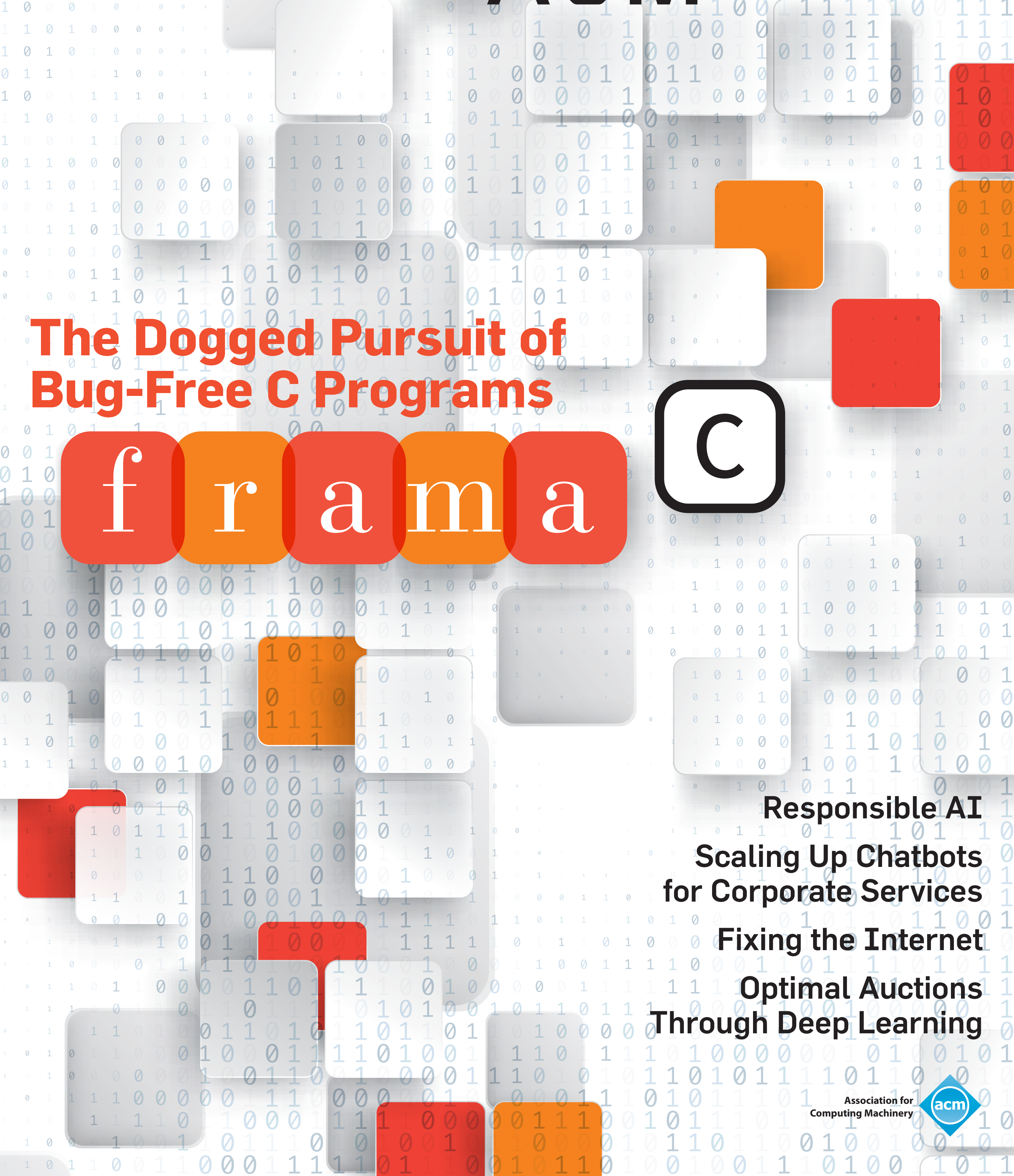
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

08/2021 VOL.64 NO.08

The Dogged Pursuit of Bug-Free C Programs



Responsible AI

Scaling Up Chatbots for Corporate Services

Fixing the Internet

Optimal Auctions Through Deep Learning

Association for Computing Machinery

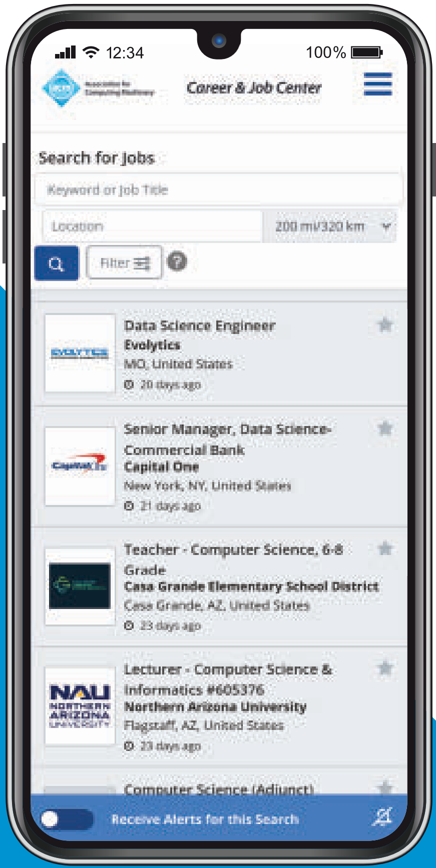
Association for Computing Machinery

Career & Job Center

The #1 Career Destination to find Computing Jobs.

Connecting you with top industry employers.

The new ACM Career & Job Center offers job seekers a host of career-enhancing benefits, including:



12:34 100%

Career & Job Center

Search for Jobs

Keyword or Job Title

Location 200 mi/320 km

Filter

Data Science Engineer
Evolytics
MO, United States
20 days ago

Senior Manager, Data Science-Commercial Bank
Capital One
New York, NY, United States
21 days ago

Teacher - Computer Science, 6-8 Grade
Casa Grande Elementary School District
Casa Grande, AZ, United States
23 days ago

Lecturer - Computer Science & Informatics #605376
Northern Arizona University
Flagstaff, AZ, United States
23 days ago

Computer Science (Adjunct)

Receive Alerts for this Search

- Access to new and exclusive career resources, articles, job searching tips and tools.
- Gain insights and detailed data on the computing industry, including salary, job outlook, 'day in the life' videos, education, and more with our new Career Insights.
- Search and apply to an increased amount of the best jobs at organizations that value your credentials.
- Receive the latest jobs delivered straight to your inbox with **new exclusive Job Flash™ emails**.
- Get a free resume review from an expert writer, listing your strengths, weaknesses, and suggestions to give you the best chance of landing an interview.
- Receive an alert every time a job becomes available that matches your personal profile, skills, interests, and preferred location(s).

Your next job is right at your fingertips.
Get started today!

Visit jobs.acm.org

COMMUNICATIONS OF THE ACM

Departments

5 Cerf's Up
On Communication
By Vinton G. Cerf

7 Letters to the Editor
Tales of Two Turings

10 BLOG@CACM
A Journal for Interdisciplinary Data Science Education
Orit Hazzan and Koby Mike on the need for a journal to cover data science education exclusively.

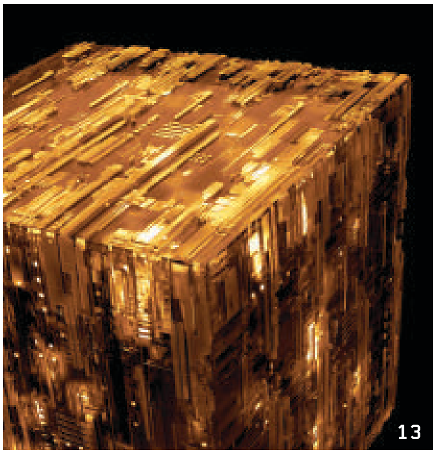
126 Careers

Last Byte

128 Future Tense
Agent Algorithm
Crime-solving computer plays by its own rules.
By Brian Clegg

Association for Computing Machinery
Advancing Computing as a Science & Profession

News



13 **Better Security Through Obfuscation**
The quest to find greater security through obscurity.
By Chris Edwards

16 **Fixing the Internet**
Internet security was once based on trust and needs to be updated.
By Keith Kirkpatrick

18 **The Unionization of Technology Companies**
New unions could change how tech giants engage with their employees.
By Logan Kugler

21 **Upholding ACM's Principles**
Repeated ethical violations ends with membership revocation and ban.
By Marty J. Wolf, Don Gotterbarn, and Michael Kirkpatrick

Viewpoints

24 Economic and Business Dimensions
A European Union Approach to Regulating Big Tech
Considering a new regulatory proposal for addressing digital market competition concerns.
By Georgios Petropoulos

27 Education
What Does Saying That 'Programming Is Hard' Really Say, and About Whom?
Shifting the focus from the perceived difficulty of learning programming to making programming more universally accessible.
By Brett A. Becker

