.
2. Burnford, J. Flexible working: The way of the future. *Forbes*, 2019; https://www.forbes.com/sites/joyburnford/2019/05/28/flexible-working-the-way-of-the-future/#609f10ee4874.
3. Hagel, J., Schwartz, J., and Bersin, J. Navigating the future of work: Can we point business, workers, and social institutions in the same direction? *Deloitte Review* (2017).
4. Hemsley, J., Erickson, I., Jarrahi, M.H., and Karami, A. Digital nomads, coworking, and other expressions of mobile work on Twitter. *First Monday* (2020); doi:10.5210/fm.v25i3.10246.
5. Jarrahi, M.H., Sutherland, W., Nelson, S.B., and Sawyer, S. Platformic management, boundary resources for gig work, and worker autonomy. *Comput. Support. Coop. Work* (2019), 1–37.
6. Kalleberg, A.L. and Vallas, S.P. *Precarious Work: Causes, Characteristics, and Consequences.* Emerald, Bingley, U.K., (2018).
7. Kraemer, K.L. and King, J.L. Computer-based systems for cooperative work and group decision making. *ACM Computing Surveys 20* (1988), 115–146.
8. Newlands, G. Algorithmic surveillance in the gig economy: The organisation of work through Lefebvrian conceived space. *Organization Studies* (2020); doi:10.1177/0170840620937900.
9. Newlands, G. and Lutz, C. Crowdwork and the mobile underclass: Barriers to participation in India and the United States. *New Media & Society* (2020); doi:10.1177/1461444820901847.
10. Sawyer, S., Crowston, K., and Wigand, R.T. Digital assemblages: Evidence and theorising from the computerisation of the US residential real estate industry. *New Technology, Work and Employment 29*, (2014), 40–56.
11. Wakabayashi, D. Google's shadow work force: Temps who outnumber full-time employees. *NY Times*; https://www.nytimes.com/2019/05/28/technology/google-temp-workers.html.
12. Winter, S., Berente, N., Howison, J., and Butler, B. Beyond the organizational 'container': Conceptualizing 21st century sociotechnical work. *Information and Organization 24* (2014), 250–269.

Mohammad Hossein Jarrahi (jarrahi@email.unc.edu) is an associate professor at the University of North Carolina, Chapel Hill, NC, USA.

Gemma Newlands is a Ph.D. candidate at BI Norwegian Business School, Oslo, Norway.

Brian Butler is a professor at the University of Maryland, College Park, MD, USA.

Saiph Savage is an assistant professor at Northeastern University, Boston, MA, USA.

Christoph Lutz is an associate professor at BI Norwegian Business School, Oslo, Norway.

Michael Dunn is an assistant professor at Skidmore College, Saratoga Springs, NY, USA.

Steve Sawyer is a professor at Syracuse University, Syracuse, NY, USA.

Attention: Undergraduate *and* Graduate Computing Students

STUDENT RESEARCH COMPETITION

acm

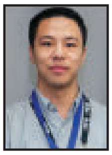
Association for Computing Machinery
Advancing Computing as a Science & Profession

SPONSORED BY Microsoft

The ACM Student Research Competition (SRC), sponsored by Microsoft, offers a unique forum for undergraduate and graduate students to present their original research before a panel of judges and attendees at well-known ACM-sponsored and co-sponsored conferences. The SRC is an internationally recognized venue enabling undergraduate and graduate students to earn many tangible and intangible rewards from participating:

- **Awards:** cash prizes, medals, and ACM student memberships
- **Prestige:** Grand Finalists and their advisors are invited to the Annual ACM Awards Banquet, where they are recognized for their accomplishments
- **Visibility:** opportunities to meet with researchers in their field of interest and make important connections
- **Experience:** opportunities to sharpen communication, visual, organizational, and presentation skills in preparation for the SRC experience

It's hard to put the ACM Student Research Competition experience into words, but we'll try. . .



"It is an excellent chance for me to participate in the ACM SRC to present my work, and I got much valuable feedback from experts and peers. Moreover, I practiced my communication and presenting skills, laying a solid foundation for my future research life. I also made friends with outstanding students from all over the world who are interested in scientific research. I strongly recommend ACM SRC to all undergraduate and graduate students because it is a perfect platform for all of us to exchange ideas and make improvements."

Shengcheng Yu
Nanjing University | ASE 2019

"ACM SRC provided an excellent opportunity to showcase my research work and solicit feedback from experts in my field. Winning this competition gave a boost to my confidence and motivated me to make a greater impact to my field. In addition, I enjoyed learning about other research areas from fellow participants and made several connections. Thus, I am extremely grateful for the organizers of this competition who provided me an opportunity to present my work. I would strongly encourage undergraduate and graduate students to participate!"

Peter Zhi Xuan Li
Massachusetts Institute of Technology | SIGMICRO 2019

"The SRC was a great opportunity to present our work-in-progress research results to leading experts in the community. I appreciated the different kinds of presentations – extended abstract, poster, and talk – which gave me the opportunity of getting different kinds of feedback. All in all, the SRC was a great experience to both make new connections and discuss early-stage results with expert researchers."

Ari Rasch
University of Muenster | CGO 2020

ACM Student Research Competition

STUDENT RESEARCH COMPETITION

Association for Computing Machinery
Advancing Computing as a Science & Profession

SPONSORED BY Microsoft

"Participating in the ACM SRC was a phenomenal experience for me. As an undergraduate student, it gave me the opportunity to present my work to experts in the field and seek valuable feedback. Interacting with researchers and fellow participants at the conference was a great learning experience and allowed me to network as well. The SRC is a great way to broaden one's horizons and I would strongly encourage student researchers to participate in it."

Milind Srivastava
Indian Institute of Technology Madras | ICCAD 2019

"Participating in the ACM SRC gave me confidence in my research and technical experiences. Receiving feedback during my presentation helped me improve my work and develop my communication skills. It was a fantastic opportunity to present in English at an international conference for the first time and to connect with other researchers during the competition. I really admire the program because it gives opportunity and support so that anyone worldwide can apply and attend the conference. I strongly recommend other undergraduate students submit their research to the competition."

Ana Solórzano
Universidade Federal de Santa Maria, Brazil | GHC 2019

"ACM SRC was a high-visibility platform for sharing research ideas. It was exciting to meet people who share similar interests and get their perspective and thoughts on ideas we presented. We got exposed to the best collection of research work, arts, and emerging technologies. It would be really difficult for our team to recall an instance where we got bored. Rather, the challenge was to choose between multiple, equally enticing events. My heartfelt thanks to my team and ACM for organizing this platform!"

Sai Ganesh
Texas A&M University, College Station | SIGGRAPH 2019

"The ACM SRC made it possible for me to experience the world of research in my field of interest, far more broadly than I had previously been able to from my home institution. At the conference, I was able not only to hear from and talk to prominent researchers and fellow students, but also to better understand my possible future career paths as I prepared to apply to graduate schools. I am incredibly grateful to my mentors for guiding me to participate in this opportunity. If you're a student interested in research, I highly recommend you apply!"

Samuel Estep
Liberty University | SPLASH 2019

Check the SRC Submission Dates: ***https://src.acm.org/submissions***

Questions? Contact Nanette Hernandez, ACM's SRC Coordinator: ***hernandez@hq.acm.org***

DOI:10.1145/3440752

Conversing about places with a computer poses a range of challenges to current AI.

BY STEPHAN WINTER, TIMOTHY BALDWIN, MARTIN TOMKO, JOCHEN RENZ, WERNER KUHN, AND MARIA VASARDANI

Spatial Concepts in the Conversation With a Computer

HUMAN INTERACTIONS WITH the physical environment are often mediated through information services, and sometimes depend on them. These human interactions with their environment relate to a range of scales,[28] in the scenario here from the "west of the city" to the "back of the store," or beyond the scenario to "the cat is under the sofa." These interactions go far beyond references to places that are recorded in geographic gazetteers,[37] both in scale (the place where the cat is) and conceptualization (the place that forms the west of the city[29]), or that fit to the classical coordinate-based representations of digital maps. And yet, these kinds of services have to use such digital representations of environments, such as digital maps, building information models, knowledge bases, or just text/documents. Also, their abilities to interact are limited to either fusing with the environment,[44] or using media such as maps, photos, augmented reality, or voice. These interactions also happen in a vast range of real-world contexts, or in situ, in which conversation partners typically adapt their conversational strategies to their interlocutor, based on mutual information, activities, and the shared situation.[2] Verbal information sharing and conversations about places may also be more suitable when visual communication through maps or imagery is inaccessible, distracting, or irrelevant, such as when navigating in a familiar shopping mall.

Much of human conversation is about places without well-known names or with only local significance, such as *the cinemas* (see the scenario). They are often identified not by names at all, but by qualities (*the rear car park*; more examples are depicted in the accompanying table). Sometimes, places lack a well-established location or extent, such as *the West of the city*, or are characterized in their location just by spatial relations (*the elevators at the cinema*).

Thus, we highlight here to broader concept of *place* we are confronted

» key insights

- **Conversing about places challenges machine-understanding due to the flexibility, ambiguity, and context-dependency of place references, and the vagueness of the nature of places.**
- **Conversing about places assumes an understanding of the structural and hierarchical knowledge of the world in order to use and interpret qualitative spatial prepositions.**
- **Yet, discourse patterns about places can be modeled or learned, and translated into formal interactive conversation patterns, thus supporting the development of intelligent assistance systems.**

SCENARIO

Kim, who is shopping for comfortable shoes, turns to her intelligent conversational agent Marvin.

Kim, looking at an advertisement:
Where can I buy these sneakers? It needs to be a physical store because I'm not sure of my size in that brand.
Marvin:
Hype in the Imperial Shopping Centre, in the west of the city, currently has the best price.
OK, let's drive there.
She connects Marvin to a self-driving vehicle …
The main car park is full.
OK, then take me to the rear one, behind the cinemas. But I've never entered the shopping center from the back.

Compare this conversation to the maps or to the step-by-step route descriptions provided by current location-based services: this conversation is concise, situated, and personalized. More importantly, it is a conversation about 'where' that does not rely on well-known toponyms, points-of-interest, or on coordinates. It uses *places* and *their relations*. This is precisely what current AI is not capable of.

IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

with. Borrowing from the core concepts,[24] we say that every object (in the core concepts) is located. And places are objects.[32] To be more precise, references to objects make them places if the conversation is about localization—in contrast to other object properties. But localization of an object can happen in a variety of ways, and geometric localization—that is, in a spatial reference frame—is only one of them. Another one is the one by qualitative spatial relations. In this article we focus on the latter.

But while vital for human information sharing, decision-making, and coordination, conversations about places are still poorly supported by information services and situated robotic assistants such as self-driving vehicles, or, in our scenario, by Marvin. Typical conversational contexts about places include:

- Partners establishing a shared situation awareness, such as in command and control[10] for emergency response (*Point the firehose more to the right*), in defense (*The sniper must be on the roof of the building next to the tree*), and in collaborative scenarios in games.
- Conversational location-aware services used as recommender systems (*Where can I buy these sneakers?*) or self-driving vehicles (*Go to the car park behind the cinemas*).
- Services that obfuscate or abandon geometric descriptions in favor of vague and qualitative descriptions to protect private or sensitive information (*Kim lives in the city's East* is protecting the exact home address by referring to a vague and coarser place).

The wide-ranging opportunities to engage users in a verbal information exchange when seeking and providing place information extend beyond pure information seeking, and include commands as well the provision of explanations to machines. Here, we outline the

Illustrations of additional complexities in references to places.

Example	Places referred to through …	Scale
under the sofa	a qualitative spatial preposition with a located object	hyperlocal
This place to sit down	an affordance facet	hyperlocal
The beautiful beach	a quality facet and a generic noun	local
The West of the city	a vague and ambiguous place description	city
The capital of Slovakia	through a function of a place within another gazetted place	city
The Newer Volcanic Province	a vague region with a gazetted placename	regional
River crossing Austria and Slovakia	a qualitative spatial relation with gazetted placenames	country

underlying challenges of conversing about places with computers, and report on recent progress in this space. We also identify salient areas for immediate and longer-term future work.

Scope and Challenges

Communication about our physical environment—small scale and large scale—is special and difficult: the content of the communication strongly depends on the personal perspectives and preferences of the individuals engaged in the conversation, and may contain high levels of ambiguity that can only be resolved via mutual information and local context. Yet, this communication needs to enable physical interaction with the environment and others in it, such as coordination or planning of movement.[5] In conversations with a computer, references to both places and their relations are difficult to parse and represent formally for a variety of reasons, among them the inherent flexibility, vagueness, and ambiguity of language (see the table), the context-dependency impacting on language grounding and disambiguation, and the interpretation of *part-of* relations (scale) and other spatial relations.

Language and facets of place. When conversing about places, the expressions used in linguistic strategies convey place qualities (*the beautiful beach*), relations between places (*the river crossing Austria and Slovakia*) and relations between people and places, or roles of places. Such functional, social, and operational roles of places are imposed by individuals, or groups. A function of a place can be tied to what it affords to an individual. The social role is derived from ownership. And the operational role is derived from intention (and spatial configuration). A recent comprehensive review,[15] using facet theory,[3] categorized these means of expression into three top-level categories of facets of places. *Anthropocentric* facets relate to the relations between people and places. They include the affordances of places and activities linked to them, as well as emotive facets that cover the sense of place and cultural attachment. *Geographic* facets describe the spatial and physical properties of places. Spatial facets enable localization of a place in space and describe the meaning of the place in relation to space and other places, while physical facets capture the structure, material, or form of a place. And finally, *linguistic* facets present a distinct category, capturing purely linguistic manifestations of place through place-names and identifiers. Linguistic facets can pick up anthropocentric facets (*the sneaker store*) or geographic facets (*the Lower Eastside*), but do not have to (*Hype*).

While the bulk of research and development in assistance systems thus far has focused only on linguistic facets (for example, *place names*), it falls short in the interpretation of references to place in natural language conversations. Situated assistants that are able to interact with the physical environment such as self-driving vehicles will require richer support for multifaceted natural language interaction about places. In this direction points already work in controlling robots by spatial references, for example identifying objects in an environment that can be referred to on the basis of a range of spatial reference systems,[30,41] or reflecting on deductive human reasoning with preferred models instead of formal qualitative spatial reasoning.[34]

Vagueness. References to places are inherently vague in their specification of a location.[32] Since place descriptions are usually *referring expressions*—expressions *uniquely* identifying a referent in a set of possibilities—they are typically as specific as necessary.[40] This specificity is given not by coordinates—that is, neither by crisp nor by fuzzy geometric descriptors (for example, Derungs et al.[8])—but by contrast.[45] For example, the location of the *West of the city*[29,33] is sufficiently described as in contrast to the East, South, or North, and *the back of the store* is sufficiently specified in contrast to its front. Vagueness also applies to the qualities of places (*the full car park*), and to the relations between places (*elevators at the cinema*), each with an intent to be only as distinctive as necessary to be relevant in the given context.

Ambiguity. Most references to places are ambiguous. For example, while country names are globally unique, the same is already no longer true for city names (*Paris*) or landscape names (*Alps*). *The sneakers shop on the second floor* is not a globally unique reference. Disambiguation requires further locative discriminators, which are enabling the popular cluster-based disambiguation, further non-locative discriminators, which are enabling iterative identification such as through dialogue, and references to spatial relations, which are enabling disambiguation by spatial reasoning. Popular spatial relations are containment hierarchies, emphasizing the importance of scale (*the sneakers shop in the Imperial Shopping Centre* [*in this city, in this country*]), but may also be a relation with a local landmark (*the sneakers shop near the food court*). Dialog-based disambiguation methods[46] still require intense research, as they flip the roles of the computer and the human user, in that the human user becomes the source of information provided to the computer. Verbal disambiguation by all three approaches is easier for a situated than for a non-situated conversational agent.

Scale, or containment, is an intrinsic property of places that is impacting on spatial reasoning and enabling to adapt to spatial context. The question *Where can I buy those sneakers* can be answered correctly but of varying relevance by *in the West of the city* (for example, telling Kim the general direction), *at the shopping mall* (for example, if Kim is in the neighborhood), *on the second floor* (if Kim is in the car park), or by *in the back of the shop* (if Kim is entering the shop). These descriptions range between scales, and benefit from containment hierarchies that allow zooming and reasoning between levels of granularity.[36]

Context. Conversational context provides implicit information taken into account when generating or interpreting place references. While this elementary function of context is well understood and undisputed, the concept of context itself is still ill-defined.[9] Context is broader than the immediate perceptual cues of a situated agent, such as positioning,[39] orientation and vista analysis for meaningful salient features. It includes a consideration of the human communication partner, of the physical environment beyond the vista, and of social and technical aspects.[19] Despite of this common broader understanding, and in tacit recognition of the ill-defined challenge, existing research on context-aware computing often links context only to position or any semantically richer notion of location. For example, some work matches a position with points-of-interest and the activities they afford (for example, Horvitz et al.[16] and Schilit et al.[38]), and on the smaller scales of smart environments sensors are used for activity recognition, which links to location,[1] or a broader notion of location is introduced to deal with positioning uncertainty in these smart environments.[4] Semantically richer notions of location can enable assuming semantically meaningful places, including their relative description by selecting *salient* facets.[36] It also includes the choice of a degree of specificity that is *sufficient* for the communication purpose. It requires selection of *relevant* facets to avoid unnecessary ambiguity. And it requires a choice of scale that *relates* to the current decision making. Human conversation partners naturally adapt to context even when they are physically removed (for example, speaking on the phone, a form of co-presence[47]).

Descriptions can be personalized for familiarity (in contrast to Kim, a newcomer to the shopping mall would also need instructions on how to get to the second floor); ability (a person in a wheelchair might require nuanced instructions on how to access the second floor); role (a cleaner might have access to staff-only entrances); and adapted to environmental characteristics (a complex environment will generally require a more detailed description) and dynamics (think of the full car park). This context-dependency in question-answering is currently avoided in most route direction services that maximally enrich instructions by a few personalized landmarks selected from the user's history.

Since natural language uses predominantly qualitative relations, the preferred reasoning is qualitative reasoning.

Similarly, the nearness relation is known to be context-dependent and occasionally even asymmetric. For example, *the sneaker shop near the food court* and the *airport near the city centre* require different handling of the proximal relation. The interpretation of proximity prepositions is a persisting challenge.

Anatomy of a Place Assistance System

In order to converse about place, an assistance system needs to manage the challenges in handling place information along the four interleaved tasks of capturing an individual's expression as it refers to places (covering the parsing of natural language), modelling extracted knowledge about places in some representation, processing such knowledge, and, thus, finally, interacting with the individual.

Capture. Place knowledge expressed in common language—questions or answers, commands, or explanations—can be parsed, and place references extracted and interpreted within their conversational context. This necessarily includes the ability to parse and interpret the rich means of referring to the various facets of place, including but not limited to toponym resolution and disambiguation.

The intricacies of handling context, scale, and vagueness become salient when locative expressions from natural language interactions are matched against the content of a knowledge base, with the main challenge of resolving ambiguities. When these knowledge bases are geometric ones (*spatial databases*) this matching is called geocoding.[12,13,27] The knowledge base can also be a qualitative one; see next section.