30 Kode Vicious
In Praise of the Disassembler
There is much to be learned from the lower-level details of hardware.
By George V. Neville-Neil

32 Viewpoint
Responsible AI: Bridging From Ethics to Practice
Recommendations for increasing the benefits of artificial intelligence technologies.
By Ben Shneiderman

36 Viewpoint
Science Needs to Engage With Society: Some Lessons From COVID-19
Recent experiences toward communicating science to the general public.
By Carlo Ghezzi

39 Viewpoint
The Domestic Computer Science Graduate Students Are There, We Just Need to Recruit Them
Proven practices to recruit domestic computer science graduate students.
By David Whalley, Xin Yuan, and Xiuwen Liu

IMAGE BY ARLEKSEY/SHUTTERSTOCK

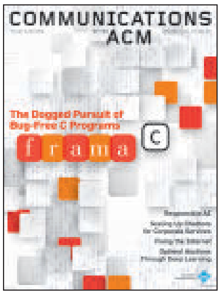
08/2021
VOL. 64 NO. 08

Practice

44 **Biases in AI Systems**
A survey for practitioners.
By Ramya Srinivasan and Ajay Chander

50 **WebRTC: Real-Time Communication for the Open Web Platform**
What was once a way to bring audio and video to the Web has expanded into more use cases than we could ever imagine.
By Niklas Blum, Serge Lachapelle, and Harald Alvestrand

Articles' development led by acmqueue
queue.acm.org



About the Cover:
This month's cover story presents a panorama of Frama-C, a popular platform for C program analysis and verification, including its basic analyzers and their sophisticated uses. Cover image by Andrij Borys Associates.

IMAGES BY: (L) SUPERPUAY/SHUTTERSTOCK; (R) KTSDESIGN

Contributed Articles

56 **The Dogged Pursuit of Bug-Free C Programs: The Frama-C Software Analysis Platform**
A panoramic view of a popular platform for C program analysis and verification.
By Patrick Baudin, François Bobot, David Bühler, Loïc Correnson, Florent Kirchner, Nikolai Kosmatov, André Maroneze, Valentin Perrelle, Virgile Prevosto, Julien Signoles, and Nicky Williams

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

69 **Unveiling Unexpected Training Data in Internet Video**
Using clever video curation and processing practices to extract video training signals automatically.
By Tali Dekel and Noah Snavely

80 **Multimedia Data Delivery Based on IoT Clouds**
A method for reducing delivery delays for multimedia data produced by the Internet of Things.
by Xiaonan Wang and Xingwei Wang

Review Articles

88 **Scaling Up Chatbots for Corporate Service Delivery Systems**
A future-state architectural strategy designed to support chatbot integration with service delivery systems.
By Alistair Barros, Renuka Sindhgatta, and Alireza Nili

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

98 **PL and HCI: Better Together**
Collaborations between two communities have unearthed a sweet spot for future programming efforts.
By Sarah E. Chasins, Elena L. Glassman, and Joshua Sunshine

Research Highlights

108 Technical Perspective
The Quest for Optimal Multi-Item Auctions
By Constantinos Daskalakis

109 **Optimal Auctions Through Deep Learning**
By Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai S. Ravindranath

117 Technical Perspective
eBP Rides the Third Wave of Mobile Health
By Josiah D. Hester

118 **eBP: An Ear-Worn Device For Frequent and Comfortable Blood Pressure Monitoring**
By Nam Bui, Nhat Pham, Jessica Jacqueline Barnitz, Zhanan Zou, Phuc Nguyen, Hoang Truong, Taeho Kim, Nicholas Farrow, Anh Nguyen, Jianliang Xiao, Robin Deterding, Thang Dinh, and Tam Vu

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.

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

ACM COUNCIL

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

BOARD CHAIRS

Education Board
Elizabeth Hawthorne and Chris Stephenson
Practitioners Board
Terry Coatta

REGIONAL COUNCIL CHAIRS

ACM Europe Council
Chris Hankin
ACM India Council
Abhiram Ranade
ACM China Council
Wenguang Chen

PUBLICATIONS BOARD

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

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

Computer Science Teachers Association
Jake Baskin
Executive Director

STAFF

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

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
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

REGIONAL SPECIAL SECTIONS
Co-Chairs
Jakob Rehof, Haibo Chen, and P J Narayanan
Board Members
Sherif G. Aly; Panagiota Fatourou;
Chris Hankin; Sue Moon; Tao Xie;
Kenjiro Taura; David Padua

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:10.1145/3472146

Vinton G. Cerf

On Communication

As I write this, summer is upon us in the Northern Hemisphere. I have just attended an online lecture about non-human species communication, sponsored by the Interspecies Internet project (interspecies.io). While the primary objective of the project is to determine experimentally whether it is possible to demonstrate communication between non-human species, there is also considerable interest in understanding the nature of intraspecies communication. The lecturer, Ofer Tchernichovski, explored years of experience with zebra finches. Of particular interest were their songs and how they propagated through generations of "tutors" and "pupils" among families of finches. Among the interesting observations he made was a concern that we sometimes bring preconceived but unwarranted notions to science. For example, consider the way in which we might analyze bird songs. We make audio recordings and spectral Fourier diagrams of the songs. We segment these vocalizations as if they might represent phonemes, but our segmentation could be inappropriately influenced by what we know of human speech.

Linguists have learned a great deal about human speech, how it is produced, and how the phonemes give structure to utterances. Whether we can apply such structural assumptions to bird songs is a matter for research. Tchernichovski points out that an alien arriving on planet Earth, even if it is capable of sensing human speech, might not have any idea how to segment sounds into phonemes and words. Language is a concept that organizes sound into phonemes, words, and sentences representing structures that follow grammatical rules and from which semantic content can be derived. The alien might not have any a priori clue as to how human languages are expressed, parsed, and give rise to semantic meaning. If the alien itself has language, it might adopt a protocol for human language discovery, starting, for example, with self-identification.

This made me wonder whether some of our unsupervised pattern-detection mechanisms used in machine learning could be used to discover plausible phonemes in bird songs without making arbitrary judgments as to how to segment the patterns by pitch, duration, and repetition. In another lecture by Con Slobodchikoff, the warning signals of prairie dogs were analyzed and correlated with the arrival of identifiable predators. Evidence was offered from which we could infer the signals had sufficient descriptive capacity to distinguish among various predators, their size, location (for example, ground direction or airborne), and possibly other characteristics.

Other efforts analyzing gray parrots (Irene Pepperberg), dolphins (Diana Reiss), and whale songs, (Roger Payne), among others, are also seeking to discover the structure and semantics of these species' signals. What is particularly interesting to me is whether any of these vocalizations or other signals (gestures, postures) can bridge the species gap and be understood by unlike species. That it is possible seems not to be in doubt. One has only to speak with a happy dog owner to be convinced the owner is fairly certain the dog has a significant ability to respond to a vocabulary of commands or queries. I was certain our beagle knew what "walk" and "ice cream" meant.

At the heart of the interspecies Internet effort lies the question: "Can we discover language in the vocalizations and/or gestures of non-human species?" And within that question lies another question: whether rich machine learning methods can demonstrate that interspecies communication is possible. Of course, this presupposes there is a sharable semantics between two or more species. There is some evidence that the warning calls of one species may be understood and even propagated to others. To go between species, the information might have to be transduced into new vocalizations, gestures, or visible displays. What I find most exciting about this exploration is the power and diversity that computing brings to the problem. We can imagine using a wide range of computational tools, statistics, machine learning, and perhaps newer methods to analyze intraspecies signals and to use them to facilitate interspecies communication. C

At the heart of the interspecies Internet effort lies the question: "Can we discover language in the vocalizations and/or gestures of non-human species?"

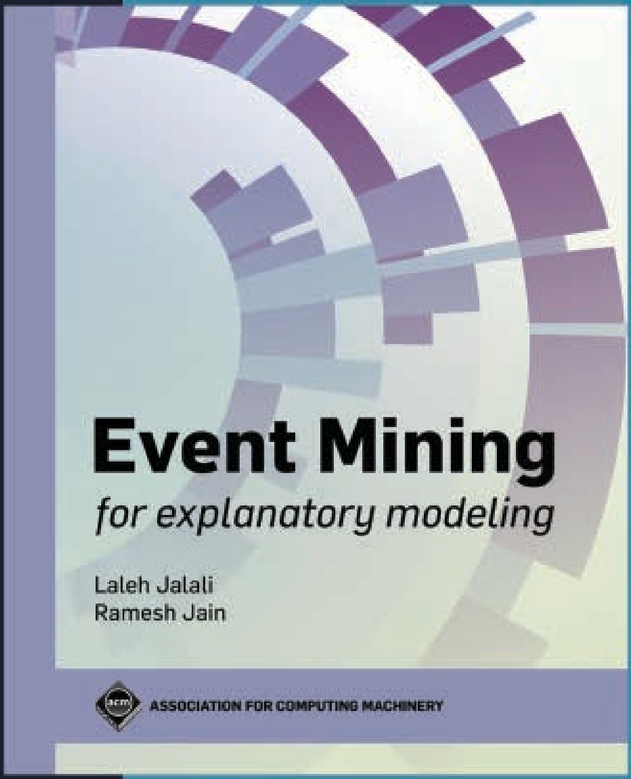
Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

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

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

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

DOI:10.1145/3469876

Tales of Two Turings

IN THE JUNE issue of *Communications*, Editor-in-Chief Andrew A. Chien suggested in his Editor's Letter (p. 5) that ACM consider bestowing two A.M. Turing Awards per year. Reader reactions to his idea included the following:

Immediately upon reading your June Editor's letter, my reaction was "No!" because I thought two annual awards would reduce the stature of each and minimize the honor to recipients and even to Alan Turing. But I was hasty in forming my opinion. I reread your argument and changed my opinion—I now believe we need to think even bigger.

The number "two" suggests a division between hardware and software. But our discipline, as you note, has grown far and wide. It is more complex than this dichotomy. I propose *four* categories, understanding that not all need be awarded in a given year. These are: hardware design or fabrication; software languages or algorithms; networks or communications; and ethical or sustainable practices. The last category may appear out of place. With respect, here is where I disagree with your interpretation that the A.M. Turing Award is for contributions by leading *researchers* [my emphasis.]

The published criterion for the Award states: "The contributions should be of lasting and major technical importance to the computer field." There is no mention of academia vs. industry. ACM has for a long time recognized the practice of computer science as well as pure research. An excellent example is Charlie Bachman, industry practitioner, and the 1973 Turing recipient.

Applications of ethics and the practice of sustainable methods can still meet the "major technical importance" criterion while encouraging meaningful contributions by practitioners and researchers alike. (I would include in this contributions to teaching and advancing knowledge, as evidenced by Aho and Ullman.)

Additionally, ethical and sustainability dilemmas are rapidly appearing with the rise of AI and ML, resource scarcity, manufacturing practices, and cryptocurrency mining. We, the ACM, should encourage practical solutions to these global issues that affect humankind.

I welcome dialogue, discussion, and debate of these ideas. Thank you for initiating the process.

Gary Rector, Cave Creek, AZ, USA

Very bad idea to have two Turing awards. That award is for truly outstanding computer scientists. Do not cheapen it.

Alexander Simonelis, Montreal, Canada

Perhaps it would be a good idea to take all the other ACM awards and make them all Turing Awards. The Turing Award could be more like the Oscars, and given by Category at a big event.

Maurice van Swaaij, Brooklyn, NY, USA

I think the award is not truly international at this point and efforts should be made in that direction.

Alex Thomasian, Pleasantville, NY, USA

You asked for input about it being time for "two annual Turing Awards" each year. I feel very dubious about this change. Why, you may ask? Because, unless something is done to dramatically alter the way that the Awardees are selected, ACM will likely just be *doubling* the number of Caucasian males who have overwhelmingly received the Association's highest honor.

Certainly Aho and Ullman were long-deserving of this award. Indeed their colleague, John Hopcroft, received his in 1986. As for myself, as someone who both studied from and later taught from the AHU collection of books, I was glad to see the remaining duo's pioneering efforts take their place this year alongside many other notable gurus and graybeards over the decades. I am not intending to disparage the work of anyone on the Turing Award list. But, as a female computer scientist, it is yet another disappointment to see two more white guys get selected for this recognition.

ACM can and must do better than this. There are many outstanding Black, Indian, Asian, Latinx, female, non-binary, and other computation pioneers, deserving of this award, who continue to not be represented here. What message is ACM giving to youth? Would Turing himself, who was prosecuted and subjected to chemical castration because he was homosexual, be proud to see such overt bias perpetuated, year after year after year by the ACM in his name? I think not.

So, if you are intent upon giving out *two* Turing Awards each year, going forward, then the rule *must* be that only *one* of them can be given to a Caucasian male. You can start by making sure that your nominating committee has a preponderance of non-Caucasian non-males doing the search and selection of awardees. Perhaps you should have a hiatus of five years where *zero* Caucasian males can be eligible to receive the Turing Award, so that the other races and genders have a chance to catch up. Just look harder. There are plenty of highly qualified candidates in the world that would fit these expansive demographics.

I hope you will seriously consider this suggestion.

Rebecca Mercuri,
Senior Life Member of ACM
Hamilton, NJ, USA

Editor-in Chief's response:

Its great to see that thoughtful and provocative responses that my proposal to have two Turing Awards evoked! There really are many reasons why increasing the number of Turing Awards would advance computing as a field, community, and recognize more of our remarkable leaders. Keep the letters coming!

Andrew A. Chien, Chicago, IL, USA

Getting Down to Basics

I was thrilled when I opened my email weeks ago to discover Aho and Ullman were the ACM A.M. Turing Award 2020 recipients, and I was delighted to see

Al Aho and Jeff Ullman on the cover of June *Communications* when it came in the mail today.

But I was dismayed when reading the article by Neil Savage, which contains two glaring mistakes.

Ullman joined the faculty of Princeton, not Columbia, from Bell Labs. I know because I had the great fortune to learn from Professor Ullman at Princeton—he broke open the palace gates of computer science for his students, and sparked a lifelong passion for CS.

The article states that Aho and Ullman are being awarded for *their contribution to both the theory and practice of computer languages*, but it is for so much more than that—they were also instrumental in the foundations of the Analysis of Algorithms, as the Turing citation indicates. As students at Princeton, we enjoyed the inherent beauty of a gorgeous and complex subject explicated by some of the finest minds and authors, using the seminal text *The Design and Analysis of Computer Algorithms* by Aho, Hopcroft, and Ullman, known affectionately as AHU. In addition, Ullman was also instrumental in advancing database theory, as well as codifying VLSI theory, as his textbooks and courses at Stanford University will attest.

Also noteworthy, Ullman was able to bring Unix 6th edition to Princeton—in 1975, we were set up in a small lab in the Engineering Quad on a PDP-11/45. With great support from Bell Labs, Peter Eichenberger, Tom Lyon, Eric Schmidt, and I started a port of Unix to the IBM 370 architecture. None of that would have been possible without JDU.

It is fitting that Aho and Ullman are recognized for their many seminal contributions—let's appreciate all the great work they did—it is a lot more than "just compilers."

Joseph P. Skudlarek,
ACM Senior Member
Lake Oswego, OR, USA

Who Programmed the Early Generations of Commercially Available Computers in the U.S.?

Thomas J. Misa's "Dynamics of Gender Bias in Computing" (June 2021), judging that the composition of the programming workforce before 1970 has been mischaracterized, overlooks important sources in establishing a valid workforce composition "starting point." The article looks at SHARE (IBM "scientific" users), CDC Coop (CDC 1604 "scientific" users), UNIVAC USE (UNIVAC 1100-series "scientific" users), Burroughs CUBE (mostly Burroughs 5000 "scientific" users) and Mark IV (Informatics Mark IV utility users of IBM "business-oriented" machines) user groups as a basis for estimating the proportion of women programmers in the programming workforce. Even after assigning weights to these numbers according to installed computer base, the great weight given "scientific" installations invalidates their use as a representation of the total work force. If user groups are a valid source, IBM GUIDE and COMMON for "business-oriented" users must be included. There were far more "business-oriented" installations than "scientific" installations. Installed computer base was reported monthly in the early days of *Datamation* magazine and issues are available at the Computer History Museum.

Civilian and military "business-oriented" data processing departments already existed in significant numbers before the first computers were sold. They were filled with—mostly IBM—CAMs (punched card accounting machines), programmed with wired plugboards. Programming was complex; in 1955 I took an intensive six-week introductory course in the Air Force. In military and civilian businesses these machines were handled and programmed by male high school graduates. They were as skilled as automobile mechan-

Civilian and military "business-oriented" data processing departments already existed in significant numbers before the first computers were sold.

ics but happened, upon graduation, to take their first job in an office rather than a repair shop. "Business-oriented" computers were usually introduced into shops that already had PCAM-based data processing departments. The natural source of programmers was the pool of plugboard wirers who understood the existing applications and their PCAM process, and understood their old machines and were not in awe of their new ones.

An interesting study of "business-oriented" first-generation computers and their programming would look at USAF Air Materiel Command, headquartered at Wright-Patterson AFB, Dayton. AMC was probably the biggest single user of first-generation computers in the U.S. Their larger depots used large-scale IBM 702s and 705s. Smaller depots used IBM 650/RAMACs, which were complex configurations for their time. The depots were staffed mostly by civilians who, I assume, fit the profile of "business-oriented" programmers I described.