Some challenges in this matching

process concern the unification between different references to the same place (for example, finding out that *the sneakers shop* and *Hype* are actually one and the same place in particular contexts), and the disambiguation where the same term refers to different places (for example, finding out that *Hype* in one conversation is a different place to a *Hype* store in another context). A related challenge is to establish that such a reference refers to an object missing in the knowledge base (for example, if the mall is in the knowledge base, but no *Hype*), or to a changed one (for example, if *Hype* is referring to the place that is stored in the knowledge base still as *Pens&Paper*).[6,33,35]

Modeling. Place knowledge is inherently relational, structured by encoding primarily qualitative relations between places encoded in place references, instead of coordinate-based grounding. The representation of such relational place knowledge—whether inherent in the questions, captured in knowledge bases or encoded in answers—requires relational data structures adapted for place reasoning. Such a representation is by so-called *place graphs*[43]—as illustrated in the figure here—an application of the concept of property graphs. Property graphs have recently received intensive attention in the database community, and are the basis of the first new query language standard considered by ISO in 35 years (https://www.gqlstandards.org/resources/committees-and-processes). Place graphs also provide the means to support the capture of conversational context, scale and rich facets of place.[7]

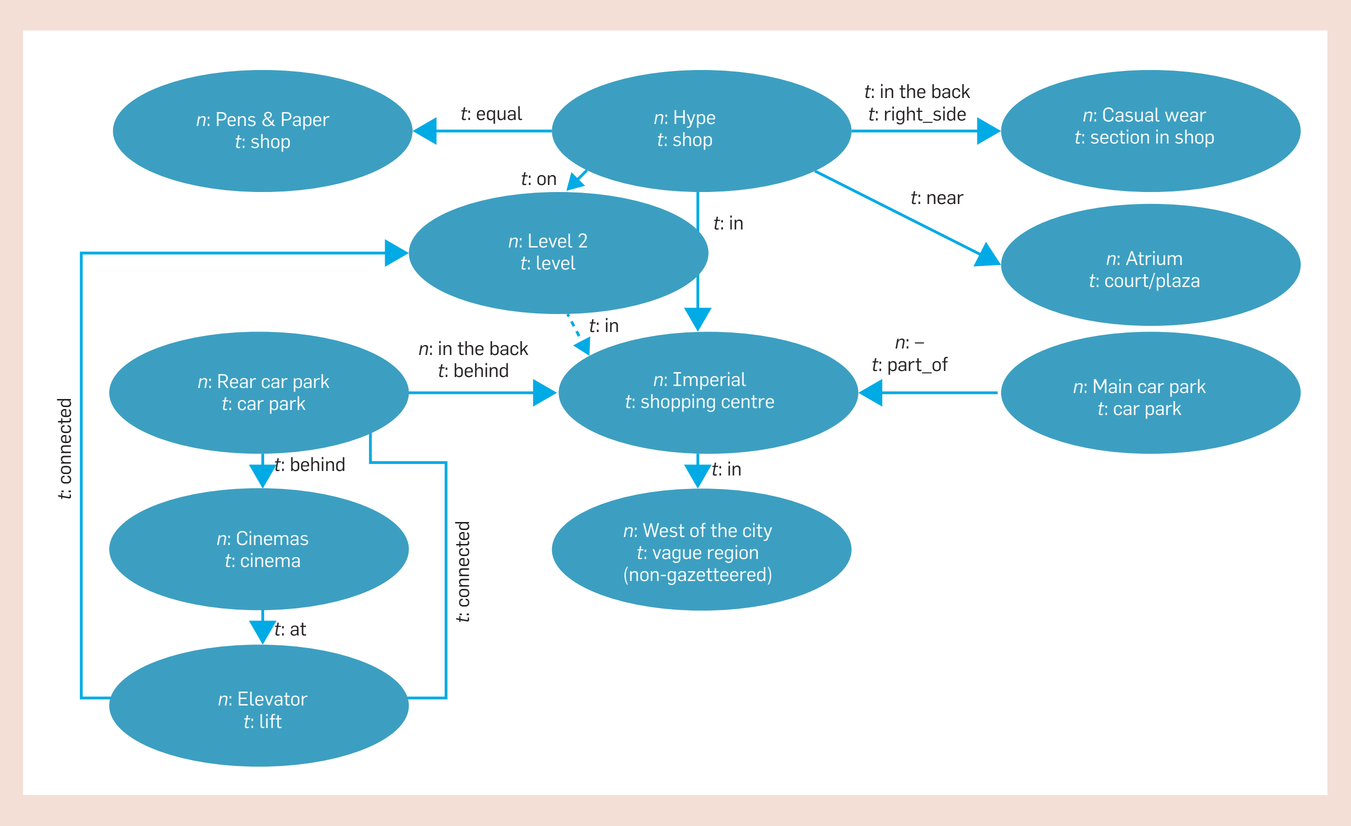
Place graph structures enable the capture of knowledge about relations between places, even in cases where places have vague extents. As place graphs do not need to be anchored into a geographical coordinate reference system, the knowledge captured enables the reasoning flexibility observed when humans reason about places.[6] In particular, the matching of the place knowledge against the knowledge gathered from additional utterances during conversations enables local consistency checking, without requirements for perfect matching against a globally consistent place knowledge base.

Reasoning. Reasoning refers to the task of inferring unknown information from known or given information. Since natural language uses predominantly qualitative relations, the preferred reasoning is qualitative reasoning. For example, if *the sneakers shop is inside the shopping center*, and the *shoes are inside the sneakers shop*, it is possible to infer with certainty that the *shoes are also inside the shopping center*. This kind of compositional transitive reasoning relies on formalisms that precisely define vague spatial relations between entities.[26]

Yet, individuals may have their own vague conceptualizations of common spatial relations,[22,23] and a richer set of spatial prepositions than a formal model of relations provides.[30] They also can flexibly choose spatial reference frames.[21,25,41] For example, telling Kim *to the right* can refer to the walking direction, the actual body orientation, or the shop layout. The interpretation of those projective prepositions requires access to viewpoints, and hence a situated assistant. Its reasoning has to carry the reference frame explicitly, and the determination whether new information is consistent with previous information remains a challenging problem.

While reasoning with predefined sets of relations with well-described compositions is relatively simple, rea-

A graph of the place knowledge extracted from the conversation in the scenario.



soning between different types of spatial relations (for example, a direction relation and a distance relation) is still an open problem. Furthermore, in certain place-related assistance tasks, the standard way of doing spatial reasoning is not applicable. For example, if the food court is left from the elevator, and *Hype* is left from the food court, for formal reasoning it remains ambiguous what the relation between *Hype* and the elevator is (*left or back*). But even if it were known, the composition is typically not helpful since Kim has to pass the food court first in this example and cannot walk directly in the direction of *Hype*. Instead, a valid inference needs to concatenate the two paths. In order to deal with the problem of context and user location, we have developed a unified representation of direction relations that allows us to disambiguate relative direction terms and to reason about them uniformly.[18]

User interaction. The initial—and still dominant—focus of place assistance systems about places.[20] This is now mostly happening ubiquitously and pervasively, including through conversational assistants such as Cortana and Siri. Yet, with the nearing commodification of autonomous systems, users will no longer only serve as consumers of knowledge served by computers, but will increasingly also communicate knowledge to computers that will act on it, including statements about preferences and refined specifications (*Get me to the shopping center—Which entrance would you prefer?*).

This transition to bidirectional interaction will pose new challenges to context handling and language grounding, personalization and privacy,[42] and consideration of the physical affordances of the places as experienced by the combined user-machine system (*The rear entrance, please. Sorry, we lack after-hours access to the rear car park*). Also, language grounding and question-answering systems have thus far focused only on supporting the verbal interaction of users and machines either at the rlocal scale (vista space),[5,30] or conversely, the global geographical scale with relatively coarse place knowledge,[31] while place assistance as in our scenario interacts across all scales in an integrated manner. Furthermore, while situated interaction offers significant contextual clues for refined interpretations of questions, the modalities for machines to capture the situation and to seek additional feedback from users are still in their infancy. This is related to the ability to converse in dialogue as well as to capture facets of the environment that go beyond location and personalization of the user. In particular, the ability of machines to reflect on vague and multifaceted place knowledge is thus far weak both in construction (*West of the city*) and in reasoning (*behind the cinemas*).

A Roadmap

The challenges of conversing about place have not been resolved by artificial intelligence so far, neither conceptually nor technologically. This is primarily caused by:

- Spatial databases (aka geographic information systems) remain strongly grounded in geometry and therefore lack capabilities to handle either the above challenges of capturing information about places (instead enforcing a crisp, scale-dependent, geometry-grounded spatial extent), or for computing and reasoning with qualitative spatial relations.
- Qualitative spatial representations and reasoning[26] capture some of the qualitative spatial relations referred to in language, but in formal interpretations that are not designed to directly cope with the flexibility of language and the embedding of the relations in conversational context. For example, we can reason with near yet we lack the ability to ground its meaning.
- Machine learning methods are neither suited to disambiguate places nor to interpret spatial relations. Due to the cross-cutting factors described above (context-dependence, scale, facets, vagueness, and ambiguity), statistical models often fail to detect significant correlations, or conversely, find spurious correlations—the challenge to interpret the meaning of near remains an excellent example.

In this situation, the anatomy of an idealised place-based assistance system as sketched in our scenario is intended to inspire a vision of a true conversational interaction with a computational assistant that can act in the physical environment, and interact with the user in a natural manner. This vision also enabled us to review some of the progress toward this vision, and to demonstrate that the state-of-the-art approaches are neither sufficient nor have we sufficiently tried to integrate the wide field of research involved.

A first, and enabling stepping stone is the establishment of a knowledge base of place knowledge. The obvious choice for a gold-standard of place knowledge is human verbal expressions—they are qualitative, contextualized (hence, sufficiently disambiguated), and independent from geometric descriptions, from commonly agreed collections of places such as gazetteers, or from named entities in general. Instead, the knowledge base is set up by places committing to an ontology of objects used to refer to (abstract) locations. Note that others have pointed to the contested nature of such extraction.[11]

The next step is extending qualitative spatial representations and reasoning models to enable the handling of context. For example, Hua et al.[17] have shown that reasoning with relative directions (*left, right*) involves a non-trivial ternary relation of two explicit referents and a third, possibly implicit anchor point which has to be identified from the context of the conversation.

A third critical step toward the vision presented here is the ability to effectively query place knowledge bases in natural language. This step profits from learning human answering patterns on place-related questions.[14] The translation of natural language questions into queries (for example, GeoSPARQL) can already be achieved. Thus, the only unresolved dependency for this last step is knowledge bases that contain place knowledge including qualitative spatial relations, which cannot be computed from geometry without context.

More importantly even, this paper makes a case that it is the integration of these areas that will be required to make real progress. In that sense we encourage work on the intersection between spatial databases and machine learning, or natural language processing, in the construction of knowledge bases such as place graphs, and then further between spatial databases, machine learning, and qualitative spatial reasoning for approaching contextualized reasoning.

In summary, situated conversations about places require the machine to understand verbal place-based expressions (commands, descriptions, or questions) and to respond adequately, accessing place knowledge bases. We will soon no longer just ask conversational assistants about well-known place knowledge (*What is the capital of Slovakia?*), but will require systems to understand complex interactions about places, enable machines to seek additional specifications to resolve uncertainty or ambiguity, and have the facility to add new facts extracted from such conversations to its knowledge base. To achieve this, however, a concerted research effort in spatial information science, natural language processing, database management, qualitative spatial reasoning, and AI is necessary.

Acknowledgment. This work was supported by a Discovery Project funded by the Australian Research Council, Grant DP170190109.

References

1. Alirezaie, M. et al. An ontology-based context-aware system for smart homes: E-care@home. *Sensors 17*, 7 (2017), 1586:1–23; https: //www.mdpi.com/1424-8220/17/7/1586
2. Ballatore, A. and Bertolotto, M. Personalizing maps. *Commun. ACM 58*, 12 (Dec. 2015), 68–74; https://doi.org/10.1145/2756546
3. Canter, D. The facets of place. *Toward the Integration of Theory, Methods, Research, and Utilization.* G.T. Moore and R.W. Marans (Eds.). Springer, Boston, MA, 1997, 109–147; https://doi.org/10.1007/978-1-4757-4425-5_4
4. Chahuara, P., Portet, F., and Vacher, M. Context-aware decision making under uncertainty for voice-based control of smart home. *Expert Systems with Apps. 75* (2017), 63–79; https://doi.org/10.1016/j.eswa.2017.01.014
5. Chai, J., Fang, R., Liu, C., and She, L. Collaborative language grounding toward situated human-robot dialogue. *AI Magazine 37*, 4 (2017), 32–45; https://doi.org/10.1609/aimag.v37i4.2684
6. Chen, H., Vasardani, M., and Winter, S. Georeferencing places from collective human descriptions using place graphs. *J. Spatial Information Science 17* (2018), 31–62; https://doi.org/10.5311/JOSIS.2018.17.417
7. Chen, H., Vasardani, M., Winter, S., and Tomko, M. A graph database model for knowledge extracted from place descriptions. *Intern. J. Geo-Information 7*, 6 (2018), Paper 221 (30 pages); https://doi.org/10.3390/ijgi7060221
8. Derungs, C. and Purves, R. From text to landscape: Locating, identifying and mapping the use of landscape features in a Swiss Alpine corpus. *Intern. J. Geographical Information Science 28*, 6 (2014), 1272–1293.
9. Dey, A., Abowd, G., and Salber, D. A Conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction 16*, 2 (Dec. 2001), 97–166; https://doi.org/10.1207/S15327051HCI16234_02
10. Endsley, M. and Jones, D. *Designing for Situation Awareness: An Approach to Human-Centered Design.* CRC Press, Boca Raton, FL, 2011.
11. Ford, H. and Graham, M. Provenance, power and place: Linked data and opaque digital geographies. *Environment and Planning D: Society and Space 34*, 6 (2016), 957–970.
12. Goldberg, D., Wilson, J., and Knoblock, C. From text to geographic coordinates: The current state of geocoding. *URISA J. 19*, 1 (2007), 33–46.
13. Gritta, M., Pilehvar, M., and Collier, N. 2018. Which Melbourne? Augmenting geocoding with maps. In *Proceedings of the 56th Annual Meeting of the Assoc. Computational Linguistics.* (Melbourne, Australia, 2018) 1285–1296; https: //doi.org/10.18653/v1/P18-1119
14. Hamzei, E., Li, H., Vasardani, M., Baldwin, T., Winter, S., and Tomko, M. Place questions and human-generated answers: A data analysis approach. *Geospatial Technologies for Local and Regional Development.* P. Kyriakidis, D. Hadjimitsis, D. Skarlatos, and A. Mansourian (Eds.). Springer, Cham, Switzerland, 2019, 3–19.
15. Hamzei, E., Winter, S., and Tomko, M. Place facets: A systematic literature review. S*patial Cognition & Computation 20*, 1 (2020), 33–81; https://doi.org/10.1080/13875868.2019.1688332
16. Horvitz, E., Kadie, C., Paek, T., and Hovel, D. Models of attention in computing and communication: From principles to applications. *Commun. ACM 46*, 3 (Mar. 2003), 52–59; https://doi.org/10.1145/636772.636798
17. Hua, H., Renz, J., and Ge, X. Qualitative representation and reasoning over direction relations across different frames of reference. In *Proceedings of the 16th Intern. Conf. on Principles of Knowledge Representation and Reasoning.* AAAI, Arizona State University, 2018, 551–560.
18. Hua, H., Zhang, P., and Renz, J. Qualitative place maps for landmark-based localization and navigation in GPS-denied environments. In *Proceedings of the 27th ACM SIGSPATIAL Intern. Conf. Advances in Geographic Information Systems.* ACM (Chicago, IL, 2019), 23–32; https://doi.org/10.1145/3347146.3359107
19. Huang, H., Gartner, G., Krisp, J., Raubal, M., and Van de Weghe, N. Location based services: Ongoing evolution and research agenda. *J. Location Based Services 12*, 2 (2018), 63–93; https://doi.org/10.1080/17489725.2018.1508763
20. Jones, C. and Purves, R. Geographical information retrieval. *Encyclopedia of Database Systems.* Springer, Boston, MA, 2009, 1227–1231.
21. Klatzky, R. Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections. *Spatial Cognition.* C. Freksa, C. Habel, and K. F. Wender (Eds.). *LNAI 1404.* Springer, Berlin, 1998, 1–17.
22. Klippel, A., Dewey, C., Knauff, M., Richter, K., Montello, D., Freksa, C., and Loeliger, E.-A. Direction concepts in wayfinding assistance systems. In P*roceedings of the Workshop on Artificial Intelligence in Mobile Systems 2004*. J. Baus, C. Kray, and R. Porzel (Eds.). Sonderforschungsbereich 378, Saarbrücken, Germany, 2004, 1–8.
23. Klippel, A. and Montello, D. Linguistic and non-linguistic turn direction concepts. In *Spatial Information Theory*. S. Winter, M. Duckham, L. Kulik, and B. Kuipers (Eds.). *LNCS 4736.* Springer, Berlin, 2007, 354–372.
24. Kuhn, W. Core concepts of spatial information for transdisciplinary research. *Intern. J. Geographical Information Science 26*, 12 (2012), 2267–2276.
25. Levinson, S. Frames of reference and Molyneux's question: Crosslinguistic evidence. *Language and Space.* P. Bloom, M.A. Peterson, L. Nadel, and M.F. Garrett (Eds.). The MIT Press, Cambridge, MA, 1996, 109–169.
26. Ligozat, G. *Qualitative Spatial and Temporal Reasoning.* John Wiley & Sons, Inc., Hoboken, NJ, 2013; https://doi.org/10.1002/9781118601457
27. Melo, F. and Martins, B. Automated geocoding of textual documents: A survey of current approaches. *Trans. in GIS 21*, 1 (2017), 3–38; https://doi.org/10.1111/tgis.12212
28. Montello, D. Scale and multiple psychologies of space. *Spatial Information Theory.* A.U. Frank and I. Campari (Eds.). *LNCS 716.* Springer, Berlin, 1993, 312–321.
29. Montello, D., Goodchild, M., Gottsegen, J., and Fohl, P. Where's downtown? Behavioral methods for determining referents of vague spatial queries. *Spatial Cognition and Computation 3*, 2&3 (2003), 185–204.
30. Moratz, R., and Tenbrink, T. Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations. *Spatial Cognition & Computation 6*, 1 (2006), 63–107; https://doi.org/10.1207/s15427633scc0601_3
31. Punjani, D. et al. Template-based question answering over linked geospatial data. In *Proceedings of the 12th Workshop on Geographic Information Retrieva.* ACM, NY, 2018, Article 7; https://doi.org/10.1145/3281354.3281362
32. Purves, R., Winter, S., and Kuhn, W. Places in information science. *J. Association for Information Science and Technology 70*, 11 (2019), 1173–1182; https://doi.org/10.1002/asi.24194
33. Purves, R. et al. Geographic information retrieval: Progress and challenges in spatial search of text. *Foundations and Trends in Information Retrieval 12*, 2-3 (2018), 164–318.
34. Ragni, M. and Knauff, M. A theory and a computational model of spatial reasoning with preferred mental models. *Psychological Rev. 120*, 3 (2013), 561–588; https://doi.org/10.1037/a0032460
35. Rahimi, A., Cohn, T., and Baldwin, T. Continuous representation of location for geolocation and lexical dialectology using mixture density networks. In *Proceedings of the 2017 Conf. Empirical Methods in Natural Language Processing.* ACL (Copenhagen, Denmark, 2017), 167–176; https://doi.org/10.18653/v1/D17-1016
36. Richter, K., Winter, S., and Rüetschi, U. Constructing hierarchical representations of indoor spaces. In *Proceedings of the 10th Intern. Conf. Mobile Data Management, Workshop on Indoor Spatial Awareness.* Y-C Tseng, P. Scheuermann, and R.H. Güting (Eds.). IEEE Press (Taipei, Taiwan, 2009), 686–691; https://doi.org/10.1109/MDM.2009.117
37. Samet, H. et al. Reading news with maps by exploiting spatial synonyms. *Commun. ACM 57*, 10 (Oct. 2014), 64–77; https: //doi.org/10.1145/2629572
38. Schilit, B., Hilbert, D., and Trevor, J. Context-aware communication. *IEEE Wireless Commun. 9*, 5 (2002), 46–54; https://doi.org/10.1109/MWC.2002.1043853
39. Schmidt, A., Beigl, M., and Gellersen, H. There is more to context than location. *Computers & Graphics 23*, 6 (1999), 893–901; https://doi.org/10.1016/S0097-8493(99)00120-X
40. Sperber, D. and Wilson, D. Relevance Theory. *Handbook of Pragmatics.* L. Horn and G. Ward (Eds.). Blackwell, Oxford, U.K., 2004, 607–632.
41. Tenbrink, T. and Kuhn, W. A model of spatial reference frames in language. *Spatial Information Theory.* M. Egenhofer, N.A. Giudice, R. Moratz, and M. Worboys (Eds.). Springer, Berlin, 2011, 371–390.
42. Toch, E., Wang, Y., and Cranor, L. Personalization and privacy: A survey of privacy risks and remedies in personalization-based systems. *User Modeling and User-Adapted Interaction 22*, 1 (2012), 203–220; https://doi.org/10.1007/s11257-011-9110-z
43. Vasardani, M., Timpf, S., Winter, S., and Tomko, M. From descriptions to depictions: A conceptual framework. *Spatial Information Theory.* T. Tenbrink, J. Stell, A. Galton, and Z. Wood (Eds.). LNCS 8116. Springer, Cham, 2013, 299–319.
44. Weiser, M. The computer for the 21st century. *Scientific American 265*, 3 (1991), 94–105.
45. Winter, S. and Freksa, C. Approaching the notion of place by contrast. *J. Spatial Information Science 2012*, 5 (2012), 31–50.
46. Winter, S., Tomko, M., Vasardani, M., Richter, K., Khoshelham, K., and Kalantari, M. Infrastructure-independent indoor localization and navigation. *Comput. Surveys 52*, 3 (2019), 61:1–61:24.
47. Zhao, S. Toward a taxonomy of copresence. *Presence: Teleoperators & Virtual Environments 12*, 5 (2003), 445–455.