Ben Schwartz, Brooklyn, NY, USA

Author's response:

Thanks to Ben Schwartz for pointing out IBM GUIDE and COMMON to fill out "business-oriented" users. Additional diverse user groups would enrich our understanding. I now believe researchers should sum gender probabilities and not merely tally men's and women's names, since names change gender. See names in Mattauch et al.[1]: 'Johnnie' born in 1925 has 0.39 probability of being female; but 0.17 if born in 1975 (Social Security Administration data). Same years, 'Leslie' switches from p(F)*=0.08 to 0.86. There are plenty of accomplished computer science Leslie's, both genders, that should be accurately identified.*

Reference

[1] Mattauch, S. et al. A bibliometric approach for detecting the gender gap in computer science. *Commun. ACM 63*, 5 (May 2020), 74–80; https://doi.org/10.1145/3376901

Thomas J. Misa, Lopez Island, WA, USA

Keying Users

As a UX professional, I read the article "AZERTY amélioré: Computational Design on a National Scale" (*Communications*, Feb. 2021) with interest. I compliment the authors on thorough research and excellent use of graphics and summaries in the article. The authors focus on translating goals such as "facilitate typing and learning" into quantifiable objective functions. I miss objective information about how the redesign affected real users' performance and satisfaction. I would also have liked to know what the authors learned from usability tests of the keyboard, which are unavoidable in user-centered design. In my interpretation of user-centered design, users should not just be involved in a public comment phase but throughout development. I welcome algorithmic optimization of usability, but I recommend that algorithms never replace real user involvement.

Rolf Molich, Denmark

Authors' response:

We agree that user involvement is important. As we describe, different stakeholders were involved in the design process, painstakingly and throughout. Moreover, rigorous empirical research is the very foundation of the models that our objective functions utilize. Keyboards are not traditionally evaluated in "usability tests," but in carefully controlled transcription tasks. Historically, predictions made by predictive models—such as the one that is the basis of our optimizer—have agreed very well with empirical measurements (see Zhai et al.[1]).

Reference

[1] Zhai, S. et al. Performance optimization of virtual keyboards. *Human-Computer Interaction 17*, 2–3 (2002), 229–269.

Antti Oulasvirta, on behalf of co-authors, Denmark

Rewriting the Fine Print

As covered in many recent *Communications* articles and commentary, there has been much handwringing about recent ransomware attacks. The current solutions turn to sanctions, government-public partnerships, and the new defense agency—the Cyber Command. Lost in all this is the simplest, most effective defense—the EULA.

What is the EULA? It is the End User License Agreement we must accept when we install software. That "agreement" absolves the software creator of any liability for software errors and provides unlimited access to your information generated with the software.

There is no technology or consumer protection for stupidity or inattention.

Congress could eliminate the EULA providing instead consumer protection like those for medical equipment and automobiles. Those agreements say if these devices do not work, the manufacturer will recall them and make them work.

Think how much more bulletproof software would be if the engineers and managers of software companies were personally liable for the safety of their products. Sure the tech companies would whine about how much more expensive their products would be. Then think about how your personal identity and bank account would be protected without all the virus checkers and firewalls you have to buy to make up for the lack of safety in software.

BTW, there is no technology or consumer protection for stupidity or inattention. When you are lured by phishing to click on something that loads hacking software into your computer, it's on you. There could be however congressional protection from the spam email and phone calls we get.

Bert Laurence, Life Member of ACM
Palo Alto, CA, USA

Editor-in Chief's response:

Thanks for the observation, particularly timely as governments around the world are reconsidering what are appropriate responsibilities and regulations for "tech companies" both big and small.

Andrew A. Chien, Chicago, IL, 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

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:10.1145/3469281 http://cacm.acm.org/blogs/blog-cacm

A Journal for Interdisciplinary Data Science Education

Orit Hazzan and Koby Mike on the need for a journal to cover data science education exclusively.

Orit Hazzan and Koby Mike
An Open Call to Establish an Interdisciplinary Data Science Education Journal
https://bit.ly/3eBrHgk
April 26, 2021

If your research is in data science and you wish to present your insights and research about the teaching of data science, you probably have no choice but to try submitting your manuscript to a journal that deals with a data science-related topic and is dedicating a special issue to data science education, or to an educational conference on a topic related to data science.

One exception is the *Journal of Statistics Education* published by The American Statistical Association, which in January changed its name to the *Journal of Statistics and Data Science Education*. The change reflects the growing attention and importance attributed to data science education. This journal, however, still holds the statistics perspective and targets the statistician community.

In other words, no journal exists today that deals exclusively with data science education, let alone highlights data science education from an interdisciplinary perspective.

In this blog post, we describe our vision for a journal that would focus on data science education from the interdisciplinarity perspective. In this blog, we will call it *The Interdisciplinary Journal of Data Science Education* (*IJDSE*).

Motivation

Data science is a new interdisciplinary field of research focused on extracting value from data and integrating knowledge and methods from computer science, mathematics and statistics, and the domain knowledge of the data. As an interdisciplinary field, it receives attention from each of the disciplines that comprise it, so the potential of its interdisciplinary nature has not been fully exhausted and should be explored more deeply. We suggest this applies also to the *educational* aspect of data science.

Accordingly, *IJDSE* would aim to explore the *interdisciplinarity* of data science from the education perspective. Such an examination would educate future learners better and more effectively than when the educational perspective is taken up separately by each of the disciplines that make up data science. This statement relies on the working assumption that when data science education is explored from each angle separately, learners are not exposed to the comprehensive picture and interdisciplinary nature of data science needed to use it meaningfully in their personal and professional lives.

Target Audience

We anticipate professionals and researchers from all disciplines would be interested in the subjects the journal would cover. In addition, since data science is relevant to all industries today and they all need to educate all their employees on how to use data to add value to the firm, they probably would find *IJDSE* relevant for their growth.

We anticipate a data science education community would form and grow in the coming years, a trend reflected, for example, by the growing number of data science programs recently established in K–12.

Accordingly, IJDSE would:

- Target the three communities that comprise data science: computer science, math and statistics, and the various data domains (the natural sciences, social science, and digital humanities), and within these communities, scholars interested in data science education.

▸ Explore data science education for all levels" from K–12, through undergraduate and graduate programs, to academic scholars and industry professionals.

Scope and Topics

To capture the interdisciplinarity of data science education, *IJDSE* would encompass a broad topic list and welcome all research methods. It would publish research papers, case study reports, opinion papers, work-in-progress papers, and commentaries on the following topics:

▸ Data science education for all levels: K–12, undergraduate as major or minor, graduate, research, and industry.

▹ Data science curriculum and study programs.

▹ Design of data science study programs.

▹ The structure of data science study programs:

▫ Connections between data science education and education in mathematics, statistics, computer science, and the domain of the data.

▸ Data science education pedagogy.

▹ Pedagogies and approaches to teaching data science to diverse populations.

▹ Pedagogy of interdisciplinary education bridging the humanities, social sciences, sciences, and other disciplines.

▹ Teaching methods, tools, and practices of data science education:

▫ Assignments and tutorials.

▫ Class activities.

▫ Data visualization and animation tools.

▫ Data science project management.

▫ Assessment.

▫ Applications in various domains.

▫ Pedagogical challenges, opportunities, and risks.

▸ Learners' cognitive and social processes.

▹ Cognitive and behavioral perspective on data science education.

▹ Cognitive and social biases in data science.

▹ Metacognition.

▹ Learner difficulties.

▹ Learning styles and models.

▹ Diversity, representation, and equality; recruitment and retention of underrepresented groups in data science.

▸ Teacher preparation for data science.

▹ Teacher training programs.

▹ Communities of data science educators.

▸ The essence of data science education .

▹ Is data science education different from other educational fields?

▹ Policy: Data science for all; Should everyone take an introductory data science course?

▹ The interdisciplinary challenges of data science education.

▹ Characteristics of data scientists.

▹ Ethical issues of data science.

▹ Connection between data literacy, data science education, and related topics.

▹ University/public/for-profit and non-profit collaborations in data science education.

▸ Specific topics in data science education – representative list .

▹ Machine learning.

▹ Artificial intelligence.

▹ Data mining.

▹ Bayesian statistics.

▹ Historical perspective.

▹ The data science cycle.

In her 2020 article[1], Jeannette M. Wing asks, "What will data science be in 10 or 50 years? The answer to this question is in the hands of the next-generation researchers and educators."[a]

In this post, we addressed the educational aspect of Wing's answer by proposing our vision for a journal dedicated to data science education that would highlight the interdisciplinary nature of data science and address the growing community of data science educators from a wide range of disciplines, organizations, and fields of research. *IJDSE* would be a place where such scholars would be able to share their practice and research-based expertise in data science education. Thus, beyond contributing to the creation of a new interdisciplinary field of research, *IJDSE* would support the creation of an important, influential worldwide community of data science education researchers.

As an *interdisciplinary* journal, *IJDSE* would enable scholars from different fields of research to learn from each other, improve their professional work, and educate 21st-century professional data scientists and non-scientist citizens to use data in a responsible, ethical, meaningful way. Furthermore, since potential contributors to *IJDSE* are the pioneers of data science education, they would have the potential to influence the creation of new curricula and study programs.