Stephan Winter (winter@unimelb.edu.au) is Professor for Spatial Information at The University of Melbourne, Australia.

Timothy Baldwin is Melbourne Laureate Professor in Cognitive Computing at The University of Melbourne, Australia.

Martin Tomko is Senior Lecturer for Spatial Information at The University of Melbourne, Australia.

Jochen Renz is Professor for Artificial Intelligence at the Australian National University, Australia.

Werner Kuhn is Professor for Geography at the University of California at Santa Barbara, CA, USA.

Maria Vasardani is Senior Research Fellow in Spatial Science at RMIT University, Australia.

2021 ACM 0001-0782/21/7

research highlights

DOI:10.1145/3465214

To view the accompanying paper, visit doi.acm.org/10.1145/3465216

Technical Perspective
An Elegant Model for Deriving Equations

By Sriram Sankaranarayanan

WE HAVE ENCOUNTERED units and elementary dimensional analysis in our high school science classes. For instance, the mass of an object is expressed in kilograms ($\mathtt{kg}$). Likewise, length is expressed using meters ($\mathtt{m}$) and time in seconds ($\mathtt{s}$). Other physical quantities such as acceleration has dimensions $\mathtt{m\ s}^{-2}$ (derived from its definition), whereas force has dimensions $\mathtt{kg\ m\ s}^{-2}$. The latter arises from Newton's second law that states that force (F) is equal to the mass (m) times the acceleration (a). Thus, the dimensional units of quantities reflect important relationships between them.

Suppose we are onboard an aircraft with an array of sensors that are independently measuring, among other things, the values of force, mass and acceleration. We could use the equation $F = ma$ to check, for instance, that a single sensor has not failed. However, in many cases, deriving such laws from "first principles" may be quite cumbersome, if not outright impossible.

Imagine a system running by a patient's bedside in the intensive care unit of a hospital with a continuous stream of data that includes the patient's blood pressure B_P ($\mathtt{kg\ m}^{-1}\ \mathtt{s}^{-2}$), lung volume V (m^3), pulse P ($\mathtt{s}^{-1}$), and body weight W ($\mathtt{kg}$). In this situation, it is unclear whether there are "precise" equations derivable from first principles, or even "approximate" empirical equations that may hold under some situations. Be they exact or approximate, these relationships are useful in numerous applications such as the run-time monitoring of safety critical systems.

Discovering possible relationships between various quantities given observational data suffers from the classic "needle in the haystack" problem. The number of possible hypotheses is astronomically large whereas, in practice, very few of these hypotheses will survive empirical tests. The following paper addresses the key problem of discovering relationships that hold between physical quantities from data using dimensional analysis to drastically narrow down the space of hypotheses.

Machine learning provides many powerful approaches for regression using neural network models to detect relationships between quantities. However, many of the existing approaches do not consider the dimensions of the quantities being modeled. The authors propose a simple, yet elegant approach based on the idea of dimensional analysis in physics: a powerful approach that can postulate possible physical relationships by examining the dimensions of the quantities being related. The "Buckingham π" theorem, which formalizes earlier methods going back to the 19th century, provides an elegant recipe for generating such relationships by finding dimensionless parameters. Using this, given the dimensions of the quantities measured, we may setup a system of linear equations to discover such products. For instance, $F \times m^{-1} \times a^{-1}$ is seen to be dimensionless using this approach, from the dimensions of F, m, and a. Similarly, for the ICU bedside monitor described earlier, the quantity $B_P \times V^{\frac{1}{3}} \times P^{-2} \times W^{-1}$ is dimensionless. However, unlike Newton's second law, the relationship between blood pressure and pulse is much more complex and variable. Thus, suitable statistical tests on the data are used to further classify the relationships obtained from the inference approach presented in the paper.

The authors demonstrate their approach to effectively derive physical relationships from observational data for systems such as an unpowered glider and a pendulum. Their approach empirically discovers Newton's equa tions, which are then used to accurately predict the altitude of the glider or the familiar relationship between the length of the pendulum and its time of oscillation. A more sophisticated and general approach uses the derived dimensionless parameters as input features to train machine learning models on the observed data. This approach compares quite favorably to other off-the-shelf approaches.

Thus, the authors present an elegant approach to inferring models from data that incorporate some of the known relationships between the quantities being modeled using dimensional analysis. Elsewhere, dimensional analysis has been shown to be quite effective in detecting defects in robotic software using dimensions as type annotations that can be derived using program analysis techniques.[2] Furthermore, dimensions provide a type system for physical quantities. Such *type systems* are quite useful in machine learning models wherein we often seek to avoid overfitting by imposing constraints such as monotonicity on the models.[3] I see the proposed dimensional consistency approach as a precursor to *strongly typed* machine learning models that can leverage the power of dependent type systems to specify more sophisticated properties including monotonicity.[1]

References

1. Clancy, K. and Miller, H. Monotonicity types for distributed dataflow. In *Proceedings of the Programming Models and Languages for Distributed Computing*. ACM, 2017.
2. Ore, J-P.W. Dimensional Analysis of Robot Software without Developer Annotations. Ph.D. thesis, Univ. of Nebraska, Lincoln, 2019.
3. Sill, J. Monotonic networks. *Advances in Neural Information Processing Systems 10.* M. Jordan, M. Kearns, and S. Solla, Eds. MIT Press, Cambridge, MA, 1998, 661–667

Sriram Sankaranarayanan is an associate professor of computer science at the University of Colorado, Boulder, CO, USA.

DOI:10.1145/3465216

Deriving Equations from Sensor Data Using Dimensional Function Synthesis

By Vasileios Tsoutsouras, Sam Willis, and Phillip Stanley-Marbell

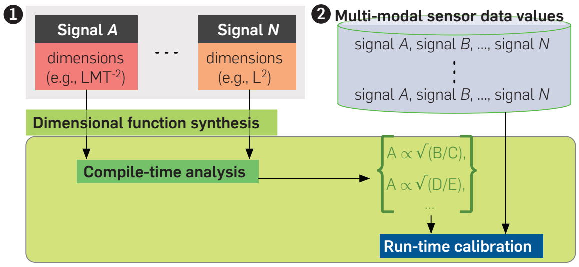
Abstract

We present a new method for deriving functions that model the relationship between multiple signals in a physical system. The method, which we call *dimensional function synthesis*, applies to data streams where the dimensions of the signals (e.g., length, mass, etc.) are known. The method comprises two phases: a compile-time synthesis phase and a subsequent calibration using sensor data. We implement dimensional function synthesis and use the implementation to demonstrate efficiently summarizing multimodal sensor data for two physical systems using 90 laboratory experiments and 10,000 synthetic idealized measurements. The results show that our technique can generate models in less than 300 ms on average across all the physical systems we evaluated. This is a marked improvement when compared to an average of 16 s for training neural networks of comparable accuracy on the same computing platform. When calibrated with sensor data, our models outperform traditional regression and neural network models in inference accuracy in all the cases we evaluated. In addition, our models perform better in training latency (up to 1096× improvement) and required arithmetic operations in inference (up to 34× improvement). These significant gains are largely the result of exploiting information on the physics of signals that has hitherto been ignored.

1. INTRODUCTION

Physical systems instrumented with sensors can generate large volumes of data. These data are useful in understanding previous behaviors of the systems that generate them (e.g., monitoring properties of components in aircraft) as well as in predicting future behaviors of those systems (e.g., predicting failures of components in machinery). Unlike data sources such as speech or text, data from sensors of physical phenomena must obey the laws of physics. Existing methods for constructing predictive models from sensor data however do not fully exploit prior knowledge of the physical interpretation of sensor data. In this work, we use information about physical dimensions of sensor signals to synthesize compact predictive models from sensor data. In keeping with the convention in physics, we use the term *dimensions* to refer to quantities such as length or time and we use the term *units* to refer to a value in a standardized system for quantifying values of a given dimension, such as centimeters or miles for length and Pascals or mmHg for pressure. The state of the art in deriving models from such data streams is to apply some form of machine learning.[11, 19] Blindly applying machine learning to data from physical systems however ignores important prior knowledge about the physical implications of the signals.

Figure 1. Dimensional function synthesis uses information about physical dimensions to generate a family of candidate equations. It then uses sensor measurements to calibrate the set of candidate equations.



1
Signal *A*
dimensions (e.g., LMT^{-2})
. . .
Signal *N*
dimensions (e.g., L^2)
2 Multi-modal sensor data values
signal *A*, signal *B*, ..., signal *N*
⋮
signal *A*, signal *B*, ..., signal *N*
Dimensional function synthesis
Compile-time analysis
A ∝ √(B/C),
A ∝ √(D/E),
...
Run-time calibration

1.1. Contemporary methods ignore physics

Despite its use in programming languages for tasks such as extending type systems with units of measure,[1, 2, 3, 5, 8, 12, 10, 14, 17, 23] physical information in the form of dimensions (e.g., time, temperature, etc.) has seen limited use in building models of physical systems from data. Physical constraints can be viewed as a form of Bayesian prior.[4] Kalman filters incorporate information about the physical constraints of systems but use this information primarily to guide their state update equations. Today, no principled techniques exist which learn models from sensor data by exploiting the requirements of dimensional consistency of sensors to learn more compact models.

1.2. Dimensional function synthesis

Dimensional function synthesis is a new method to efficiently derive functions relating the values from multiple streams of data from physical systems with known physical dimensions. The insight behind the method is that any equation relating physical quantities must obey the principle of *dimensional homogeneity* from dimensional analysis[6]: the two sides of an equation, an addition, or a subtraction, must have the same physical dimensions.

The original version of this paper was published in *ACM Transactions on Embedded Computing Systems*, October 2019.

In a first offline analysis phase, dimensional function synthesis forms monomials of physical parameters whose dimensions, when combined in a monomial, cancel out. Then, in a second run-time stage and using data from sensors of the physical parameters in question, the method calibrates dimensionally plausible equations formed from those monomials to obtain a set of predictive models.

Figure 1 shows a schematic view of the process. The inputs to dimensional function synthesis are a list of signals with known dimensions relevant to the system under study and a set of data values corresponding to instances of those signals. The outcome is a model relating the signals and predicting the expected physical system output. We developed the method of dimensional function synthesis with the objective of creating inference models that can fit within the memory, computation, and energy constraints of low-power embedded systems. The method may also apply to computing systems that are not constrained by compute resources or by energy, but which nonetheless need simple models defined over a large parameter space.

2. MATHEMATICAL FOUNDATION

Dimensional analysis is often introduced in engineering curricula as a simple method for checking the validity of computations on physical quantities. It is frequently used in engineering, fluid mechanics, and electrodynamics in cases such as deflection of turbine blades in turbo machine designs.[20] The approach to dimensional analysis familiar to most researchers in computing systems and computer science involves taking some physical quantity (e.g., acceleration) and expressing it in terms of basic dimensions such as length (L) and time (T) to obtain its dimensions (LT^{-2} for acceleration). Dimensional analysis, however, has a well-developed mathematical framework that combines a few basic principles from physics with an analytic formulation based on linear algebra and group theory.[6, 9, 16] The remainder of Section 2 provides a brief overview of this mathematical formalization of dimensional analysis.

2.1. Parameters in physical equations and dimensionless products

Let i be an index over a set of symbols in a physical equation and let Q_i be one of those symbols in an equation describing a physical system. Typically, these symbols will correspond to parameters of some physical model and we will therefore use the term *parameter* and *symbol* interchangeably. Let $\mathcal{D}(\cdot)$ be a function from symbols to some product of basic dimensions. For any equation describing a physical system, we introduce the set $\mathbb{S}_{\text{symbols}}$, where

$$\mathbb{S}_{\text{symbols}} = \{Q_1, Q_2, \ldots, Q_n\}. \tag{1}$$

For the system described by $\mathbb{S}_{\text{symbols}}$ to be physically plausible, each member Q_i of $\mathbb{S}_{\text{symbols}}$ can be rewritten in terms of a set of basic *dimensions* (e.g., mass, length, time) or is otherwise *dimensionless*. For the example equation $F = m \cdot a$, relating a force F applied to a mass m and its resulting acceleration, a, we have $\mathbb{S}_{\text{symbols}} = \{F, m, a\}$, $Q_1 = F$, $Q_2 = m$, and $Q_3 = a$. The dimensions of the members of $\mathbb{S}_{\text{symbols}}$ are $\mathcal{D}(Q_1) = MLT^{-2}$, $\mathcal{D}(Q_2) = M$, and $\mathcal{D}(Q_3) = LT^{-2}$. Table 1 shows additional examples of parameters and their dimensions for data from sensors in physical systems that can be instrumented with sensors to monitor their behavior. For example, the altimeter subsystem of a fitness tracker uses changes in atmospheric pressure to estimate changes in elevation and hence to estimate the number of flights of stairs climbed.

Table 1. Examples of physical systems and their $\mathbb{S}_{\text{symbols}}$

Physical system	Parameters, $\mathbb{S}_{\text{symbols}}$	Parameters	Dimensions
Altimeter in a fitness tracker	$\mathbb{S}_{\text{symbols}} = \{p, h\}$	Pressure, p	$\mathcal{D}(p) = ML^{-1}\ T^{-2}$
		Elevation, h	$\mathcal{D}(h) = L$
Pendulum	$\mathbb{S}_{\text{symbols}} = \{t, l, g\}$	Period, t	$\mathcal{D}(t) = T$
		Rod length, l	$\mathcal{D}(l) = L$
		Gravity, g	$\mathcal{D}(g) = LT^{-2}$

The key idea in the mathematical formulation of dimensional analysis is that for a set $\mathbb{S}_{\text{symbols}}$ such as in the example above, we can often arrange the members Q_i of $\mathbb{S}_{\text{symbols}}$ into groups of products where the dimensions of the symbols in the product cancel out and as a result each monomial is dimensionless.[6, 7]

Why finding dimensionless products is useful: Given a set of parameters $\mathbb{S}_{\text{symbols}}$ for a physical system, each of the dimensionless products we can form from a subset of $\mathbb{S}_{\text{symbols}}$ directly gives us a dimensionally valid equation between those parameters: we can equate the dimensionless product to any dimensionless quantity to obtain a dimensionally correct equation; if we then rearrange that equation to move one of the parameters to be the only term on one side of the equation, we have a dimensionally valid equation of that parameter in terms of the other parameters in the dimensionless product.

Definition 1. *Let i be an index over the set $\mathbb{S}_{\text{symbols}}$ of symbols in the description of a physical system, let n be the cardinality of $\mathbb{S}_{\text{symbols}}$, and let m be an index such that $m \leq n$. Let k_i be a value drawn from the set of rational numbers $\mathbb{Q}$. A dimensionless product Π of parameters $Q_i \in \mathbb{S}_{\text{symbols}}$ is a monomial of parameters raised to powers such that $\mathcal{D}(\Pi) = 1$, that is,*

$$\Pi = \frac{Q_1^{k_1} Q_2^{k_2} \cdots Q_m^{k_m}}{Q_{m+1}^{k_{m+1}} Q_{m+2}^{k_{m+2}} \cdots Q_n^{k_n}}. \tag{2}$$

For a physical system defined by a set of parameters $\mathbb{S}_{\text{symbols}}$, we can define groups of one or more dimensionless products based on Definition 1. Because of the form of Equation (2), these groups of dimensionless products are often referred to as *Π groups*.[6, 7]

Example: for $\mathbb{S}_{\text{symbols}} = \bigcup_i \{Q_i\}$ and the dimensionless product

$$\frac{Q_1^{k_1} Q_2^{k_2} \cdots Q_m^{k_m}}{Q_{m+1}^{k_{m+1}} Q_{m+2}^{k_{m+2}} \cdots Q_n^{k_n}},$$

we can equate the dimensionless product to a constant to obtain

$$\frac{Q_1^{k_1} Q_2^{k_2} \cdots Q_m^{k_m}}{Q_{m+1}^{k_{m+1}} Q_{m+2}^{k_{m+2}} \cdots Q_n^{k_n}} = C.$$

We can then obtain an expression for any of the $Q_i \in \mathbb{S}_{\text{symbols}}$. For example, for Q_1,

$$Q_1 = \sqrt[k_1]{\frac{C Q_{m+1}^{k_{m+1}} Q_{m+2}^{k_{m+2}} \cdots Q_n^{k_n}}{Q_2^{k_2} \cdots Q_m^{k_m}}}.$$

This simple idea generalizes to a method for obtaining a function relating all the parameters, $Q_i \in \mathbb{S}_{\text{symbols}}$, relevant to a system, in terms of one or more dimensionless products that we can form from $\mathbb{S}_{\text{symbols}}$.