We note *IJDSE* would aim to address challenges of data science education we identified in our post "Ten Challenges of Data Science Education" (https://bit.ly/2R2c9cu), including the interdisciplinary nature of data science and its diverse body of learners and teachers.

We suggest it is urgent to establish an educational journal for data science that approaches data science education from the interdisciplinary perspective. To that end, we call on all potential stakeholders to take up the gauntlet and establish such a platform for all data science educators. We would be happy to assist in any such initiative in its establishment process and to offer our comprehensive perspective on data science education.

Further Reading

1. Anderson, P. et al. An undergraduate degree in data science: Curriculum and a decade of implementation experience. In *SIGCSE 2014 - Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (2014), pp. 145–150; https://bit.ly/2SsJLAP
2. Gould, R. Mobilize: a Data Science Curriculum for 16-Year-Old Students. (2018); https://bit.ly/3gqW2xx
3. Heinemann, B. et al. Drafting a data science curriculum for secondary schools. In *Proceedings of the ACM Int. Conf. Proceeding Ser. (Nov. 2018), 1–5; https://bit.ly/3wgv6Yb*
4. Havill, J. Embracing the liberal arts in an interdisciplinary data analytics program. *SIGCSE 2019—Proceedings of the 50th ACM Tech. Symp. Comput. Sci. Educ., (2019), 9–14; https://bit.ly/3xelSf5*
5. Khuri, S., Vanhoven, M., and Khuri, N. Increasing the capacity of STEM workforce: Minor in bioinformatics. In *Proceedings of the Conference on Integrating Technology into Computer Science Education*, ITiCSE (Mar. 2017), 315–320; https://bit.ly/3gfzVv4.
6. Mike, K., Hazan, T., and Hazzan, O. Equalizing Data Science Curriculum for Computer Science Pupils. *Koli Calling—International Conference on Computing Education Research 20 (Nov. 2020), 1–5; https://bit.ly/35emPYJ*
7. Tartaro, A., and Chosed, R.J. Computer scientists at the biology lab bench. *In SIGCSE 2015—Proceedings of the 46th ACM Tech. Symp. Comput. Sci. Educ. (2015), 120–125; https://bit.ly/3xd6wre.*

Orit Hazzan is a professor at the Technion's Department of Education in Science and Technology, whose research focuses on computer science, software engineering, and data science education. **Koby Mike** is a Ph.D. student in the Technion's Department of Education in Science and Technology under the supervision of Orit Hazzan.

[1] Wing, J.M. Ten research challenge areas in data science. *Harvard Data Science Review 2*, 3 (Sept. 2020); https://bit.ly/3znFnna

2017
Volume 6, Number 1

ACM Transactions on
Human-Robot Interaction

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

Association for
Computing Machinery

Advancing Computing as a Science & Profession

Science | DOI:10.1145/3469283

Chris Edwards

Better Security Through Obfuscation

The quest to find greater security through obscurity.

LAST YEAR, THREE mathematicians published a viable method for hiding the inner workings of software. The paper was a culmination of close to two decades of work by multiple teams around the world to show that concept could work. The quest now is to find a way to make indistinguishability obfuscation (iO) efficient enough to become a practical reality.

When it was first proposed, the value of iO was uncertain. Mathematicians had originally tried to find a way to implement a more intuitive form of obfuscation intended to prevent reverse engineering. If achievable, virtual black box (VBB) obfuscation would prevent a program from leaking any information other than the data it delivers from its outputs. Unfortunately, a seminal paper published in 2001 showed that it is impossible to guarantee VBB obfuscation for every possible type of program.

In the same paper, though, the authors showed that a weaker form they called iO was feasible. While iO does not promise to hide all the details of a logic circuit, as long as they are scrambled using iO, different circuits that perform the same function will leak the same information as each other; an attacker would not be able to tell which implementation is being used to provide the results they obtain.

IMAGE BY ARLEKSEY/SHUTTERSTOCK

"Our motivation in defining the notion of iO was that it escaped the impossibility result for VBB. However, we had no idea if iO could be constructed, and even if it could be constructed, would it be useful for applications," says Boaz Barak, George McKay professor of computer science in the John A. Paulson School of Engineering and Applied Sciences at Harvard University, and co-author of the 2001 paper on VBB.

Whatever its utility, for more than a decade iO seemed to be out of reach. A major breakthrough came in 2013, when a team came up with a candidate construction and described a functional-encryption protocol that could be built on top of it. This was quickly followed by a slew of proposals for applications that could make use of iO.

One possible application is functional encryption, which makes it possible to selectively hide parts of the same program or data from different users through the use of different decryption keys. This could provide far more fine-grained protection than conventional encryption, where a single key unlocks everything encrypted with it. Other more exotic forms of encryption enabled by iO include deniable encryption, where a user could provide a false key that appears to work but does not reveal information secured by a true key.

Huijia Lin, associate professor in the Paul G. Allen School of Computer Science and Engineering at the University of Washington, points to the possibility of efficient secure multiparty communication, which is difficult to implement using conventional cryptography. "We want multiparty communications, where the overhead to achieve security is so small that it's as easy as insecure communication. In principle, with iO, you can come up with versions where this is possible."

The 2013 paper demonstrated a plausible technique for delivering iO, but the novel techniques it employed to obfuscate programs could not guarantee they would not leak too much information. Similar to cryptography, mathematically guaranteed obfuscation relies on mathematical constructs, such as one-way functions, that are practically impossible to reverse without knowledge of the keys used to encode them. An ongoing problem for iO implementors is finding constructs considered secure that, at the same time, provide enough expressive power to transform real-world programs into a form that does not leak information unexpectedly. It was an uphill struggle that took another seven years of work by multiple groups. Often a paper would present a plausible mixture of techniques that would almost as quickly be demonstrated as insufficient to the task.

Amit Sahai, Symantec Chair professor of computer science and director of the Center for Encrypted Functionalities at the University of California at Los Angeles (UCLA), who worked on the 2001 and 2013 papers, says the cat-and-mouse game of iO constructions being presented and then broken paved the way to a solution. "The process was very important in building our understanding," he says.

A key breakthrough came last year with the publication of a paper that was the result of a collaboration between Lin, Sahai, and UCLA Ph.D. student Aayush Jain, which was based on assumptions they consider to be well-founded, though some of them are novel in the field of cryptography. "We showed how to construct iO from problems that have been around for at least a decade," Sahai says.

The paper rests on the assumed security of four mathematical problems that the authors claim have well-established histories. Some, such as problems based on the elliptic curves used in cryptography, have been widely used. They also found a technique that has not been heavily explored cryptographically, but which seems to offer a high degree of protection against information leakage. Mathematician Richard Hamming proposed the idea of random linear codes for error cor-

ACM News

Semantics Beats Syntax

IBM, Google, and Microsoft all are poised to release semantic engines (algorithms using the meaning of words) to supplement their current syntax engines (using the spelling of words). Their common goal is to extend their natural language processing (NLP) capabilities into engines that rival human semantics (our understanding of what language, words/sentences, mean).

Today's syntax-only engines are blind to the meaning of keywords used to ascertain results. A human understands that "where Alan Turing was born" means the same as "the birthplace of Alan Turing" and "the town where Alan Turing was delivered as a baby." Their syntax differs, but each phrase's meaning, or semantics, are identical ("London" is the answer to all three). People understand this immediately, but computers—not so much.

Consequently, all three companies are developing algorithms that understand the meaning of words. Google and Microsoft are both building semantic engines that add metadata to sentences (Google) or words (Microsoft) using clusters of processors running multiple deep neural networks (called transformers, which use massive parallelization).

IBM

For its semantic engine, IBM chose to augment neural networks with symbolic logic, reducing the number of examples it requires to learn. Said Forrester Research principal analyst Kjell Carlsson, "IBM's semantics uses a much more efficient encoding of knowledge, enabling high performing enterprise use-cases to be built with significantly smaller training examples."

In addition, said Carlsson, "IBM's neuro-symbolic approach enables higher accuracy with less training data, plus it also enables engineers to 'teach' a model logical relationships that domain experts know to be true, which is far more efficient than having these relationships be learned by transformers."

Google

Google and Microsoft have both released free test versions of their semantic transformers. Google's, called Semantic Experiences, tackles four separate application domains, plus a roll-your-own capability.

Google's demos include "Verse-by-Verse," a semantic "experience" that composes poetry; "Talk-to-Books," which answers queries based on statements found in current books; "Semantris," a word-association game, and the free-form "Create Your Own Semantic Experience" tool.

Microsoft

Microsoft aims to release the first semantic-based commercial product. Using word-level granularity in meaning encoding works best, according to Microsoft's Luis Cabrera-Cordon, a group program manager for Azure, who describes Microsoft's "semantic search on Azure [as offering] the best combination of search relevance, developer experience, and cloud service capabilities."

Forrester Research's Carlsson said, "The biggest recent advancements in AI have been in (deep) learning, which has opened up the world of unstructured data (vision, text, voice, logs) for analysis at scale, but what we really want is both learning and knowledge. Learning enables us to update and acquire new knowledge, and knowledge makes learning more efficient, governable, and valuable. What makes these new deep learning-infused semantic methods exciting is their potential to deliver both, dramatically expanding not just NLP, but all machine learning use-cases."

—R. Colin Johnson is a Kyoto Prize Fellow who has worked as a technology journalist for two decades.

rection 70 years ago, in which a message is encoded using a matrix of values that are generated randomly. Since then, scientists have searched unsuccessfully for an efficient way to reverse the process without knowledge of how the data was encoded. That, in turn, led to the conjecture that performing such decoding efficiently is hard, and that the constructs could be employed to help build iO implementations. Sahai stresses that to defeat this conjecture, it would take a mathematical breakthrough that has eluded communication scientists for decades.

Though it rests on assumptions that are on firm footing, a stumbling block of the Jain, Lin, and Sahai proposal is the complexity of the construction. "This is just the first construction where the pieces finally connected to form a secure scheme. But we have all these steps, all these transformations we have to make to the program to obfuscate it," says Sahai. "Each step introduces a huge overhead."

Estimates of some older work on iO illustrate the computational complexity gap that researchers need to bridge. One paper published in 2016 based on techniques now considered insecure showed that even a simple 80-bit point function that is zero for all inputs except one would consume more than 10GB of memory and take three minutes to execute. Sahai says the overhead of their current scheme is so high, it is not practical to even estimate it.

Barak says the computational overhead is unlikely to be insurmountable. "In cryptography, we've had examples, such as multiparty secure computation and probabilistically checkable proofs, where the initial constructions were almost comically inefficient, but over time people have improved them by 20 or so orders of magnitude," he says.

Though computational overhead is an issue and may mean practical applications will not appear for over a decade, Lin says it is equally important to create a construction using fewer or simpler assumptions. A more compact approach would improve confidence in iO as a building block for secure computation. Numerous groups are now looking to see what can be distilled out of the existing work to create a construction that is conceptually simpler.

> **"Often you go back and realize you didn't need certain steps. We are in that tightening mode, and also alternative-finding mode, but we just don't know how long it will take."**

"Often you go back and realize you didn't need certain steps. We are in that tightening mode, and also alternative-finding mode," Sahai says. "But we just don't know how long it will take."

The main question to be answered in future constructions is which assumptions will provide a way forward for reducing complexity and overhead while ensuring acceptance of iO is on a firm footing.

One approach that has been taken by multiple groups over the past couple of years is to try to let an iO construction rest on a single-core mathematical problem. A major candidate for that is the Learning With Errors (LWE) problem developed by computer scientist and mathematician Oded Regev more than a decade ago, for which he was awarded the 2018 Gödel Prize.

LWE already forms the basis for lattice-based cryptographic systems and is being actively pursued because it is generally considered to be safe from attack by quantum computing. Barak says LWE has conceptual similarities to the random linear codes used in the work published by Jain, Lin, and Sahai, which makes it seem a viable approach.

Yet that is not necessarily the path iO will take.

Jain says the direction taken in work following the 2020 paper has led him and his colleagues to focus instead on building iO without LWE. Lin points out that noise that helps LWE maintain security in conventional cryptography leaks information that could compromise iO. Lin says the assumption that underpins the random linear codes seems to have peculiar properties that are worth exploring further.

Though there may be skepticism in the cryptographic community about the more-novel assumptions that iO seems to rely on, at least for the time being, some of that may simply be attributed to their relative novelty. Lin points to the way that papers from several decades ago used to justify why they used Diffie-Hellman key exchange, whereas today it is widely accepted.

"People's confidence in an assumption tends to grow over time, and with the number of papers that use the assumption," Lin says.

Sahai believes further work may revive the concept of VBB obfuscation. "The impossibility result was overinterpreted by the community at large," he says, on the basis that though the 2001 paper ruled out VBB obfuscation for all software, many common forms of software could yet be candidates. That could in turn make it possible to run software on untrusted machines without fear of the code being reverse-engineered from its code.

Such applications, including those of iO, likely lie some distance into the future.

Further Reading

Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., and Yang, K.
On the (Im)possibility of Obfuscating Programs
Advances in Cryptology (2001). 2010 revision: https://www.boazbarak.org/Papers/obfuscate.pdf

Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., and Waters, B.
Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits
54th Annual Symposium on Foundations of Computer Science (FOCS), (2013)

Jain, A., Lin, H., and Sahai, A.
Indistinguishability Obfuscation from Well-Founded Assumptions
IACR Cryptology ePrint Archive: 1003 (2020) https://eprint.iacr.org/2020/1003

Brakerski, Z., Döttling, N., Garg, S., and Malavolta, G.
Candidate iO From Homomorphic Encryption Schemes
IACR Cryptology ePrint Archive: 394 (2020) https://eprint.iacr.org/2020/394

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

Technology | DOI:10.1145/3469287

Keith Kirkpatrick

Fixing the Internet

Internet security was once based on trust and needs to be updated.

FEW PEOPLE PAY much attention to how the electrical grid works until there is an outage. The same is often true for the Internet.

Yet unlike the electrical grid, where direct attacks are infrequent, vulnerabilities and security issues with the Internet's routing protocol have led to numerous, frequent malicious attacks that have resulted in widespread service outages, intercepted and stolen personal data, and the use of seemingly legitimate Web sites to launch massive spam campaigns.

The Internet is an interconnected global network of autonomous systems or network operators, like Internet service providers (ISPs), corporate networks, content delivery networks (such as Hulu or Netflix), and cloud computing companies such as Google and Microsoft Cloud. The Border Gateway Protocol (BGP) is used to ensure data can be directed between networks along the most efficient path, similar to how a GPS navigation system maintains a database of street addresses and can assess distance and congestion when selecting the optimal route to a destination.

Each autonomous system connected to the Internet has an Internet Protocol (IP) address, which is its network interface, and provides the location of the host within the network; this allows other networks to establish a path to that host. BGP routers managed by an ISP control the flow of data packets containing content between networks, and maintains a standard routing table used to direct packets in transit. BGP makes routing decisions based on paths, rules, or network policies configured by each network's administrator.

BGP was first described in a document assembled by the Internet Society's Network Working Group in June 1989 and was first put into use in 1994. BGP is extremely scalable, allowing tens of thousands of networks around the world to be connected together, and if a router or path becomes unavailable, it can quickly adapt to send packets through another reconnection. However, because the protocol was designed and still operates on a trust model that accepts that any information exchanged by networks is always valid, it remains susceptible to issues such as information exchange failures due to improperly formatted or incorrect data. BGP can also be at the mercy of routers too slow to respond to updates, or that run out of memory or storage, situations that can cause network timeouts, bad routing requests, and processing problems.

Aftab Siddiqui, senior manager of Internet technology at the Internet Society, says the initial BGP protocol was conceived by experts at research institutions, defense organizations, and equipment vendors. "When they designed [BGP], it was based on the premise that everybody trusts each other," Siddiqui says. "Fast-forward 30 years, I'm pretty sure we cannot claim that anymore."

IMAGE BY DIYAJYOTI/SHUTTERSTOCK

As such, BGP is also vulnerable to BGP hijacks or inadvertent IP address leaks, in which route and IP address information can be deliberately intercepted, redirected, or dropped, simply by the advertisement of incorrect or corrupted routes via the BGP protocol. All a malicious actor needs to do is announce a route to IP prefixes that it doesn't own, thereby funneling traffic to its own servers where it can do whatever it pleases with that data, including stealing personal, business, or financial information, or launching cyberattacks from that hijacked IP address. Further, because the base BGP protocol accepts all route advertisements as legitimate, traffic to those legitimate IP prefixes can be routed through to that malicious actor's site until someone notices and fixes the error.

BGP has been updated to include tools to validate the originator of these routing messages, as well as filter out known malevolent IP addresses, but not every network operator is using those tools. Just as a tiny, undetected hole can sink an entire ship, a single security lapse used in an attack can shut down entire networks. In fact, BGP hijacks such as that described here are frequent, with an average of 14 attacks per day between mid-January and June 2020, according to the Internet Society.

Leading the charge to increase the focus on routing security is the Internet Society's Mutually Agreed Norms for Routing Security (MANRS) working group. MANRS announced in December 2020 a new task force charged with defining and publishing an updated set of actions and metrics to measure the progress of networks adopting the more-robust routing security tools and practices.

Siddiqui, who serves as the MANRS project lead, says that while the number of autonomous systems or networks has swelled to more than 70,000 worldwide, the baseline BGP protocol is still working, thanks to its ability to grow as the number of networks rises. "BGP is very scalable, so nobody wants to make any changes in the baseline of the protocol, because it works perfectly fine, except for the trust issue," Siddiqui says. He adds that additional processes have been put into place to make routing more secure, but networks need to use them.