2.2. Groups of dimensionless products and the Buckingham Π theorem

The primary insight exploited in many contemporary applications of dimensional analysis[18, 21] is that for any physical system represented by a set of physical parameters $\mathbb{S}_{\text{symbols}}$, it is often possible to reparametrize the system in terms of a smaller number of parameters. This basic observation is often used in the engineering and design of mechanical systems to reduce the number of parameters needed in experimentation. The principle behind the observation is what is commonly known as the Buckingham Π theorem[6]:

Theorem 1. *Let n be the number of parameters in a description of a physical system, that is, $n = |\mathbb{S}_{\text{symbols}}|$. Let r be the number of dimensions from some orthogonal dimensional bases that are sufficient to express the dimensions of the parameters in $\mathbb{S}_{\text{symbols}}$. Then, n – r dimensionless products Π_i can be formed from the parameters.*

The $n - r$ dimensionless products Π_i are the roots of some function Φ, that is,

$$\Phi(\Pi_1, \Pi_2, \cdots, \Pi_{n-r}) = 0. \tag{3}$$

Let Φ′ be a function over the dimensionless products Π_i. It follows for the i-th product, Π_i, that,

$$\Pi_i = \Phi'(\Pi_1, \Pi_2, \cdots, \Pi_{i-1}, \Pi_{i+1}, \ldots, \Pi_{n-r}). \tag{4}$$

when $n - r$ equals 1, that is, when there is only one Π product in the Π groups, then

$$\Phi(\Pi_1) = 0. \tag{5}$$

It follows that there exists some real-valued constant C such that

$$\Pi_1 = \frac{Q_1^{k_1} Q_2^{k_2} \cdots Q_m^{k_m}}{Q_{m+1}^{k_{m+1}} Q_{m+2}^{k_{m+2}} \cdots Q_n^{k_n}} = C. \tag{6}$$

There are multiple possible Π groups: for the same parameter set $\mathbb{S}_{\text{symbols}}$, of cardinality *n,* there are multiple possible groups of dimensionless products (i.e., multiple possible Π groups).

3. DIMENSIONAL FUNCTION SYNTHESIS

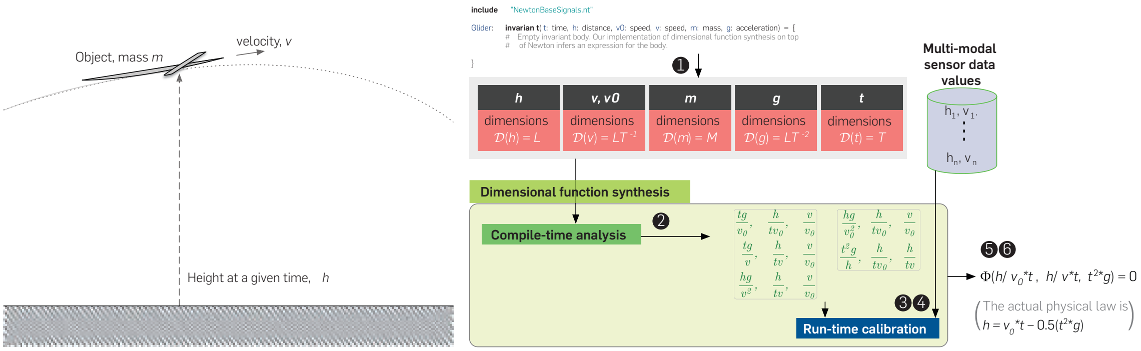
From the set $\mathbb{S}_{\text{symbols}}$ of parameters defining a physical system, we can construct a matrix representation of the system, where the columns are the parameters that are members of $\mathbb{S}_{\text{symbols}}$, the rows are base dimensions such as length, mass, or time, returned by the function $\mathcal{D}$ (Section 2.1), and the elements in the matrix are the exponents of the base dimensions.

Dimensional function synthesis consists of a compile-time step which automatically computes *all the valid Π products across all possible Π groups*. Then, a run-time step calibrates the functional relationship between the derived Π products. Similar to other data-driven techniques, it uses sensor measurements as inputs and produces a model that maps those measurements to an expected output. Its advantage is the use of dimensional information to learn a simpler model than would otherwise be possible. Because of the small size of the produced model and the small amount of data required to calibrate it, dimensional function synthesis is well suited for execution on resource-constrained embedded systems. Figure 2 shows the steps using the terminology introduced in this section and a physical system comprising an unpowered flying object (glider) as an example.

3.1. Deriving the dimensionless product groups

Let the set of base dimensions be $\mathbb{S}_{\text{base dimensions}}$. We assume without loss of generality that $\mathbb{S}_{\text{base dimensions}} = \{I, \Theta, T, L, M, J, N\}$ corresponding to the base S.I. dimensions for electric current, thermodynamic temperature, time, length, mass,

Figure 2. A glider of mass *m* launched with initial velocity v_0 moves through space with velocity *v* under gravitational acceleration *g*. Dimensional function synthesis can derive a set of candidate equations relating its height *h* to time *t*. Next, using sensor data, it can calibrate that set of candidate equations to obtain the model for height as a function of time and gravity.



```
include    "NewtonBaseSignals.nt"

Glider:    invarian t( t: time, h: distance, v0: speed, v: speed, m: mass, g: acceleration) = {
           #   Empty invariant body. Our implementation of dimensional function synthesis on top
           #    of Newton infers an expression for the body.

}
```

Object, mass *m*
velocity, *v*
Height at a given time, *h*
❶
h
v, v0
m
g
t
dimensions $\mathcal{D}(h) = L$
dimensions $\mathcal{D}(v) = LT^{-1}$
dimensions $\mathcal{D}(m) = M$
dimensions $\mathcal{D}(g) = LT^{-2}$
dimensions $\mathcal{D}(t) = T$
Multi-modal sensor data values
$h_1, v_1, \vdots, h_n, v_n$
Dimensional function synthesis
Compile-time analysis
❷
$\frac{tg}{v_0}, \frac{h}{tv_0}, \frac{v}{v_0}$
$\frac{tg}{v}, \frac{h}{tv}, \frac{v}{v_0}$
$\frac{hg}{v^2}, \frac{h}{tv}, \frac{v}{v_0}$
$\frac{hg}{v_0^2}, \frac{h}{tv_0}, \frac{v}{v_0}$
$\frac{t^2g}{h}, \frac{h}{tv_0}, \frac{h}{tv}$
❸❹
Run-time calibration
❺❻
$\Phi(h/ v_0{*}t,\ h/ v{*}t,\ t^2{*}g) = 0$
(The actual physical law is $h = v_0{*}t - 0.5(t^2{*}g)$)

luminous intensity, and amount of matter, respectively. Let r be the cardinality of $\mathbb{S}_{\text{base dimensions}}$, let j be an index over r, and let $q_j \in \mathbb{S}_{\text{base dimensions}}$ be one of the base dimensions. As in Section 2.1 and Equation (1), let i be an index over the set of parameters for a physical system and let Q_i be one such parameter. Let a_{ij} be an exponent of one of the base dimensions of Q_i as returned by the function $\mathcal{D}$ from Section 2.1. We can express the dimensions of any Q_i in terms of the base dimensions q_j:

$$Q_i = q_1^{a_{i1}} q_2^{a_{i2}} \cdots q_r^{a_{ir}}. \tag{7}$$

We can represent the system of $n = |\mathbb{S}_{\text{symbols}}|$ equations, one for each of the $1 < i \leq n$ instances of Equation (7) with a matrix called the *dimensional matrix*.[7, 9, 13]

Definition 2. *Let n be the number of parameters in* $\mathbb{S}_{\text{symbols}}$ *and let r be the number of fundamental dimensions required to express them. Let i be an index over the set of n parameters for a physical system and let j be an index over r. Then we define the dimensional matrix* **A**, *as*

$$\mathbf{A} = (a_{ji})_{(r,n)}. \tag{8}$$

The products Π from Definition 1 and Equation (2) will be dimensionless (i.e., the dimensions in the monomial will cancel out) if and only if $\mathbf{Ak} = \mathbf{0}$, where the matrix k contains the exponents of the base dimensions needed to yield a dimensionless product. The solution of $\mathbf{Ak} = \mathbf{0}$ is the null space $N(\mathbf{A})$.

Physical restrictions on solutions of $N(\mathbf{A})$**:** because of our objective of finding physically plausible dimensionless groups that are efficiently computable, we restrict the solutions to the null space computation to rational powers of a_{ji} as opposed to permitting arbitrary real-valued exponents. As a result of this insight, we compute the *rational null space* of **A** which will by definition give us a_{ji} values that are ratios of integers. To compute the rational null space of **A**, we first use Gauss-Jordan elimination to reduce the matrices to their reduced row-echelon form (RREF), where all pivots equal one, with zeros below each pivot.[22] Once the matrix is in RREF, we find the special solutions to $\mathbf{Ak} = \mathbf{0}$. If for a specific **A**, the only solution is the zero vector, then we conclude that no nontrivial null space is available and as a result it is not possible to form a dimensionless product with rational exponents from the set of parameters in $\mathbb{S}_{\text{symbols.}}$

The number of linearly independent columns of the dimensional matrix **A** is equal to rank(**A**). Thus, to find all possible solutions to $\mathbf{Ak} = \mathbf{0}$ and hence all possible groups of dimensionless products, we can rearrange the n columns of **A** in $\binom{n}{\mathrm{rank}(\mathbf{A})}$ ways to yield different null space solutions.[6, 13] Our final set of dimensionless product groups is the union of all the unique dimensionless product groups resulting from computing the null spaces.

3.2. Calibration: using sensor data to transform Π groups to equational models

The dimensionless groups obtained by analyzing a description of the physical system in the form of the set $\mathbb{S}_{\text{symbols}}$ give proportionality relations between the parameters in $\mathbb{S}_{\text{symbols}}$.

In the general case where more than one of the dimensionless products are not constant, then, from Equation (4), there is a function Φ′ that relates the values of one of the Π products to the rest of them. We can use a data-driven approach to find the form of Φ′ and we call this step *calibration*. In this case, we apply the generated Π products to transform the data at calibration-time and achieve dimensionality reduction. This allows simpler models to perform better, allowing smaller models to be learned with less data for a given prediction performance.

When a dimensionless product group contains a single item, Equation (6) showed that we can equate the dimensionless product to a constant and obtain a proportionality relation between the symbols in the dimensionless product. We still need to determine the value of the constant of proportionality and we can do so given one or more values of the parameters in the dimensionless group. When a dimensionless product group contains more than one dimensionless product, we can still apply this method if we can determine that all but one of the products in any of the dimensionless groups are effectively constant for the range of values of the parameters of interest.

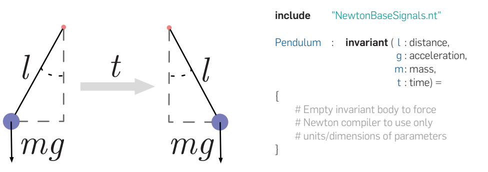
Like any model construction method, dimensional function synthesis will produce incomplete results if the inputs to the method do not fully describe the problem being modeled: an incomplete $\mathbb{S}_{\text{symbols}}$ can result in an empty set of dimensionless products.

3.3. Implementation using Newton language

We implemented dimensional function synthesis by extracting the set $\mathbb{S}_{\text{symbols}}$ from the intermediate representation of descriptions of physical systems written in Newton,[15] a domain-specific language for describing physical systems. We use Newton solely as a convenient way to obtain the set $\mathbb{S}_{\text{symbols}}$ from a human-readable description.

Pendulum example: Figure 3a shows a pendulum instrumented with a sensor that measures movement. By measuring, for example, angular movement with a gyroscope or acceleration with an accelerometer, we can measure the period of oscillation t by computing the Fourier transform of time series data from the sensor. Our goal is to obtain a model relating t, the length of the rod l, and the component g of the acceleration due to gravity in the plane of rotation of the pendulum. The insights from this example are applicable to many sensor-instrumented mechanical systems such as ones where the period of oscillation

Figure 3. (a) A simple pendulum with mass *m*, rod of length *l*, period of swing *t*, and with the component of the acceleration due to gravity in its plane of motion being *g*. (b) Physical description for the ideal pendulum written in Newton.



```
include      "NewtonBaseSignals.nt"

Pendulum   :   invariant ( l : distance,
                           g : acceleration,
                           m: mass,
                           t : time) =
{
      # Empty invariant body to force
      # Newton compiler to use only
      # units/dimensions of parameters
}
```

(a) Simple pendulum.

(b) Newton description.

might be affected when lengths of system parts expand or contract with temperature, or when the component of gravitational acceleration affecting the system changes due to the system being tilted at an angle. Figure 3b shows a physical description for the ideal pendulum written in Newton. Dimensional function synthesis, implemented as a new backend for the Newton compiler, takes this description as input and performs the following steps.

Step 1: Dimensional matrix construction. For the system in Figure 3a, the parameter set is $\mathbb{S}_{\text{symbols}} = \{l, g, m, t\}$. The last row of Table 2 shows the dimensions of the members of the parameter set $\mathbb{S}_{\text{symbols}}$ along with the dimensionless group computed by the method described above in Section 3.1. Following the formulation in Section 3.1, the dimensional matrix **A** for the pendulum's parameter set $\mathbb{S}_{\text{symbols}}$ is

$$\mathbf{A} = \begin{array}{c|cccc} & l & g & m & t \\ \hline T & 0 & -2 & 0 & 1 \\ L & 1 & 1 & 0 & 0 \\ M & 0 & 0 & 1 & 0 \end{array}$$

Step 2: Dimensional matrix column permutation and Π group computation. The total number of parameters is $n = |\mathbb{S}_{\text{symbols}}| = 4$. From Definition 1 (Section 2.2), the pendulum system has $n = 4$ physical quantities and $r = 3$ base dimensions. Consequently, $n - r = 1$ and there is a single unique Π product:

$$\Pi_0 = \begin{array}{cccc} g & l & m & t \\ \hline 1 & -1 & 0 & 2 \end{array}$$

From Equation (6) (Section 2.2), it follows that we can equate the corresponding monomial to some constant C:

$$\frac{g * t^2}{l} = C. \tag{9}$$

Given sensor measurements for different values of l, g, and t, we can determine the value of the constant C.

4. MODEL EVALUATION

To demonstrate the potential of dimensional function synthesis, we compare it against black-box data-driven approaches for the characterization of a physical system. The fundamental idea is that a scientist has assembled a physical system and is able to measure a subset of its parameters either by inspection (e.g., measuring the length of a component) or by using sensors (e.g., accelerometers, tachometers, etc.). Given that a complex physical system requires effort and expertise to be analytically defined, its data-driven characterization is a promising idea. The designer can collect a large dataset of observations from the physical system and then use regression and machine learning to derive a model that fits the measured parameters to an expected output.

However, deriving an effective data driven model requires good sampling of the physical system's parameters and extensive exploration of the design space of available data fitting models. In practice, both these requirements are hard or impossible to meet, especially in the case of complex, multiparametric systems. On the contrary, the outcome of dimensional function synthesis can either *fully characterize the system or act as a starting point for targeted data-driven analysis*. In addition, simple dimensional functions have significantly less computational requirements compared to the majority of data-driven characterization techniques.

Table 2. Examples of physical system descriptions ($\mathbb{S}_{\text{symbols}}$) and the dimensionless groups our technique generates for them. Our implementation generates the LATEX for the equations shown in the last column.

Physical system	Input to our technique		Dimensions	Example of one dimensionless group generated by our automated method
Vibrating string	$\mathbb{S}_{\text{symbols}} =$	String tension, t	$\mathcal{D}(t) = MLT^{-2}$	$\left\{\frac{(L^2)(\mu)(f^2)}{(t)}, \frac{(t)}{(\mu)(f^2)(\rho^2)(\theta^2)}\right\}$
	$\{t, L, \mu, f, \rho, \theta\}$	String length, L	$\mathcal{D}(L) = L$	
		String mass per unit length, μ	$\mathcal{D}(\mu) = ML^{-1}$	
		String vibration frequency, f	$\mathcal{D}(f) = T^{-1}$	
		Thermal expansion coefficient, ρ	$\mathcal{D}(\rho) = \Theta^{-1}$	
		String temperature, θ	$\mathcal{D}(\theta) = \Theta$	
Unpowered flying object	$\mathbb{S}_{\text{symbols}} =$	Object elevation, h	$\mathcal{D}(h) = L$	$\left\{\frac{(t^2)(g)}{(h)}, \frac{(h)}{(t)(v_0)}, \frac{(h)}{(t)(v)}\right\}$
	$\{h, v_0, v, m, g, t\}$	Object initial velocity, v_0	$\mathcal{D}(v_0) = LT^{-1}$	
		Object velocity, v	$\mathcal{D}(v) = LT^{-1}$	
		Object mass, m	$\mathcal{D}(m) = M$	
		Acceleration due to gravity, g	$\mathcal{D}(g) = LT^{-2}$	
		Time, t	$\mathcal{D}(t) = T$	
Pendulum	$\mathbb{S}_{\text{symbols}} =$	Rod length l	$\mathcal{D}(l) = L$	$\left\{\frac{(g)(t^2)}{(l)}\right\}$
	$\{l, g, m, t\}$	Acceleration due to gravity, g	$\mathcal{D}(g) = LT^{-2}$	
		Mass, m	$\mathcal{D}(m) = M$	
		Oscillation period, t	$\mathcal{D}(t) = T$	

4.1. Evaluation for synthetic data

We first compare dimensional function synthesis to regression and neural networks using synthetic idealized data. We examine several neural network topologies from the FitNet family of curve-fitting neural network architectures, which are optimized for equation fitting. We target an unpowered flying vehicle (glider) with initial velocity v_0, mass m, acceleration due to gravity g, and, trajectory height h at time t, similar to the example of Figure 2. We examine the ability of our method to find the relation between trajectory height and the rest of the physical parameters of the glider. The parameters used to describe the glider result in multiple Π groups, each of which includes multiple Π products. In this case, the form of the function Φ′ for combining the Π products into an equational model is unknown and we must use a data-driven approach to find its form. Dimensional function synthesis provides two options for the calibration phase: (1) performing calibration on the target embedded system; and (2) performing calibration offline on a computing system that is not constrained by resources. In both cases, the calibrated models target the embedded platform, so final model complexity is still a key restriction.

In contrast to Φ and Φ′, which are functions of dimensionless products, let Ψ be a function directly relating the parameters of a system. For the glider example, we compare our approach to a data-driven approach for fitting the feature vector $<v_0, m, g, t>$ to a predicted height h through the function Ψ:

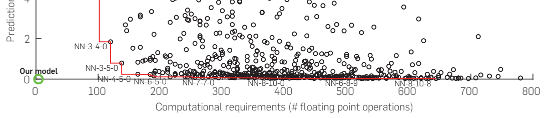
$$h = \Psi(v_0, m, g, t). \tag{10}$$

The ideal trajectory equation of a glider is $h = v_0 \cdot t - 0.5 \cdot (t^2 \cdot g)$. Using the ideal trajectory equation, we synthesize a dataset by uniformly sampling the initial velocity of the glider (v_0) in the range of 1 m/s to 10 m/s, with a step size of 0.5 m/s. We considered acceleration due to gravity (g) from 6.0 m/s^2 to 9.5 m/s^2, with 0.5 m/s^2 step size, and a time window for gliding (t) ranging from 0.1 to 100 s, with a step of 0.1 s.

Using dimensional function synthesis, the chosen description of the system leads to three Π groups, each with two Π products, that is, Π **group 0** = $\{\Pi_1 = t \cdot g/v_0, \Pi_2 = h/t \cdot v_0\}$, Π **group 1** = $\{\Pi_1 = h \cdot g / v_0^2, \Pi_2 = h / t \cdot v_0\}$, Π **group 2** = $\{\Pi_1 = t^2 \cdot g/h, \Pi_2 = h/t \cdot v_0\}$. In Π group 0, h appears only in Π_2, thus according to Equation (4), we can express h as a function Φ′ of Π_1:

$$\frac{h}{t \cdot v_0} = \Phi'(\frac{t \cdot g}{v_0}). \tag{11}$$

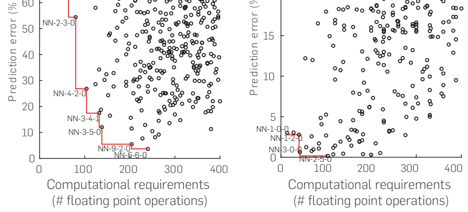
Figure 4. Prediction error versus computational requirements for predicting the trajectory of a glider. Our model uses linear regression for fitting function Φ′ of Equation (11) (denoted as "our model" in the lower left corner). It Pareto-dominates all the neural network variants (891 different network topologies), which are used for fitting function Ψ of Equation (10).



In contrast to traditional methods that must learn a function over a four-dimensional space $\langle v_0, m, g, t\rangle$, dimensional function synthesis only needs to use data to learn the single-variable function Φ′ of Equation (11). This simpler form is particularly valuable when our goal is to perform the final calibration on a resource-constrained embedded system. Figure 4 shows the comparative performance of using linear regression to find the dimensionally reduced Φ′, against linear, quadratic, and neural network-based regression to find Ψ. Linear regression on Φ′ outperforms the same technique on Ψ by more than 12%, although having similar computational requirements. Neural networks are capable of minimizing the prediction error, at the expense of over 80× greater required computation. We quantify the computational requirements of each network as the total number of floating-point operations (additions, multiplications) that it requires per inference instance. Overall, the neural network models require between 0.3 and 50 s training per model for 5-fold cross-validation, with an average of 16 s. The total training latency was approximately 240 minutes on an Intel Core i7-7820X CPU at 3.60 GHz, with 32 GB RAM. This is 1096× slower than our approach which requires 1.5 ms on average for the examined physical system running on the same workstation. We have examined a total of 16 physical systems of increasing complexity and our method requires less than 300 ms on average to generate the dimensional functions, with a maximum of 3428.7 ms.

Figure 5 shows model approximation performed by neural networks trained against 20 data points, with (Figure 5b) and without (Figure 5a) dimensionally reducing the number of input parameters by making use of dimensional function synthesis. The most accurate neural network for

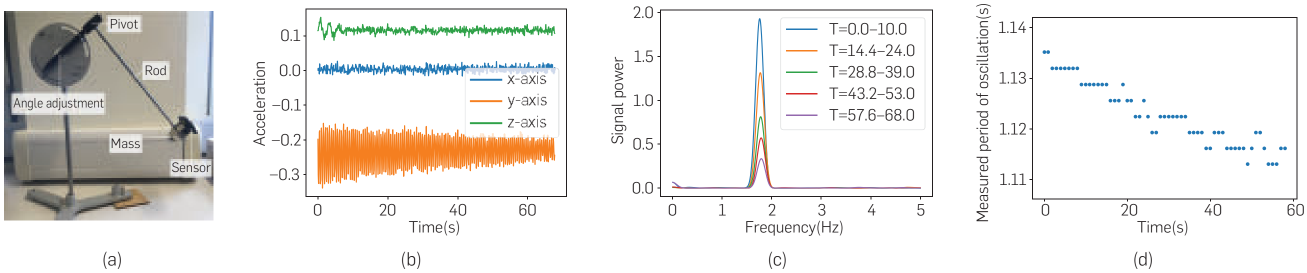
Figure 5. Prediction error versus computational requirements for predicting the trajectory of a glider. Subfigure (a) corresponds to the straightforward application of neural networks for fitting function Ψ of Equation (10). Subfigure (b) corresponds to our approach using a neural network for fitting function Φ′ of Equation (11). We train all models against a set of 20 input data points. Our method achieves prediction error of 0.17% via an approximately 2.5× less computationally demanding model.



(a) Black box approach.

(b) Our approach.

Figure 6. (a) Our experimental setup for the variable-*g* pendulum. (b) Data collected from the 3-axis accelerometer over time using the wireless sensor on the pendulum. The largest component of oscillation is due to the motion of the pendulum. (c) Discrete Fourier Transform (DFT) of 10 s windows of the sampled acceleration data. Despite the variation of signal properties over time, the dominating frequency remains around 2 Hz. (d) The time period of the pendulum, calculated according to the dominating frequency in each time window of DFT, exhibits a small variation of about 20 ms over a 1-minute interval.



fitting the function Φ' over the four-dimensional space $\langle v_0, m, g, t\rangle$ has prediction error of 0.17%. It consists of two layers with 2 and 5 neurons, whereas the most accurate for fitting function Ψ is composed of two layers of 6 neurons each. This highlights dimensional function synthesis as a tool for training models in situations where there are insufficient data to train more complex models.

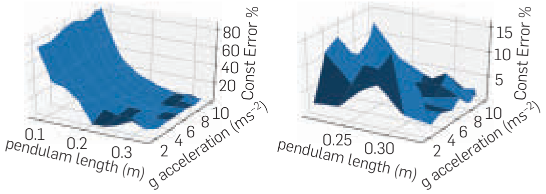
The simpler models and higher prediction accuracy of dimensional function synthesis are the result of its ability to use the physical information available. This enables better training of simpler models with less data. Most importantly, these reductions are not based on ad-hoc assumptions or approximations, but are dictated by physical laws. Models from dimensional function synthesis are more efficient for resource-constrained embedded systems as they require fewer computations during inference and less data for their training.

4.2. Evaluation on a physical pendulum

We evaluate our method in the presence of nonsynthetic data where the underlying relationship is more complex than a simple closed-form equation. We perform a series of experiments in our laboratory using an apparatus known as a *variable-g pendulum* (Figure 6a). This apparatus uses a mass on a stiff rod swinging about a pivot which is at an angle that is not perpendicular to the horizon. We instrument this apparatus with a wireless sensor containing a 3-axis accelerometer at the "bob" end of the pendulum to provide a data stream from which we automate measuring the period of oscillation, *t*. We run 90 physical experiments on this apparatus for different values of the pendulum rod length *l* in the range of 3–33 cm in steps of 3 cm and for a range of effective gravitational acceleration *g* resulting from pendulum pivot angles of 0°–80°, in 10° increments.

Figure 6b shows an example of the sensor data over 1 minute of pendulum oscillation. We recorded a time series of pendulum swing data such as that in Figure 6b for each of the 90 experiments we performed. We then used these time series data to calculate the oscillation period via its discrete Fourier transform (DFT). Figure 6c shows the resulting DFT output for one experiment, for four different processing windows of recorded data. Figure 6d shows the oscillation period over the duration of one 1-minute experiment, estimated using the DFT.

Figure 7. Percentage error of the predicted period of the variable-*g* pendulum, *t* for a given length *l* and gravitational acceleration, *g*. Subfigure (a) includes all experimental instances in a subset of which the ideal pendulum model assumptions are violated leading to high deviations. Subfigure (b) zooms in the region, where the error of synthesized dimensional functions is minimized.



(a) All experimental data.

(b) Zoom for *l* > 20 cm.

Figure 7 shows the ability of our method to generate a model that accurately predicts the period of oscillation of the variable-*g* pendulum. The calibration step of our method takes as input the periods estimated from the actual experiment. Our method requires minimal calibration data. For pendulum lengths greater than 20 cm, the prediction error is always less than 15% even though each prediction requires only four floating-point operations.

For pendulum lengths less than 20 cm, the error in the model increases due to nonidealities, such as friction, that are not captured by the form of the proportionality relation generated by our technique. The accuracy of the synthesized dimensional function is limited by the number of utilized parameters that describe the physical system. A richer choice in the set of parameters (e.g., such as the friction of the pivot and mass of the rod) is a possible solution to derive more accurate dimensional functions.

We also applied the black-box data-driven techniques on the assembled data of the pendulum experiment. Of this dataset, 75% was randomly sampled to act as training data, whereas the rest was used as testing samples. We used a 5-fold cross-validation policy to train the models. Figure 8 summarizes the prediction error of the period of

Figure 8. Percentage error of the predicted period of the variable-*g* pendulum, *t* for a given length *l*, and gravitational acceleration, *g*. We predict using neural networks and regression models.

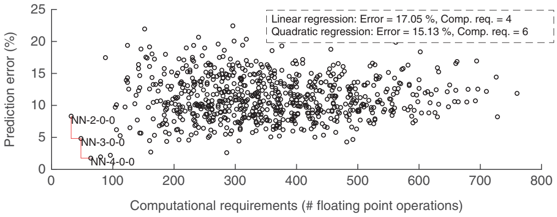
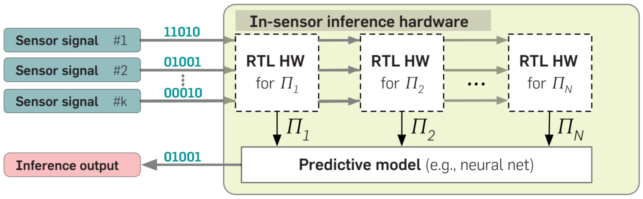


Figure 9. The hardware generated by dimensional circuit synthesis preprocesses *k* sensor signals to calculate *N* < *k dimensionless products* $\Pi_1 \ldots \Pi_N$. A predictive model takes the calculated product values as input and generates an inference output.



pendulum oscillation averaged for all models in the case of the testing dataset. Regression models have prediction error comparable to our method, but our method outperforms regression models in the zoomed area of Figure 7b. Neural networks exhibit a wide distribution of prediction error, but simple networks are able to achieve very high accuracy within the same range as our proposed model. Because we train the black-box models against data points derived from the entire range of the pendulum experiments, they can effectively capture the nonideal characteristics of the oscillation, thus achieving high accuracy.

5. SCOPE, LIMITATIONS, AND EXTENSIONS

Dimensional function synthesis uses information on the physical dimensions and units of measure of the signals relevant to a physical system to derive a set of candidate equations relating those signals. Such as many existing approaches for constructing models based on human-chosen parameters, it depends on a valid set of parameters in the set $\mathbb{S}_{\text{symbols}}$ (introduced in Section 2.1) for describing the system to be modeled. When provided with a set of parameters insufficient to generate a model that captures a system's behavior, the method will unsurprisingly generate a model that is, at best, only an approximation to the true behavior. Exciting areas of further development include automating the process of identifying parameters in $\mathbb{S}_{\text{symbols}}$ rather than extracting them from a human-written description and incorporating integrals and derivatives in formulations for Φ functions.

For physical parameters that cannot be directly measured, dimensional function synthesis faces the same challenges faced by traditional modeling approaches. In practice, for parameters that cannot be measured, designers measure surrogates that correlate to the missing parameters, for example, measuring acceleration and elapsed time instead of velocity. In this case, dimensional function synthesis has the net effect of exploiting information on the physical units of the parameters in question, whereas traditional modeling techniques have no option but to attempt to fit data with ever more complex nonlinear models. Dimensional function synthesis enables the combination of both approaches in the case of multiple Π groups as examined in Section 4.1.

5.1. Dimensional circuit synthesis

Dimensional circuit synthesis is an extension of dimensional function synthesis that provides a compile-time method to generate digital logic circuits for the calculation of Π groups. We have implemented a Verilog register transfer level (RTL) synthesis backend in Newton, which uses the information of the calculated Π groups of dimensional function synthesis and generates the RTL description of hardware modules, each of which computes a Π monomial (Equation (2)) of a selected Π group. The hardware modules take sensor signals as input and perform the pre-inference processing of the calibrated predictive module that we derive from dimensional function synthesis. An on-device (in-sensor) inference engine will integrate the synthesized dimensional circuits with the module that executes the calibrated predictive model using, for example, a neural network. This inference module can either be a custom RTL component or a programmable core. Figure 9 shows an in-sensor inference hardware system generated using dimensional function synthesis and dimensional circuit synthesis.

We evaluated the hardware generated by the dimensional circuit synthesis backend using a Lattice Semiconductor iCE40 FPGA. The iCE40 is a low-power miniature FPGA in a wafer-scale WLCSP package of 2.15 × 2.50 mm, which targets sensor interfacing tasks and on-device machine learning. We used a fully open-source FPGA design flow, comprising the YoSys synthesis tool (`version 0.8+456`) for synthesis and NextPNR (`version git SHA1 5344bc3`) for placing, routing, and timing analysis.

We performed our measurements on an iCE40 Mobile Development Kit (MDK) which includes a 1Ω current sense resistor in series with each of the supply rails of the FPGA (core, PLL, I/O banks). We measure the current drawn by the FPGA core by measuring the voltage drop across the FPGA core supply rail (1.2 V) resistor using a Keithley DM7510, a laboratory-grade 7½ digital multimeter that can measure voltages down to 10 nV. Using these voltage drop measurements, we computed the power dissipated by the FPGA core for each configured RTL design. We used a pseudorandom number generator to feed the Π monomials computation circuit modules under evaluation with random input data.

We evaluated dimensional circuit synthesis on seven different physical systems described in Newton. Table 3 presents the total FPGA resource utilization for all the generated Π product computation modules, expressed in terms of the number of four-input lookup tables (LUT4 cells) required for their synthesis. These resource utilization values also include the required resources for the synthesis of the fixed-point arithmetic modules,

Table 3. Experimental evaluation on iCE40 FPGA of dimensional circuit modules generated from descriptions of physical systems.

Name	LUT4 cells	Maximum frequency	Execution latency	Avg. power at 12 MHz	Avg. power at 6 MHz
Beam	2958	16.88 Mhz	115 cycles	3.5 mW	1.8 mW
Pendulum, static	1402	17.07 Mhz	115 cycles	2.0 mW	1.1 mW
Fluid in pipe	4258	15.65 Mhz	188 cycles	5.8 mW	3.0 mW
Unpowered flight	1930	16.44 Mhz	81 cycles	2.3 mW	1.2 mW
Vibrating string	2183	16.67 Mhz	183 cycles	2.5 mW	1.3 mW
Warm vibrating string	3137	16.77 Mhz	269 cycles	1.9 mW	1.0 mW
Spring-mass system	1419	16.67 Mhz	115 cycles	3.4 mW	1.8 mW

which we integrated in the computation module of each Π product.

The execution latency column lists the required cycles for completing the calculations of the critical path of each of the generated RTL modules. We obtained the number of cycles by simulating the execution of the RTL modules for pseudorandom inputs generated by linear feedback shift registers (LFSRs). In each RTL module, we parallelize the calculation of different Π products but the required operations per Π product are executed serially.

The last column of Table 3 shows the measured power dissipation of each design configured in the iCE40 FPGA. In all cases, the power dissipation is less than 6 mW and as low as 1 mW, demonstrating the suitability of our method for small-form-factor, battery-operated on-device inference at the edge.

6. CONCLUSION

Existing methods for constructing retrospective or predictive models for data from physical systems do not fully exploit information about the physics of the systems in question. In this work, we present an automated method for generating the family of functions from which to learn a model, based on information about the physical dimensions of the signals in the system. The method, which we call *dimensional function synthesis,* applies to data streams where the dimensions of the signals are known.

We implement dimensional function synthesis and evaluate the execution cost and accuracy of the models our method generates compared against regression models and neural networks. When calibrated with sensor data, our models outperform traditional regression and neural network models in inference accuracy in all the cases we evaluated. In addition, our models perform better in training latency (up to 1096× improvement) and required arithmetic operations in inference (up to 34× improvement). These significant gains are largely the result of exploiting information on the physics of signals that has hitherto been ignored.

Acknowledgments

This research is supported by an Alan Turing Institute award TU/B/000096 under EPSRC grant EP/N510129/1, by Royal Society grant RG170136, and by EPSRC grant EP/V004654/1. Youchao Wang, Vlad-Mihai Mandric, and James Rhodes contributed to the implementation of the linear-algebraic methods within the Newton compiler.

References

1. Allen, E., Chase, D., Luchangco, V., Maessen, J.-W., Steele, G.L., Jr. Object-oriented units of measurement. In *Proceedings of the 19th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*, OOPSLA'04 (2004), ACM, New York, NY, USA, 384–403.
2. Antoniu, T., Steckler, P.A., Krishnamurthi, S., Neuwirth, E., Felleisen, M. Validating the unit correctness of spreadsheet programs. In *Proceedings of the 26th International Conference on Software Engineering*, ICSE'04 (2004), IEEE Computer Society, Washington, DC, USA, 439–448.
3. Babout, M., Sidhoum, H., Frecon, L. Ampere: A programming language for physics. *European J. Phys. 11*, 3 (1990):163.
4. Barber, D. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, Cambridge, 2012.
5. Biggs, G., Macdonald, B.A. A pragmatic approach to dimensional analysis for mobile robotic programming. *Auton. Robots 25*, 4 (Nov. 2008), 405–419.
6. Buckingham, E. On physically similar systems; Illustrations of the use of dimensional equations. *Phys. Rev. 4*, 4 (1914), 345–376.
7. Carlson, D.E. A mathematical theory of physical units, dimensions, and measures. *Arch. Rational Mechanics Anal. 70*, 4 (1979), 289–305.
8. Cmelik, R.F., Gehani, N.H. Dimensional analysis with C++. *IEEE Softw. 5*, 3 (1988), 21–27.
9. Carlson, D.E. On some new results in dimensional analysis. *Arch. Ration. Mech. Anal. 68*, 3 (1978), Springer, 191–210.
10. Hilfinger, P.N. An ada package for dimensional analysis. *ACM Trans. Program. Lang. Syst. 10*, 2 (Apr. 1988), 189–203.
11. Hills, D.J.A., Grütter, A.M., Hudson, J.J. An algorithm for discovering Lagrangians automatically from data. *PeerJ Comput. Sci. 1*, (Nov. 2015), e31.
12. Hills, M., Chen, F., Roşu, F. A rewriting logic approach to static checking of units of measurement in C. *Electron. Notes Theor. Comput. Sci. 290*, (Dec. 2012), 51–67.
13. Jonsson, D. Dimensional analysis: A centenary update. arXiv preprint arXiv:1411.2798 (2014).
14. Kennedy, A. Dimension types. In *Proceedings of the 5th European Symposium on Programming: Programming Languages and Systems,* ESOP'94 (1994), Springer-Verlag, London, UK, 348–362.
15. Lim, J., Stanley-Marbell, P. Newton: A language for describing physics. *CoRR,* abs/1811.04626 (2018).
16. Rayleigh, L. The principle of similitude. *Nature 95* (Dec. 1915), 66–68.
17. Rittri, M. Dimension inference under polymorphic recursion. In *Proceedings of the Seventh International Conference on Functional Programming Languages and Computer Architecture,* FPCA'95 (1995), ACM, New York, NY, USA, 147–159.
18. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N. Data-driven discovery of partial differential equations. *Sci. Adv. 3*, 4 (2017), e1602614.
19. Schmidt, M., Lipson, H. Distilling free-form natural laws from experimental data. *Science 324*, 5923 (2009), 81–85.
20. Simon, V., Weigand, B., Gomaa, H. *Dimensional Analysis for Engineers*. Springer, Gewerbestrasse, Cham, Switzerland, 2017.
21. Sonin, A.A. A generalization of the *Π*-theorem and dimensional analysis. *Proc. Natl. Acad. Sci. 101*, 23 (2004), 8525–8526.
22. Strang, G. *Introduction to Linear Algebra*, 5th edn. Wellesley-Cambridge Press, Wellesley, MA, 2016.
23. Umrigar, Z.D. Fully static dimensional analysis with C++. *SIGPLAN Not. 29*, 9 (Sept. 1994), 135–139.