While MANRS has more than 500 network-operator participants, many more are simply not using the tools. "That's why we started this initiative, to engage with the network community," Siddiqui explains, noting that the message to all network operators is to "be sure that you implement the best practices to secure the global routing tables."

A key tool that has been implemented is Resource Public Key Infrastructure (RPKI), instead of solely relying on Internet Routing Registries (IRRs), where network operators store information about their routing policies and routed prefixes. While filtering rules are still useful in ensuring only valid routes are accepted from neighboring networks, the need to constantly maintain and update records is both labor- and time-intensive, often requiring networks to reach out to other networks to validate information.

On the other hand, RPKI, a distributed public database of cryptographically signed records containing routing information supplied by autonomous systems or networks, is considered to be the ultimate "truth" for network information, according to Siddiqui. RPKI is carried out by a process known as route origin validation (ROV), which uses route origin authorizations (ROAs)—digitally signed objects that fix an IP address to a specific network or autonomous system—to establish the list of prefixes a network is authorized to announce.

To conduct the validation of network prefixes, a third-party validation software is run to establish an RPKI-to-BGP router session, which downloads the ROAs in the various repositories, verifies their digital signatures, and then makes the results available for use in the BGP workflow. Once validated, an ROA can be used to generate route filters, and other networks are then able to access these records and use them to validate BGP announcements they receive as accurately identifying their origins.

While RPKI use has grown over the past few years, most of that adoption has been by large ISPs such as AT&T, NTT, and Cogent, announcing they are performing origin validation. But in order to ensure complete protection, all operational networks will need to register their routes to enable hijack protection. A key member of the initiative is Google which, to date, has registered more than 99% of its routes in the RPKI, and has announced plans to deploy ROV this year to ensure any invalid routes are rejected.

Google is among the large networks that has participated in MANRS-led task forces to incorporate better security practices, to ensure BGP can continue to serve as a secure routing protocol. "We can keep tackling this [within Google], but why don't we make it easier for these other players who are now emerging and have more interdependence on the BGP infrastructure than they realize," says Royal Hansen, vice president of security at Google. "So it was a chance to use Google's position in the market and our experience in these kinds of problems to see what we could do to make it easier for others so that we close those final gaps in the way BGP routing works."

Just as a tiny hole can sink an entire ship, a single security lapse used in an attack can shut down entire networks.

Another way Google is helping to promote stronger RPKI and other BGP security controls is via its peering portal, which is a way for Google to assess how well its peers (other networks that have shared BGP routing information with Google to let traffic flow in a more efficient, stable way) are implementing BGP best practices. The peering portal provides Google with a way to share potentially invalid route information, show peers their RPKI status, and also flag peers when they are not applying the proper filtering or other safeguards to routes shared with Google.

Google's lead-by-example approach is helpful in raising awareness of the steps needed to ensure BGP can be used securely; it may be able to coerce other large ISPs and networks to adopt these best practices simply because so much traffic flows through Google's networks.

"If we did not participate in this initiative, the probability of its success would be pretty low," says Bikash Koley, vice president of Google Global Networking (GGN). Koley said that in order to truly close the holes inherent in BGP, there is still a significant amount of engineering work required, which means small ISPs must dedicate resources to these types of initiatives, and that may be difficult to achieve quickly. That is one reason why Google has been working with MANRS to publish information about how it implements route filtering, which should make it easier for smaller networks that want to peer with Google to simply piggyback on an established approach.

However, it is not just Google that is working to implement better practices, such as moving to RPKI or implementing ROV. "There's Netflix, Akamai, Microsoft, Cloudflare, and other top providers," Siddiqui says. "So, if you want to pair with them, then [smaller] operators will need to fix their registry information, and only then will they be able to talk to [the large content delivery networks and cloud providers]."

While getting the majority of the 70,000+ global networks on board with using RPKI and other best practices for network routing is a high bar to clear (and one that likely will take years to achieve), Hansen says it is imperative to close this loophole.

"Sometimes we do say the attacker only has to get it right once, and the defender has to be right all the time," Hansen says, noting the challenges faced by security systems and professionals. Further, the distributed ownership and management of Internet networks adds to the challenge.

"One of the challenges with the Internet is that you have ISPs of all sizes," Koley says. "You have really tiny ones, as well as really large ones, and they are located all around the world. And I would say that this system is as strong or as weak as the weakest link." C

Further Reading

BGP4 (Currently adopted version; IETF RFC 4271)
https://www.rfc-archive.org/getrfc.php?rfc=4271#gsc.tab=0

MANRS for Network Operators
https://www.manrs.org/isps/

A Border Gateway Protocol, Network Working Group RFC 1105, The Internet Society, https://tools.ietf.org/html/rfc1105

What is Border Gateway Protocol?
https://www.youtube.com/watch?v=A1KXPpqlNZ4

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

Society | DOI:10.1145/3469285

Logan Kugler

The Unionization of Technology Companies

New unions could change how tech giants engage with their employees.

In late 2018, thousands of workers walked out of Google offices around the globe to protest the company's handling of sexual harassment accusations against prominent executives.

The same year, hundreds of Salesforce employees signed a letter to CEO Marc Benioff protesting the fact the company sold products to U.S. Customs and Border Protection.

Also in the headlines was an effort by some Microsoft employees to protest the company's bid for work on the U.S. Department of Defense's Joint Enterprise Defense Infrastructure (JEDI) project. In a letter to Microsoft CEO Satya Nadella, the employees wrote, "many Microsoft employees don't believe that what we build should be used for waging war."

Tech employee activism is nothing new, but the momentum generated by the 2018 wave of protests was. Three years later, the momentum from that activism has resulted in the first formal technology unions.

Technology unions are new labor organizations that full-time and contract employees at major tech companies are attempting to form or have successfully formed. These unions fight for traditional issues that unions in other industries fight for, like better wages, hours, and working conditions. Yet given the high number of well-paid tech workers, they also engage in a new type of activism around the morality of tech companies' operating practices and business relationships.

Tech unions represent a new twist on an existing form of worker organization, and they're looking to disrupt the status quo of major tech companies like Google.

Tech workers at the May Day March in San Francisco, CA, USA.

PHOTO COURTESY OF TECH WORKERS COALITION FACEBOOK PAGE

"The time and energy of working people have built tech companies into some of the most valuable entities on the planet," says Liz Shuler, Secretary-Treasurer at the AFL-CIO, the largest federation of unions in the U.S.

"Tech workers have produced innovations that are changing the course of history—and made their bosses rich in the process. They deserve to take home a fair share of the enormous value they create everyday, and they deserve to be treated with dignity on the job."

A New Phenomenon

One of the most significant early tech unionization successes happened in January of this year. That is when the Alphabet Workers Union was announced, with the mission to protect workers at Google's parent company.

The union was organized in secret for a year before the announcement, and has more than 800 members as of this writing, including full-time employees, temps, vendors, and contractors. Typically, unions negotiate with a company over a contract or a single issue for the majority of employees at a company. The Alphabet Workers Union, in contrast, is a minority union, which means it represents only a fraction of employees, and lobbies for them across a range of issues.

"Our long-term goals are to build and consolidate power for workers," says Parul Kohl, executive chair of the union. "We want to ensure workers can push for real, sustainable, structural changes at the company and actually win them, whether it is about the kinds of contracts Google accepts, issues around employee classification, wages and compensation, or sexism and racism in the workplace."

For instance, Kohl says the union recently filed an Unfair Labor Practice complaint on the behalf of a Google datacenter contractor who was suspended for discussing pay with her coworkers. The contractor was brought back to work within a week.

While Alphabet is the highest profile tech firm to have its own union, it may soon have company.

The Alphabet Workers Union sprung out of a larger campaign by the Communications Workers of America (CWA), an affiliated union of the AFL-CIO, to organize workers in technology, gaming, and digital sectors.

That effort, the Campaign to Organize Digital Employees (CODE), also spurred employees at Medium, a publishing platform created by Twitter co-founder Evan Williams, to unionize. More than 70% of employees at the company have expressed their support. Like the Alphabet Workers Union, the Medium Workers Union is organizing around broad support and protections for workers, rather than a single issue or list of demands.

"Our affiliated unions have been making enormous inroads across the tech sector," says Shuler.

However, that is not the case at every tech firm. A unionization effort earlier this year by workers at an Amazon warehouse in Bessemer, AL, drew attention from labor groups, tech companies, and even the president of the United States. Workers at the warehouse voted overwhelmingly against joining the Retail, Wholesale, and Department Store Union (RWDSU). The final vote result was 1,798 workers against unionizing, and just 738 in favor.

Uncertain Effectiveness

Tech union effectiveness over time could depend on a number of factors, says Jerry Davis, a professor of management and organizations at the University of Michigan who studies corporate power.