Vasileios Tsoutsouras, Sam Willis,
and **Phillip Stanley-Marbell**
({vt298, sjw238, phillip.stanley-marbell}@
eng.cam.ac.uk), University of Cambridge,
Cambridge, U.K.

DOI:10.1145/3465173

To view the accompanying paper, visit doi.acm.org/10.1145/3465212 rh

Technical Perspective
Tracking Pandemic-Driven Internet Traffic

By Jennifer Rexford

THE INTERNET IS a research experiment that "escaped from the lab" to become a critical global communications infrastructure during our lifetimes. Over the past year of the COVID-19 pandemic, the Internet has supported friends and families staying in touch and supporting each other, remote work and learning, and the global collaboration of experts designing much-needed treatments and vaccines. As challenging as the past year (and more) has been, the Internet has made it possible for many important aspects of life, work, and culture to continue.

In March 2020, the Internet suddenly became a lifeline for people all over the world. Designed to withstand failures, attacks, and fluctuations in traffic, the Internet proved up to the task. Almost overnight, demand for Internet services grew dramatically, and shifted in both time and space. Many Internet service providers (ISPs) had network designs with spare capacity, deployed more bandwidth in critical locations, and relaxed bandwidth caps on low-income households. The Internet protocols, designed to adapt to changing conditions, were able to deliver reasonable service to many users by sharing the available resources dynamically.

As challenging as the past year (and more) has been, the Internet has made it possible for many important aspects of life, work, and culture to continue.

The following paper offers a detailed look at how Internet traffic changed during the COVID-19 pandemic. The paper is distinctive in analyzing traffic measurements from multiple networks—ISPs, three major Internet eXchange points (IXPs), a mobile provider, and a university network—across a long period of time. The combination of longitudinal data from multiple, diverse vantage points is truly unusual, and a testament to the large group of authors who worked with each other, their home institutions, and other stakeholders to acquire the measurement data.

The study shows how, as the spring 2020 lockdowns began, traffic surged for ISPs and IXPs while decreasing for mobile providers and university campuses. Residential traffic shifted quickly to having high loads during normal business hours. Normally dominated by download traffic, the volume of upstream traffic (from residential users to the Internet) increased even more dramatically due to interactive applications like videoconferencing. Other applications like video-on-demand streaming and online gaming increased, as users sought sources of news, education, and entertainment. People working from home also increased their use of Virtual Private Network (VPN) technologies for remote access to online resources within their companies and universities. The study also shows that traffic patterns changed throughout the year, with fluctuations caused by shifts in national lockdown strategies, vacation and holiday seasons, bouts of bad winter weather, and more.

While the Internet has been remarkably robust to the shifts in demand, the experiences of the past year offer valuable lessons for the future, including:

- ▸ The "digital divide" is more pronounced than ever, now that good Internet connectivity is critical for work, education, and medical information.
- ▸ Interactive applications, like video conferencing, are incredibly sensitive to even small fluctuations in network reliability and performance, leaving users frustrated.
- ▸ The enduring problems of Internet security and privacy become even more serious as attackers exploit vulnerabilities in popular applications as well as people's growing dependence on the Internet.

As the computer science community continues the important work of addressing these challenges, we should also redouble our efforts to collect and analyze the measurement data needed to understand Internet traffic, performance, and applications. With the Internet such an essential part of our daily lives, we can easily forget that no one person, company, country, or organization is truly in charge of making this "network of networks" hold together as one global infrastructure. We can only understand, and therefore improve, the Internet by observing how it responds under pressure from many locations. C

Jennifer Rexford is the Gordon Y.S. Wu Professor in Engineering and chair of the Computer Science Department at Princeton University, Princeton, NJ, USA.

DOI:10.1145/3465212

A Year in Lockdown: How the Waves of COVID-19 Impact Internet Traffic

By Anja Feldmann, Oliver Gasser, Franziska Lichtblau, Enric Pujol, Ingmar Poese, Christoph Dietzel, Daniel Wagner, Matthias Wichtlhuber, Juan Tapiador, Narseo Vallina-Rodriguez, Oliver Hohlfeld, and Georgios Smaragdakis

Abstract

In March 2020, the World Health Organization declared the Corona Virus 2019 (COVID-19) outbreak a global pandemic. As a result, billions of people were either encouraged or forced by their governments to stay home to reduce the spread of the virus. This caused many to turn to the Internet for work, education, social interaction, and entertainment. With the Internet demand rising at an unprecedented rate, the question of whether the Internet could sustain this additional load emerged. To answer this question, this paper will review the impact of the first year of the COVID-19 pandemic on Internet traffic in order to analyze its performance. In order to keep our study broad, we collect and analyze Internet traffic data from multiple locations at the core and edge of the Internet. From this, we characterize how traffic and application demands change, to describe the "new normal," and explain how the Internet reacted during these unprecedented times.

1. INTRODUCTION

The worldwide pandemic caused by the Corona Virus 2019 (COVID-19) is a once-in-a-generation global phenomenon that changed the lives of billions of people and destabilized the interconnected world economy. What started as a local health emergency in Asia at the end of 2019, turned into a global event at the beginning of 2020 when the first cases appeared on other continents. By March 2020, the World Health Organization (WHO) declared COVID-19 as a pandemic, causing many governments around the globe to impose strict lockdowns of economic and social activities to reduce the spread of COVID-19. These measures changed the habits of a large fraction of the global population, who now depend on residential Internet connectivity for work, education, social interaction, and entertainment.

Changes in Internet user behavior are common, but they normally occur gradually and over long periods of time. Notable examples of such changes are the increase in demand for peer-to-peer applications that happened in the early 2000s; the increase of traffic served by content delivery networks—such as an increase in streaming—that took place in the 2010s; and, more recently, the elevated demand for mobile applications. In all of these cases, the telecommunications industry and network operator community reacted by increasing the investment on network infrastructure. However, the changes in Internet user behavior during the pandemic have been unique because the shifts took place within weeks, leaving hardly any time to react. This raised questions of whether user behavior changes yield to changes in Internet traffic and, more importantly, concerns if the Internet is able to sustain this additional load.

In this paper, we investigate the impact of the COVID-19 pandemic on the Internet traffic by analyzing more than two years of Internet traffic data including the first year of the pandemic. More specifically, we characterize the overall traffic shifts and the changes in demand for particular applications that became very popular in a short amount of time. During the process, we try to understand if there is a "new normal" in Internet traffic and to see how the Internet reacted in these unprecedented times. We summarize our observations for the spring 2020 wave (February 2020 to June 2020) and then extend our study for the fall[a] 2020 wave (September 2020 to February 2021). To that end, we collect and analyze network traffic data from multiple vantage points, such as a large Internet Service Provider (ISP) in Europe, three Internet Exchange Points (IXPs) in Europe and the US, as well as a mobile operator and a metropolitan academic network in Europe (REDIMadrid).

Our main observations can be summarized as follows:

- Changes in traffic volume follow demand changes, causing a traffic surge of 15–20% during the fall 2020 lockdown for the ISP/IXPs in our study. In summer 2020, after the reopening of the economy, an increase of about 20% at one IXP, but only 6% at the Tier-1 ISP, is still visible. The fall 2020 wave also had an impact, with the annual traffic increase in 2020 being higher than in a typical year.
- The observed traffic increase mostly takes place during nontraditional peak hours. Daily traffic patterns are moving to weekend-like patterns, especially during the spring 2020 lockdown.

[a] We use "spring" and "fall" from the viewpoint of the Northern hemisphere, where our vantage points are located. Exchange both terms for the Southern hemisphere.

An earlier version of this paper was published in the *Proceedings of the 20th ACM Internet Measurement Conference* (IMC'20).[9]

- Traffic related to remote working applications, such as VPN connectivity applications and video-conferencing applications, surges by more than 200%. VPN traffic seems to remain at elevated levels even during the fall 2020 wave.
- Traffic changes across networks differ. For example, in the REDIMadrid campus network, there was a significant drop (by up to 55%) in traffic volume on workdays after the spring 2020 lockdown as most people were not on campus, but an increase during the fall 2020 lockdown. Traffic at the IXP and the ISP also varies depending on the mandated lockdown policy and due to the different customer profiles.

2. DATASETS

The Internet is formed by a network of networks. Depending on their size and position, networks may act as large traffic hubs forwarding network traffic between up to hundreds or even thousands of other networks (backbone/core). In other cases, they are interconnected weakly and closer to the consumer at the edge of the topology.

To obtain a broad perspective on the impact of the pandemic on the Internet, we observe the Internet from multiple and diverse vantage points. They are located at the backbone and peering points of a major Tier-1 Internet Service Provider (ISP), at the core of the Internet—namely three Internet Exchange Points (IXPs) around the globe—and at the edge (a metropolitan university network and a mobile operator).

ISP. A large Central European ISP that provides service to more than 15 million fixed line subscribers and also operates a transit network (Tier-1).

IXPs. An IXP is an interconnection facility where networks become members to exchange traffic with other members across the IXP's infrastructure. In our study, we consider three major IXPs. The largest IXP, namely IXP-CE, is located in Central Europe. It has more than 900 members and a peak traffic of more than 9 Tbps. The second IXP, namely IXP-SE, is located in Southern Europe and has more than 170 members. The third IXP, namely IXP-US, is located at the US East Coast and has more than 250 members.

REDIMadrid university network. We collect and analyze data from the REDIMadrid academic network, which interconnects 16 independent universities and research centers in the region of Madrid. It serves nearly 290,000 users such as students, faculty, researchers, student halls, WiFi networks (such as eduroam), and administrative and support staff.

Mobile operator: A European mobile provider with more than 40 million customers.

At each vantage point, we collect and analyze traffic flows before and during the pandemic. This allows us to reason about the impact of the COVID-19 pandemic on Internet traffic and discuss related traffic shifts. In order to guarantee user privacy, we only analyze anonymized or aggregated datasets. For additional details on our measurement methodology, we refer to Feldmann et al.[9]

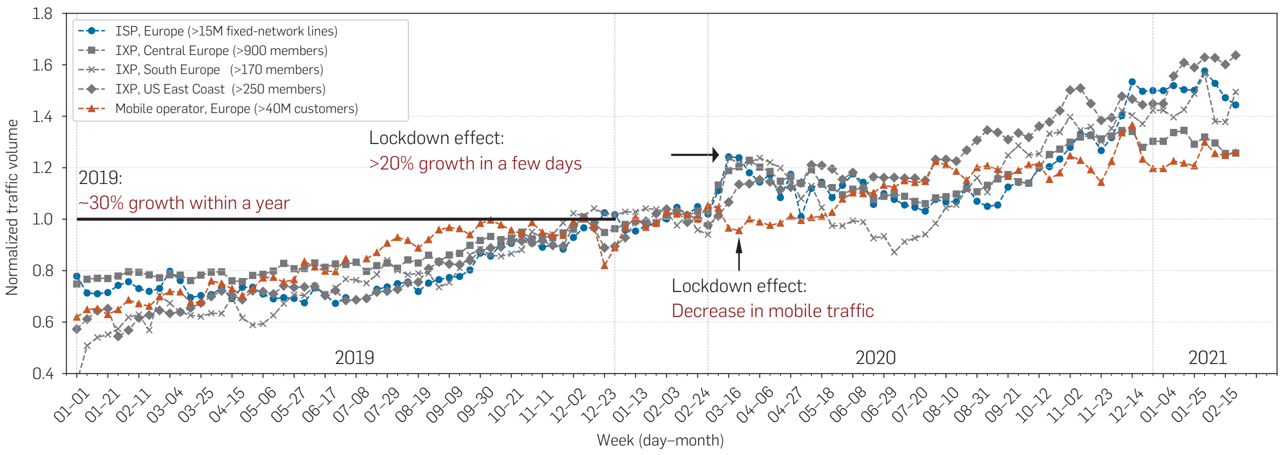
3. NETWORK TRAFFIC SHIFTS

To understand traffic changes during the COVID-19 pandemic, we first look for overall changes in well-established traffic patterns before, during, and after the strictest lockdown periods for both the spring and fall 2020 waves. Because all data sources exhibit very different traffic characteristics and volumes, we normalize the data to make it comparable. In Figure 1, we show the normalized aggregated traffic of the ISP, IXPs, and mobile operator vantage points from January 2019 until the end of February 2021. We normalize this by using the traffic of the first week of January 2020 for each corresponding vantage point.

3.1. Macroscopic observations

In Figure 1, we annotate the week of the initial lockdowns in the countries that host the ISP and IXPs in Central and Southern Europe, and the mobile operator. Although the exact dates of when the lockdown was imposed differ across Europe, these dates are very close to each other and followed the declaration of COVID-19 as a pandemic by the WHO. A first observation is that the ISP and IXPs in Central and Southern Europe show a more than 20% traffic increase within a week after the official announcement of the

Figure 1. Traffic changes during the COVID-19 pandemic's spring and fall waves at our Internet vantage points.



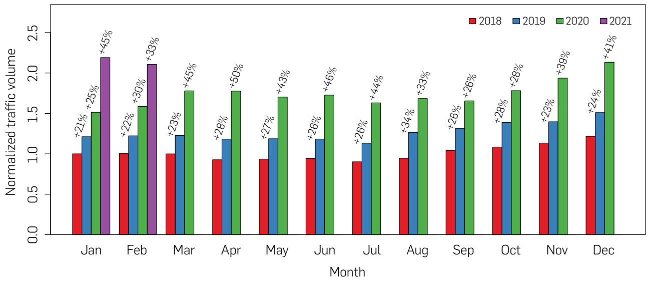
lockdown. This could be perceived as a “moderate” surge in traffic. However, in Internet reality, this is a substantial increase in traffic in only a short period of time. To put it into perspective, the figure shows that the annual increase of 2019 was around 30%, which is similar to the annual increase in previous years. This means that the expected traffic increase in one year happened only within a couple of weeks in March 2020 following the spring 2020 lockdown. Well-provisioned networks, such as the ones we measured for our study, could cope with this surge. However, networks that “ran hot” may have faced problems as this increase is significant and takes place in a relatively short period of time. In sharp contrast, the traffic of the European mobile operator decreased as users switched to WiFi and reduced their commute and traveling. This aligns with reports in other studies.[15] It is worth noting that, during the same period, the traffic at the IXP at the U.S. East Coast surged only by 2% as there was no announcement of strict lockdowns in the U.S. at that time.

The impact of the spring wave was significant as the traffic levels remain at same elevated level during the lockdown. The traffic of the IXP at the U.S. East Coast increased significantly when lockdown measures took place by the state authorities. However, when the economy opened again after June 2020, we observe a slight decrease of ISP and IXPs traffic as well as an increase of the mobile operator’s traffic.

The impact of the fall wave is clearly visible in traffic patterns beginning in September 2020. The traffic at the ISP and all IXPs surged again, whereas the traffic of the mobile operator declined, except for the holiday period at the end of 2020. Although the lockdowns in the fall of 2020 differ significantly from country to country, and in some cases there were lockdowns with on-off periods, the impact of the fall wave was significant. The 2020 annual increase for the ISP and the IXPs varied between 35% and 50%, that is, higher than the expected annual increase. The mobile operator showed an annual increase of around 20%, which is lower than expected. As the fall wave of COVID-19 continues into 2021, we observe similar trends until February 2021. It is also worth noting that in some countries, the fall wave was a superposition of multiple waves of COVID-19 and its mutations, which were faced with harder lockdown restrictions. Additionally, the severe weather conditions in Southern Europe with historic snow volumes in January and February 2021 may have also played an additional role in keeping people at home and the corresponding increase of Internet traffic at the ISP and IXPs.

Figure 2 focuses on the ISP, where we show the normalized aggregated traffic for each month for the years 2018–2021. Although during the years 2018–2019 the traffic increase was around 30% compared to the same month in the previous year, there was a dramatic change after March 2020. To understand this, we annotate the increase between a given month and the one in the previous year for 2019, 2020 until February 2021 above the bars in the figure. During the spring 2020 wave, the traffic in each month increased by around 45% compared to the traffic in 2019; the peak was in April 2020 with about 50%. Between August and October 2020, the traffic increase was similar to previous years. This aligns well with good weather conditions and the relaxation of lockdown policies. However, this period was followed by stronger lockdown policies, and there is again a surge that continues until the end of our observation period, that is, February 2021.

Figure 2. ISP monthly normalized downstream traffic change during the COVID-19 pandemic with percentage increase compared to the previous year.



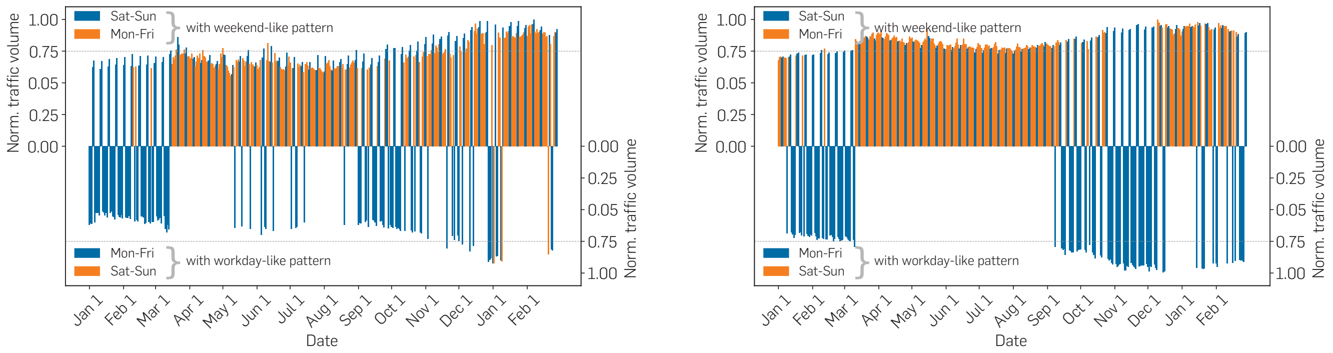
The traffic increases we have seen across vantage points can arise unexpectedly and may create a need for capacity increases by network operators. We observed capacity increases in the order of 1,500 Gbps (3%) across many IXP members at the IXP-CE alone. Beyond our datasets, some networks publicly reported that traffic shifts due to the pandemic resulted in partial connectivity issues and required new interconnections.[8, 20] The vantage points in this paper range from extremely large to moderate sizes with sufficient resources and a lot of experience in network provisioning and resilience. In general, smaller networks with limited resources may not be able to plan with sufficient spare capacities and fast enough reaction times to compensate for such sudden changes in demand. In fact, performance degradation issues have been reported in less developed regions,[1] which also highlights the digital divide.

3.2. Drastic shifts in Internet usage patterns

Beyond the macroscopic observations, our analysis sheds light on the shifts in Internet usage patterns that are also relevant to network operation and management. The Internet’s regular workday traffic patterns are significantly different from weekend patterns.[13] On workdays, traffic peaks are concentrated in the evening, typically between 18:00 and midnight, also referred to as “peak hours.” During the weekend, the activity is more distributed also in the nonpeak hours as more people are at home and using the Internet.

With the pandemic lockdown in March, this workday traffic pattern shifts toward a continuous weekend-like pattern. More specifically, we call a traffic pattern a workday pattern if the traffic spikes in the evening hours and a weekend pattern if its main activity gains significant momentum from approximately 9:00 to 10:00 am. Figure 3a and b shows the normalized traffic for days classified as weekend-like on the top and for workday-like on the bottom. If the classification is in line with the actual day (workday or weekend) the bars are colored in blue, otherwise they are colored in orange. We find that up to mid-March, most days are classified correctly. The only exception is the holiday period at the

Figure 3. Drastic shifts in Internet usage patterns during the COVID-19 pandemic. Classification of weekend- and workday-like patterns.



(a)ISP-CE:Weekend-like(top)versus workday-like(bottom).

(b)IXP-CE:Weekend-like(top)versus Workday-like(bottom).

beginning of the year in Figure 3b. This pattern changes drastically once the lockdown measures are implemented. Indeed, almost all days are classified as weekend-like. This change persists in Figure 3b until the end of August due to the vacation period, which is consistent with the behavior observed in 2019 (not shown). By contrast, Figure 3a shows that the shift toward a weekend-like pattern becomes less dominant as countermeasures were relaxed in mid-May, but in August, the pattern resembles again the weekend-pattern due to the vacation period.

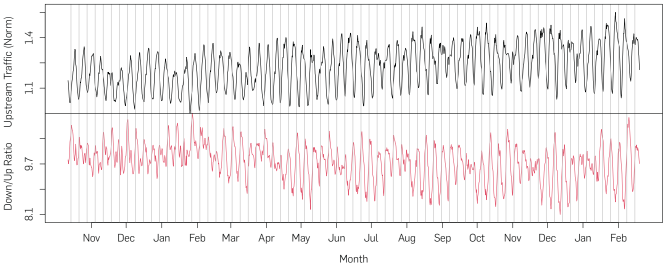
During the period of August 2020–December 2020, the patterns both at the ISP and the IXP are back to the usual weekday and weekend pattern. When the first lockdown of the fall COVID-19 wave was imposed in December 2020, this pattern was disrupted, more noticeably at the IXP. In the first two months of 2021, there was a mixed pattern for both the ISP and the IXP. We conclude that we still observe a transient behavior in 2021 and it is unclear whether the changes of daily usage patterns are here to stay.

3.3. Effect on the traffic asymmetry

As we discussed in the previous sections, residential traffic surged both during the spring and fall COVID-19 waves. In this section, we take a closer look on the directionality of the traffic and comment on new patterns in upstream and downstream traffic. Recall that residential traffic is asymmetric in nature, that is, downstream traffic is typically many times higher than the upstream one. This is to be expected as users send less traffic than they receive when using applications such as video streaming and browsing. In Figure 4 (top), we show the aggregated upstream traffic from October 2019 to end of February 2021. There is a slight increase in upstream traffic after the first lockdown in mid-March 2020. This trend manifests itself in the following months: The minimum, but more noticeably the maximum upstream level increase across the rest of the observation period.

As a result of the general elevated traffic levels, the downstream traffic also increases during this period. To assess if there is a change in the established ratio between upstream and downstream traffic, in Figure 4 (bottom), we plot the ratio of upstream versus downstream traffic. Before the COVID-19 pandemic, typical values of this ratio were around 9.8 with some noticeable variation. After the initial lockdown in the beginning of March 2020 and until the end of February 2021, this pattern changes. Indeed, the ratio of upstream versus downstream traffic decreases significantly, with typical values around 9 and very high variation. During the weekdays, this ratio is as low as 8.1. This shows that the relative increase of the upstream traffic is up to 18% higher than that of the increase of the downstream traffic. An independent study that analyzes traffic data from U.S. ISPs reported even higher upstream versus downstream traffic ratios during the pandemic.[2] We attribute this to the increase in remote working and teleconferencing applications that utilize user upstream bandwidth much more than other popular user applications, for example, video streaming and browsing. This is an important observation as ISPs in general allocate way less upstream than downstream capacity to end users. If we see a persistent change in demand from end users to push more traffic toward the Internet, ISPs may need to adapt their handling of subscriber lines. This is a notable result, because last mile capacity is notoriously expensive for ISPs and hard to replace with new technology.

Figure 4. ISP aggregated over 8 hours traffic during October 2019–February 2021: upstream traffic normalized growth (top), and downstream versus upstream traffic ratio (bottom).



4. APPLICATION TRAFFIC SHIFTS

We now turn our attention toward the traffic shifts for different application classes that were expected to be affected by the COVID-19 pandemic, namely Web conferencing applications, Video-on-Demand streaming, online gaming, and traffic that originates from university service networks. We refer to Feldmann et al.[9] for technical details on how we classified traffic in any of these categories.

4.1. Application classes' traffic shift

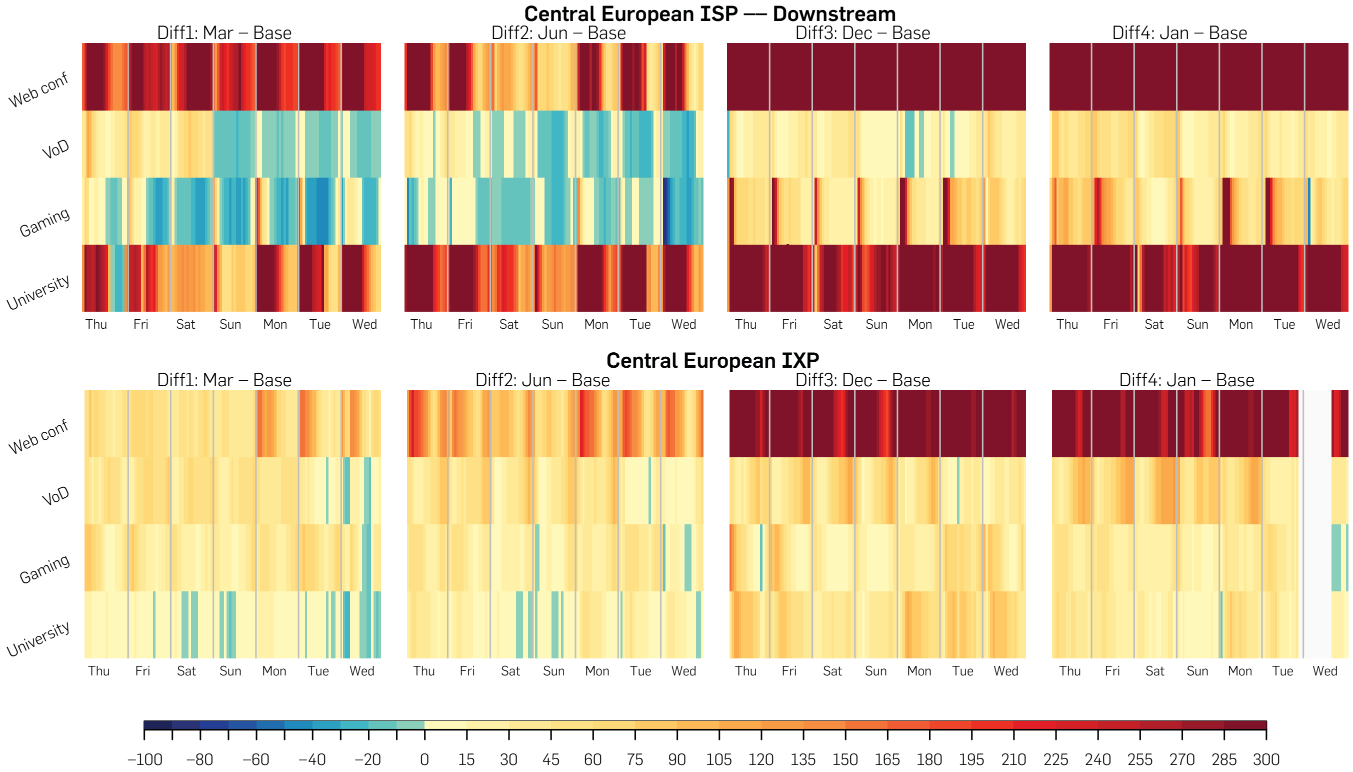
In Figure 5, we visualize two weeks in the Spring and Fall waves, namely the second week in March 2020, June 2020, December 2020, and January 2021, as the difference of the respective week. We compare them to a base week before the initial lockdowns began, that is, February 20–26, 2020. As the traffic classes we are considering show growth way beyond the expected natural increase over one year, we do not factor out that increase. Each column represents one hour of a day. This approach enables quick visual identification of increased/decreased application class usage compared to pre-COVID-19 times. We focus on the observations gathered at the ISP and the IXP in Central Europe (IXP-CE) vantage points.

Web conferencing. Web conferencing applications have seen a dramatic surge during the lockdown periods. In this category, the ISP and IXP-CE experience a large traffic growth in March—right after the first lockdown began—spanning across all hours of the day, especially during weekdays. This trend accelerates in June and culminates in December and January, with an increase exceeding 300% compared to the base week at both vantage points. Notably, in December and January, the extreme growth also persists at weekends. This indicates that not only work life has moved online but private social activities did as well.

Video-on-demand. Video streaming applications' usage shows high growth both in the Spring and Fall waves. Interestingly, the ISP only sees a moderate growth during the lockdown in the first half of March followed by a reduction of volume in the second half of March below the pre-COVID-19 reference time frame. We attribute this to major streaming companies reducing their streaming resolution in Europe by mid-March for 30 days.[18] In the case of the IXP, a similar but not that much pronounced trend can be observed in March. However, there is a significant increase of the traffic related to Video-on-Demand in June, December and January, which exceeds 200% (IXP) and 100% (ISP) for some days, especially on weekends indicating that more people stayed at home during leisure time instead of going outside.

Gaming. The strong growth of gaming applications is more coherent at the IXP vantage point, especially during the day. Although the ISP shows a significant increase during morning hours, it generally leans toward declining in the Spring wave. Note that this effect is mainly caused by unusually high traffic levels in this category during our baseline week in February 2020. The initial download of a game nowadays supersedes the amount of data transferred although playing these high levels may relate to new releases or updates of popular games. Gaming applications, typically used in the evening or at weekends, are now used at any time.

Figure 5. ISP (top), IXP-CE (bottom) heatmaps of application classes' traffic at the ISP, and IXPs during COVID-19 pandemic: spring and fall waves. Each subplot shows the change in the aggregated traffic volume per hour for the respective class compared to the base week in February 2020. White areas mark missing data.



The trend starts to flatten in June—this may in relation with people going on vacation or spending more time outside. The ISP sees an increase up to 300% in gaming-related traffic during the fall wave across all weekdays, but with emphasis to the first half of the day. A similar pattern unfolds at the IXP, but with smaller increases. One explanation for the strong increase at both vantage points in the morning hours is that schools were closed during the fall wave.

University networks. Traffic that originates from such networks behaves similar at both vantage points with the ISP showing a more pronounced trend. Both vantage points see a high increase in traffic especially during the fall wave with a growth of 100% and more. This growth could be attributed to some European educational networks providing video conferencing solutions, which are now being used by customers of the ISP/IXP. In December 2020 and January 2021, most academic collaboration and teaching activities moved to an online setting. This is in line with the smaller surge of activity at weekends.

4.2. VPN traffic shift

Working from home leads to a higher demand for Virtual Private Network (VPN) solutions as employees need to access firewall-protected resources hosted in internal company networks. We identify VPN traffic using a novel technique based on transport port data as well as DNS data.[9] In Figure 6, we show the changes in VPN traffic during the spring and fall COVID-19 waves in 2020. We use five weeks of data from the IXP in Central Europe from February 2020 to January 2021, each in different months, to highlight the differences. February 2020 serves as a baseline, that is, to show the state of VPN traffic before COVID-19 restrictions were enforced.

In March 2020—after the first lockdown restrictions were authorized in Europe—we notice a large increase in VPN traffic during working hours. This growth partially recedes in June 2020 as lockdown restrictions are relaxed again and employees could return to their workplaces. Nevertheless, VPN traffic volume is still over the February 2020 baseline levels. In the Fall wave, VPN traffic increases again but not as high as in March 2020.

We also investigate the *share* of VPN traffic among the total volume across waves. We find that the VPN traffic share remains stable from February to March 2020. This suggests that overall traffic volumes increase, regardless of the application. However, in June 2020, VPN traffic share increases, whereas overall traffic volume decreases. Toward December 2020 and January 2021, we see a slight decrease in the VPN traffic share as the overall traffic gains traction again after the summer holiday.

In summary, we see that VPN traffic increases during working hours since the first lockdown measures were implemented. The increase is higher in the spring wave than in the fall one. This finding aligns with reports indicating that more people in Central Europe were working from home in the Spring wave compared to the fall wave.[23]

4.3. A view from REDIMadrid

In addition to investigating changes at ISPs, IXPs, and mobile networks, we analyze changes at a special type of network: REDIMadrid, a large European academic network. As a large portion of traffic is generated by students and staff being physically on the campus, we expect to see quite drastic changes after lockdown measures are imposed. As a response to the COVID-19 pandemic, the regional government of Madrid announced the closure of the educational system from March 11, 2020 onward for the 2019–2020 academic year. By the end of April, most universities had fully transitioned toward an online-lecturing model. The 2020–2021 academic year followed a semipresence model, allowing for some lectures to take place on campus.

Traffic volume. Figure 7 shows the normalized traffic volume for six different weeks ranging from February 2020 (one week before announcing that the academic system would be closed down) to January 2021. We observe a significant drop in the traffic volume on working days right after the lockdown, with a maximum decrease of up to 55% on Tuesday and Wednesday. This is expected because users no longer use the network from within the campus. Traffic on weekends sees increments of up to 14% on Saturday. Similar to what we see at the IXP-CE and the ISP (cf. Section 3.1), traffic patterns on working days and weekends become more similar in terms of total volume. The traffic volume observed in September and December 2020 reveals an increase with respect to the Spring regime, though the pre-COVID-19 levels are not reached. We observe how traffic on weekends continues to increase, and the difference with work days vanishes even further. We note that the week from January 14, 2021 to January 21, 2021 constitutes an interesting anomaly caused by the Filomena snow storm that brought heavy snow to Madrid for over a week. The effects caused severe mobility limitations for more than two weeks, during which campuses were inaccessible and all lectures were canceled. The effect is noticeable as minimum traffic levels across all categories because the lockdown measures were implemented in February 2020.

Traffic in/out ratio. In the days before the lockdown, incoming traffic was up to 15× the volume of outgoing traffic during workdays. This ratio dropped to around 2× or 3× during lockdown, where traditional weekend versus workday patterns also disappeared. This change of traffic asymmetry might be explained by the nature of remote lecturing and

Figure 6. VPN traffic evolution during the COVID-19 pandemic.

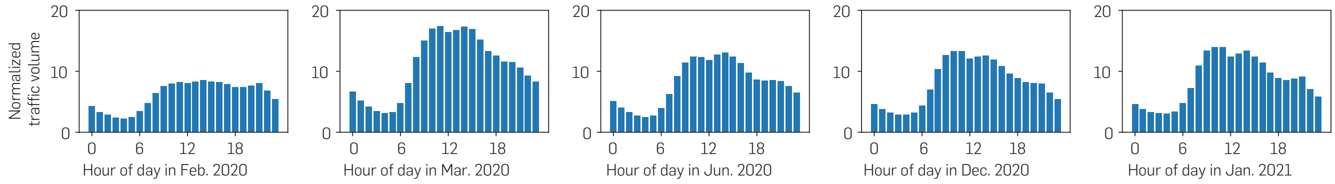
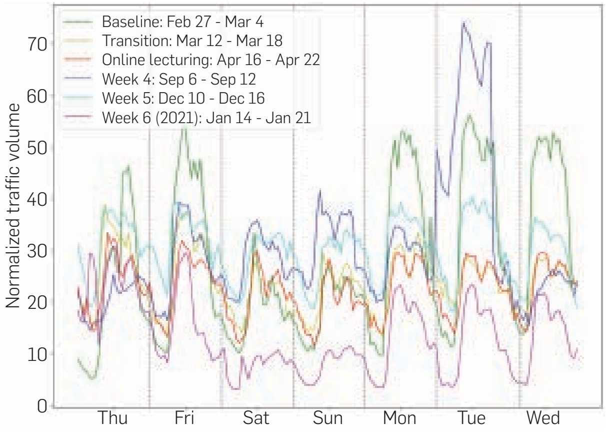


Figure 7. Volume shifts at the REDIMadrid network for selected weeks.



remote working: students and research staff connect to the services hosted at universities to access teaching resources, hence the increase in outgoing traffic. On the other hand, as students and staff no longer access the Internet from the universities, incoming traffic decreases. The semipresence teaching model implemented for the 2020–2021 academic year has resulted in a new intermediary scenario compared to the first COVID-19 wave and the pre-COVID-19 regime. This observation is corroborated at the connection level. Incoming VPN, email, and Web traffic connections remain at high levels compared to February 27, 2020 (5×, 2.3×, 2.4× growth on average, respectively) due to online lecturing and remote working. Outgoing Web traffic and push notification traffic—tightly related to mobile devices—have doubled compared to April 2020 in 2020–2021. However, their overall values are still considerably lower compared to the pre-COVID-19 regime.

5. DISCUSSION

Internet operation during the pandemic: A success story. The COVID-19 pandemic "underscored humanity's growing reliance on digital networks for business continuity, employment, education, commerce, banking, healthcare, and a whole host of other essential services."[10] At the beginning of the pandemic and the first lockdown measures to control its spread, sudden changes in user demand for online services raised concerns for network operators. In fact, the pandemic increased the demand for applications supporting remote teaching and working to guarantee social distancing which manifests itself in our analysis across all vantage points. The Internet could handle this increased load thanks to its original design concept to find efficient routes,[22] the flexibility and elasticity that cloud services offer, and the increasing connectivity of cloud providers.[3, 4, 12, 21, 24] Our results confirm that most of the applications with the highest absolute and relative increases are cloud-based.

Taming the traffic increase. In this paper, we report a traffic increase of more than 20% a week after the lockdown began. This is in line with reports of ISPs and CDNs[6, 11, 16, 17] as well as IXPs.[19] Typically, ISPs and CDNs are prepared for a traffic increase of 30% in a single year period.[3, 5, 14] Although networks perform yearly plannings, the pandemic has created substantial shifts within only a few days. As a result, ISPs either needed to benefit from over-provisioned capacity—for example, to handle unexpected traffic spikes such as attacks or flash-crowd events—or add capacity very quickly. The latter was possible due to the adoption of best practices on designing, operating, and provisioning networks which contributed to the smooth transition to the new normal. Due to the advances in network automation and deployment, for example, automated configuration management and robots installing cross connects at IXPs without human involvement, it was possible to cope with the increased demand. For example, DE-CIX, Dubai, managed to quickly enable new ports within a week for Microsoft, which was selected as the country's remote teaching solution for high schools.[7]

6. CONCLUSION

Despite the disruption due to COVID-19, life continued thanks to the increased digitization and resilience of our society, with the Internet playing a critical support role for businesses, education, entertainment, purchases, and social interactions. In this paper, we analyze Internet flow data from multiple vantage points in several developed countries. Together, they allow us to gain a good understanding of the impact that the COVID-19 waves and the lockdown measures caused on Internet traffic. One year after the first lockdown measures were enforced, the aggregated traffic volume increased by around 40%, well above the typical expected annual growth. Additionally, workday traffic patterns have rapidly changed and the relative difference to weekend patterns has almost disappeared during lockdowns. Applications for remote working and education, such as VPN and video conferencing, experienced traffic increases beyond 200%.

Our study reveals the importance of covering different lenses to gain a complete picture of these phenomena. Additionally, our observations highlight the importance of approaching traffic engineering with a focus that looks beyond Hypergiant traffic and popular traffic classes to consider "essential" applications for remote working. In fact, our study demonstrates that over-provisioning, proactive network management and automation are key to provide resilient networks that can sustain drastic and unexpected shifts in demand such as those experienced during the COVID-19 pandemic. Yet, as the pandemic is still ongoing, it is critical to continue studying the traffic activity to understand usage shifts during these unprecedented times.

Acknowledgments

This work has been partially funded by the Federal Ministry of Education and Research of Germany (BMBF, grants BIFOLD 01IS18025A and 01IS18037A, 5G-INSEL 16KIS0691 and AIDOS 16KIS0975K + 16KIS0976), the Spanish Ministry of Science, Innovation and Universities (grants PID2019–111429RB-C21 and PID2019-111429RB-C22), and by the Comunidad de Madrid (grants EdgeData-CM P2018/TCS-4499 and CYNAMON-CM P2018/TCS-4566, co-financed by European Structural Funds ESF and FEDER), and by the European Research Council (ERC) Starting Grant ResolutioNet (ERC-StG-679158). The authors would like to acknowledge the support offered by David Rincón and César Sánchez (IMDEA Software Institute and REDIMadrid) to access the academic network dataset.

References

1. Boettger, T., Ibrahim, G., Vallis, B. How the Internet Reacted to Covid-19 — A Perspective from Facebook's Edge Network. In *ACM IMC* (2020).
2. Bronzino, F., Feamster, N., Liu, S., Saxon, J., Schmitt, P. Mapping the Digital Divide: Before, During, and After COVID-19. In *TPRC 48* (2021).
3. Labovitz, C. Internet Traffic 2009–2019 (2019). In *APRICOT 2019*.
4. Chiu, Y., Schlinker, B., Radhakrishnan, A.B., Katz-Bassett, E., Govindan, R. Are We One Hop Away from a Better Internet? In *SIGCOMM HotNets* (2015).
5. Cisco. Cisco Annual Internet Report, 2020. https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internetreport/index.html.
6. Comcast. COVID-19 network update, 2020. https://corporate.comcast.com/covid-19/network.
7. DE-CIX. DE-CIX Virtual Get-together - Focus Middle East & Asia 22 Apr 2020, 2020. https://www.youtube.com/watch?v=DfPt10aopns
8. DFN. German National Research and Education Network: COVID-19 Newsticker, 2020. https://www.dfn.de/alle-meldungen-aus-demnewsticker-zur-covid-19-pandemie/
9. Feldmann, A., Gasser, O., Lichtblau, F., Pujol, E., Poese, I., Dietzel, C., Wagner, D., Wichtlhuber, M., Tapiador, J., Vallina-Rodriguez, N., Hohlfeld, O., Smaragdakis, G. The Lockdown Effect: Implications of the COVID-19 Pandemic on Internet Traffic. In *ACM IMC* (2020).
10. ITU. Press release: New 'State of Broadband' report warns of stark inequalities laid bare by COVID-19 crisis (2020). https://www.itu.int/en/mediacentre/Pages/PR20–2020-broadband-commission.aspx.
11. Labovitz, C. Pandemic impact on global Internet traffic. In *NANOG 79* (2020).
12. Labovitz, C., Lekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F. Internet inter-domain traffic. In *ACM SIGCOMM* (2010).
13. Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E.D., Taft, N. Structural analysis of network traffic flows. In *ACM SIGMETRICS* (2004).
14. Leighton, T. Can the Internet keep up with the surge in demand? 2020. https://blogs.akamai.com/2020/04/can-the-internet-keep-up-with-the-surge-in-demand.html.
15. Lutu, A., Perino, D., Bagnulo, M., Frias-Martinez, E., Khangosstar, J. A characterization of the COVID-19 pandemic impact on a mobile network operator traffic. In *ACM IMC* (2020).
16. McKeay, M. Parts of a whole: Effect of COVID-19 on US Internet traffic, 2020. https://blogs.akamai.com/sitr/2020/04/parts-of-a-whole-effect-of-covid-19-on-us-internet-traffic.html.
17. McKeay, M. The building wave of Internet traffic, 2020. https://blogs.akamai.com/sitr/2020/04/the-building-wave-of-internet-traffic.html.
18. Netflix. Reducing Netflix traffic where it's needed while maintaining the member experience, 2020. https://media.netflix.com/en/companyblog/reducing-netflix-traffic-where-its-needed.
19. Sanghani, B. COVID-19 & IXPs. RIPE 80, 2020 https://ripe80.ripe.net/wp-content/uploads/presentations/27-ripe80-covid-ixp-1.pdf.
20. Schilz, B., Maunier, R. Experience on deploying a new remote PoP during COVID-19 restriction. RIPE 80, 2020. https://ripe80.ripe.net/wp-content/uploads/presentations/26-Volterra-Ripe-connect-presentation.pdf.
21. Schlinker, B., Kim, H., Cui, T., Katz-Bassett, E., Madhyastha, H.V., Cunha, I., Quinn, J., Hasan, S., Lapukhov, P., Zeng, H. Engineering Egress with edge fabric: Steering oceans of content to the world. In *ACM SIGCOMM* (2017).
22. Timberg, C. Your Internet is working. Thank these Cold War-era pioneers who designed it to handle almost anything. The Washington Post, April 6, 2020, 2020. https://www.washingtonpost.com/technology/2020/04/06/your-internet-is-working-thank-these-cold-war-erapioneers-who-designed-it-handle-almost-anything/.
23. Wirtschafts- und Sozialwissenschaftliches Institute. Press release: Deutlicher Anstieg: 24 Prozent der Erwerbstätigen arbeiten aktuell vorwiegend oder ausschließlich im Homeoffice (in German), 2021. https://www.boeckler.de/pdf/pm_wsi_2021_02_16.pdf.
24. Yap, K-K., Motiwala, M., Rahe, J., Padgett, S., Holliman, M., Baldus, G., Hines, M., Kim, T., Narayanan, A., Jain, A., Lin, V., Rice, C., Rogan, B., Singh, A., Tanaka, B., Verma, M., Sood, P., Tariq, M., Tierney, M., Trumic, D., Valancius, V., Ying, C., Kallahalla, M., Koley, B., Vahdat, A. Taking the edge off with Espresso: Scale, reliability and programmability for global Internet peering. In *ACM SIGCOMM* (2017).

Anja Feldmann, Oliver Gasser, and Franziska Lichtblau, Max Planck Institute for Informatics, Saarbrücken, Germany.

Enric Pujol and Ingmar Poese, BENOCS, Berlin, Germany.

Christoph Dietzel and Daniel Wagner, DE-CIX, Cologne, Germany and Max Planck Institute for Informatics, Saarbrücken, Germany.

Matthias Wichtlhuber, DE-CIX, Cologne, Germany.

Juan Tapiador, Universidad Carlos III de Madrid, Madrid, Spain.

Narseo Vallina-Rodriguez, IMDEA Networks, Madrid, Spain and ICSI, Berkeley, USA.

Oliver Hohlfeld, Brandenburg University of Technology, Cottbus, Germany.

Georgios Smaragdakis, TU Berlin, Berlin Institute for the Foundations of Learning and Data, Berlin, Germany and Max Planck Institute for Informatics, Saarbrücken, Germany.

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

CAREERS

Australian National University

School of Computing

Tenure Track Lecturer/Senior Lecturer/ Associate Professor

Job no: 540228
Work type: Tenure Track
Location: Canberra / ACT
Categories: Academic
Classification: Level B / Level C / Level D
Salary package:
Level B (Lecturer): $99,809 –$113,165
Level C (Senior Lecturer): $119,844 –$133,202
Level D (Associate Professor): $143,216 –$156,453
Term: Full time, Tenure Track

Position Overview:
These positions are tenure track appointments to contribute to at least one of the Activity Clusters of the ANU School of Computing.

- ► Computing Foundations: software systems, programming languages, compilers, operating systems, software testing, software engineering, computer architecture, computer security, logic and verification, theory of computation, formal methods
- ► Intelligent Systems: artificial intelligence, machine learning, computer vision, natural language processing, robotics
- ► Data Science: data mining and statistical analysis, data storage and management, cloud storage and distributed computing, data engineering, data warehousing, data modeling and visualization, data analytics, database management and architecture, health informatics, data privacy
- ► Computational Science: scientific computing, high-performance computing, large-scale parallel systems, modeling and simulation, bioinformatics, computational biology/chemistry/physics/engineering

Applications are particularly invited from researchers whose breadth of vision reaches across traditional discipline silos. We welcome and develop diversity of backgrounds, experiences and ideas and encourage applications from individuals who may have had non-traditional career paths, who may have taken a career break or who have achieved excellence in careers outside of academia.

To enquire about these positions please contact the Director of the ANU School of Computing, Professor Antony Hosking, T: +61 2 6125 9358, E: director.comp@anu.edu.au.

Position description and application instructions are available at https://jobs.anu.edu.au/cw/en/job/540228/tenure-track-lecturersenior-lecturerassociate-professor-school-of-computing.

Furman University

Open-Rank Tenure Track Professor in Computer Science

The Department of Computer Science at Furman University invites applications for a tenure track position at the Assistant, Associate, or Full Professor rank to begin August 1, 2022. Candidates must have a Ph.D. in Computer Science or a closely related field, and all areas of specialty will be considered. The position requires a demonstrated potential for superb teaching, including the ability to work with a diverse population of students, excellence in scholarly and professional activity involving undergraduates with a broad spectrum of backgrounds and abilities, effective institutional service, and a willingness to work with colleagues across disciplines.

Furman Computer Science professors mentor undergraduates both formally and informally, and strive to build an inclusive student-faculty community in which every member is treated with dignity, and all are welcomed to participate in the life of the department and in the respectful exploration of ideas. This involves regular interactions of both curricular and extracurricular nature. The candidate should show interest in and aptitude for contributing to this endeavor.

The Department of Computer Science confers degrees with majors in Computer Science (B.S. and B.A.) and Information Technology (B.S. and B.A.), an innovative, interdisciplinary program of study. The Department values teaching breadth and versatility, research projects that bridge Computer Science with other disciplines, efforts to provide students with learning opportunities outside the classroom and in the community, contributing to Furman's university-wide First Year Writing Seminar program, and a commitment to applications of algorithmic, computing, and information technology to addressing issues of accessibility, equity, and social justice.

Furman is an Equal Opportunity Employer committed to advancing diversity, equity, and inclusion in all facets of university life, and strives to create an anti-racist community through excellence in teaching, mentorship, and programming. Numerous initiatives and programs are underway or planned to promote these ideals, including: historic projects, dialogue initiatives, the Center for Inclusive Communities, and a proposed major in Africana Studies. The University aspires to create a community of people representing a multiplicity of identities including gender, race, religion, spiritual belief, sexual orientation, geographic origin, socioeconomic background, ideology, world view, and varied abilities. In keeping with our commitment to equity and inclusion, domestic partners of employees are eligible for comprehensive benefits and faculty/staff affinity groups exist to offer support for faculty/staff that identify as LGBTQIA+ and /or Black /African-American.

The successful candidate will have the ability to work with historically underrepresented students, including students of color, and be committed to assisting the university in its continuing efforts to become a model of inclusive excellence.

Applicants should submit a curriculum vitae, cover letter, statement of teaching philosophy and experience, statement of research interests, an official copy of most recent transcripts, and a diversity statement that describes how your teaching, scholarship, mentoring and/or service might contribute to a liberal arts college community that includes a commitment to diversity as one of its core values. Three letters of recommendation should be sent separately upon request. Review of applications will commence on September 1, 2021 and will continue until the position is filled. Questions can be directed to the chair of the Department of Computer Science, Dr. Kevin Treu, at kevin.treu@furman.edu. To submit an application and letters of recommendation, please visit https://furman.wd5.myworkdayjobs.com/en-US/Furman_Careers/job/Main-Campus/Open-Rank-Tenure-Track-Professor-in-Computer-Science_R001166.

UNIVERSITY OF TARTU INSTITUTE OF COMPUTER SCIENCE

Professors in Autonomous Driving Technology

The University of Tartu is an outstanding centre of research and training in **Estonia**, one of the most digitalised countries in the world. Computer science is one of the flagships of the University, placing it among the top 200 worldwide by the Times Higher Education speciality ranking.

As the fastest growing and most international, the **Institute of Computer Science** hosts 100 academic staff and 80 PhD students representing over 30 nationalities and admits over 220 BSc and 220 MSc students annually. It is located in the ultra-modern Delta Centre, one of the truly unique environments for learning, research and innovation in all of Europe.

The common denominator for Estonian IT specialists is their creativity and the ability to ignore the concept of the impossible. Despite its small size, Estonia is home to the original core development teams of many great unicorns of the digital age, including Skype, Playtech, Wise, Pipedrive and Bolt.

In collaboration with Bolt, we have established an Autonomous Driving Lab and seek to extend its scope by ambitious top-level researchers. We are seeking **candidates in Autonomous Driving Technology** in a broad sense, from artificial intelligence and end-to-end driving to mapping and localisation or other enabling infrastructure research and development. We can also host visiting researchers and offer collaborations on the existing technology platform.

The Professor and the Associate Professor positions are both tenured positions. Qualifications for the post of Professor should include a history of strong leadership and vision. For the position of Associate Professor, an ambitious research development plan has to be presented. If in doubt, apply for both categories. The deadline for applications is the 2nd of August.

A competitive salary package of 100-130K and 60K-80K euro for Professor and Associate Professor respectively is offered, less 41% national taxes, depending on qualifications and experience.

For more information, visit **cs.ut.ee/adl** or contact the Head of the Institute, Prof. Jaak Vilo, vilo@ut.ee.

UNIVERSITY OF TARTU
Institute of Computer Science
1632

DOI:10.1145/3466564

Dennis Shasha

Upstart Puzzles
String Me Along

Seeking the ever-elusive shortest path.

See if you can add any A, B, or C after this without repeating a two-letter pattern.
ABBACCA
A

A two-complete string is one in which no two-letter subsequence is present more than once, but appending any letter would violate that condition. Would adding the A make this string two-complete?

CONSIDER THE FOLLOWING solitaire or multiperson game that will be called "String Me Along." You are given a collection of *k* distinct letters, for example, A, B, C, and a number *j*. A well-formed string has no repeats of any substrings of length *j*. Such repeats would be called *j-repeats*.

For example, A B C A C C A B has a two-repeat of A B, but

A B C A C C B A has no two-repeat.

If the string is free of *j*-repetitions, but adding any letter from the collection would cause a *j*-repetition, the string is said to be *j-complete*.

For example, let's say that *j* = 2 and the collection is A, B, C. Here is a two-complete string.

A A B B C C B A C A

It is not possible to extend this further because A has already been followed by A, B, and C.

Warm-Up: The two-complete string above for A, B, and C was of length 10. Is it possible to get a shorter two-complete string for A, B, and C?

Solution to Warm-Up: Here is one example: A B B A C C A A.

Question: What is the length of the shortest two-complete string on A, B, C?

Solution: The shortest length is six. Here is an example: A A B A C A. As in the example, it is not possible to extend this further because A has already been followed by A, B, and C. Further, any two-complete string in which it is impossible to follow A must both end with an A and must include A B, A C, as well as A A so cannot be shorter than 6.

End of Solution to Warm-Up.

A larger *j* gives more possibilities, so the shortest possible string could get longer. But how much longer?

Challenge: Find a three-complete string on A, B, C of length 11 or less.

Solution: Here is one of length 11.

A B C A B B A B A A B

The reason this is complete is that A B has already been followed by C, B, and A. Note that you might think, based on the same reasoning, that

A B C A B B A B A B

is three-complete, but it is not because B A B appears twice so it has a three-repetition already.

This last challenge suggests a game in which players alternate adding letters to the string without causing *j*-repetitions. If some player cannot do so, then that player loses.

Challenge: Suppose that in some game, the string so far is

A B C A A C

It is the Player 1's move now. Can either player force the other to cause a two-repetition?

Solution: Player 1 should not append A next because C A is already present. Player 1 could append B or C, but appending B would be foolish because then Player 2 could put in A next and then Player 1 would not be able to continue. So, Player 1 puts in C, yielding

A B C A A C C

This forces Player 2 to append B allowing Player 1 to append A, yielding

A B C A A C C B A

Player 2 cannot extend this further, so Player 1 wins.

You are ready for the upstarts.

Upstart 1: Is there a three-complete string on A, B, and C of length 10 or less?

Upstart 2: Given *k* letters and some *j*, what is the longest *j-complete* string one can get and what is the shortest?

Upstart 3: Is there a forced winning strategy for either player for *j* = 2 for *k* >= 2 letters?

Upstart 4: Is there a forced winning strategy for general *j* and *k*?

Dennis Shasha (dennisshasha@yahoo.com) is a professor of computer science in the Computer Science Department of the Courant Institute at New York University, New York, NY, USA, as well as the chronicler of his good friend the omniheurist Dr. Ecco.

All are invited to submit their solutions to upstartpuzzles@cacm.acm.org; solutions to upstarts and discussion will be posted at http://cs.nyu.edu/cs/faculty/shasha/papers/cacmpuzzles.html