One factor is an individual union's focus. The Amazon union was focused on more traditional union issues such as wages, hours, and working conditions. Membership in traditional unions focused on these conditions has been in decline for decades. However, says Davis, efforts like the Alphabet Workers Union have "evolved out of concerns with the company not living up to its vaunted social values."

The values-based approach has the advantage of riding highly publicized issues around tech company practices, policies, and customers, which affect wide swaths of employees based on their beliefs, not their economic situation.

"This builds on the momentum of the prior three years of activism at Google and elsewhere, where workers have risen up to demand changes both in who their firms do business with, and how," he says.

> **"With the broad shift to work-from-home, we might see employment dispersed more globally, which makes labor organizing much harder."**

The controversial 2020 firing of Google employee Timnit Gebru offers one example of how company decisions around values fuel worker activism.

Gebru, an artificial intelligence (AI) ethics researcher and co-founder of industry diversity organization Black in AI, worked as a co-lead on Google's Ethical Artificial Intelligence Team. She claims she was fired because she refused to withdraw a research paper on how Google speech technology could create disadvantages for marginalized groups. Two other engineers quit over the firing.

Gebru's firing is one of the unfair company practices that Parul says the Alphabet Workers Union was created to fight.

"Some of our co-workers are working in-person during the pandemic, making $15 an hour with no hazard pay, AI ethics researchers continue to be retaliated against, and Google has still not met most of the demands of the 2018 walkout," she says.

"Alphabet Workers Union represents a counterbalance to those changes—workers at the company recognize these harms, and together, we have the ability to fight them."

Being able to lobby both for traditional employment issues and values-based issues could create wide appeal for tech unions. But the pandemic could make it hard to be effective on either front, says Davis.

"Before COVID, I would have been optimistic that the unions would have a strong effect because local tech talent is at such a premium," he says. "But with the broad shift to work-from-home, we might see employment dis-

ACM Member News

AT THE INTERSECTION OF NLP AND CYBERSECURITY

Bonnie Dorr, a leading researcher in the field of natural language processing, is associate director and senior research scientist at the Institute for Human & Machine Cognition (IHMC) in Ocala, FL, an independent research institute of the state university system in Florida.

Dorr received her bachelor's degree in computer science from Boston University, and earned her master's and Ph.D. degrees in computer science at the Massachusetts Institute of Technology.

She taught at the University of Maryland, College Park (UMD) for 24 years, and is the former associate dean of the university's College of Computer, Mathematical and Natural Sciences. She continues to serve as a professor emerita in the Institute for Advanced Computer Studies at UMD.

In 2011, Dorr joined the U.S. Defense Department's Defense Advanced Research Projects Agency (DARPA), and spent three years there before joining IHMC in 2014.

Dorr's research focus includes deep language understanding and semantics, large-scale multilingual processing, and summarization, as well as cross-language divergence detection, machine translation, paraphrasing, and automatic evaluation metrics.

"More recently, I have been moving in the direction of straddling human language technology and cybersecurity," Dorr says. "It's what I call Cyber-NLP, which is cybersecurity and natural language processing."

Cyber-NLP will help figure out what people's intentions are when they speak or write to determine the underlying motivation and attitudes toward certain issues, she says. "I think it's going to take 5 to 10 years to get a machine to understand what peoples' intentions and meanings are," Dorr explains. "That's where I am headed."

—John Delaney

persed more globally, which makes labor organizing much harder."

A Murky Future

Given the new forms that tech unions are taking, it is difficult to know what the future holds for unionization at tech firms, given the diversity of conditions and efforts.

On the one hand, companies like Amazon clearly have massive leverage when it comes to traditional union demands like wages and working conditions at warehouses. As in some traditional union battles, conflicts over physical conditions can result in companies moving the physical location of warehouses and infrastructure to more business-friendly areas.

On the other, minority unions focused on advocating for workers across a range of issues, like the Alphabet Workers Union, could attract broad, diverse support from digital workers and physical labor alike.

Political pressure could also have an impact.

U.S. politicians on both sides of the aisle endorsed the Amazon unionization effort, including President Joe Biden and Republican senator Marco Rubio. Though that effort failed, it highlighted growing criticism from across the political spectrum about the size and power of big tech companies.

The AFL-CIO, for one, sees the broader unionization effort as just getting started.

"That trend is continuing to accelerate across the industry, and organizers from across the labor movement are responding to the organic energy tech workers have for a collective voice on the job," says Shuler.

It is no accident that tech unionization efforts are moving fast, planning as they go, and are unafraid to break things.

In January 2021, the AFL-CIO launched its Technology Institute, a think tank to help the labor movement address the future of work and tackle issues created by new technology. The Institute is designed to give workers a voice in how technological innovation is used to augment labor. From a unionization perspective, part of the Institute's purpose is to "connect labor organizers and workers everywhere innovation is happening."

It is no accident that tech unionization efforts are moving fast, planning as they go, and are unafraid to break things. This is the same playbook tech giants used to grow into some of the world's brightest, most successful firms.

"The beauty of unions is that they can take on whatever form, priorities, and tactics the members choose," says Shuler. "The constant is that they give those members a seat at the table."

Further Reading

Conger, K.
Hundreds of Google Employees Unionize, Culminating Years of Activism, *The New York Times*, Jan. 2021, https://www.nytimes.com/2021/01/04/technology/google-employees-union.html

Greenhouse, S.
'We deserve more': An Amazon warehouse's high-stakes union drive, *The Guardian*, Feb. 2021, https://www.theguardian.com/technology/2021/feb/23/amazon-bessemer-alabama-union

Jamieson, D.
Labor Groups and Progressives Urge Biden to Support Amazon Union Drive, *Huffpost*, Feb. 2021, https://www.huffpost.com/entry/labor-groups-and-progressives-urge-biden-to-support-amazon-union-drive_n_6037c18cc5b67259f89438e5

Paul, K.
Two Google engineers quit over company's treatment of AI researcher, *The Guardian*, Feb. 2021, https://www.theguardian.com/technology/2021/feb/04/google-timnit-gebru-ai-engineers-quit

Schiffer, Z.
Workers at Medium are unionizing, *The Verge*, Feb. 2021, https://medium.com/s/story/an-open-letter-to-microsoft-dont-bid-on-the-us-military-s-project-jedi-7279338b7132

Selyukh, A.
It's a No: Amazon Warehouse Workers Vote Against Unionizing in Historic Election, *NPR*, Apr., 2021, https://www.npr.org/2021/04/09/982139494/its-a-no-amazon-warehouse-workers-vote-against-unionizing-in-historic-election

Logan Kugler is a freelance technology writer based in Tampa, FL, USA.

Milestones

ACM Recognizes 5 for Service to Computing

ACM recently recognized five individuals for their exemplary service to the computing field. Working in diverse areas, the 2020 award recipients were selected by their peers for recognition of their long-standing efforts that have strengthened the community.

Andrew McGettrick, a professor at the U.K.'s University of Strathclyde, has been awarded the Karl V. Karlstrom Outstanding Educator Award for his scholarship and tireless volunteer work and contributions, which have fundamentally improved rigorous computer science as a field of professional practice and as an academic pursuit.

Jennifer Tour Chayes, a professor at the University of California, Berkeley, received the ACM Distinguished Service Award for her effective leadership, mentorship, and dedication to diversity during her career of computer science research, teaching, and institution building

Imperial College London professor Chris Hankin received the Outstanding Contribution to ACM Award for his fundamental contributions to ACM Europe, and for bringing a European perspective to critically important ACM committees and activities.

Richard Anderson, a professor at the University of Washington, was awarded the ACM Eugene L. Lawler Award for Humanitarian Contributions within Computer Science and Informatics for contributions bridging the fields of computer science, education, and global health.

Marc Rotenberg, an adjunct professor at the Georgetown University Law Center, and founder and president of the nonprofit Center for AI and Digital Policy, received the ACM Policy Award for his long-standing, high-impact leadership on privacy and technology policy.

ACM Statement | DOI:10.1145/3473051

Marty J. Wolf, Don Gotterbarn, and Michael Kirkpatrick

Upholding ACM's Principles

Repeated ethical violations ends with membership revocation and ban.

In response to serious violations against ACM's Code of Ethics and Professional Conduct, the ACM Council voted unanimously to revoke the ACM membership of Tao Li, a professor of computer engineering at the University of Florida, at its meeting on June 11, 2021. The Committee on Professional Ethics (COPE) recommended this action to Council after considering the evidence it received concerning Li's repeated violations of the ACM's Code of Ethics (https://www.acm.org/code-of-ethics). Council's action demonstrates ACM's commitment to advancing computing as a profession and as a service to society. ACM is not alone in this commitment. Indeed, other professional organizations have adopted ACM's Code of Ethics indicating their support of its values and the positive impact its Principles afford the computing community.

Both ACM and IEEE received complaints about Li's actions surrounding two computer architecture conferences: The 2019 IEEE International Symposium on Computer Architecture (ISCA) and the 2017 ACM Architectural Support for Programming Languages and Operating Systems (ASPLOS). A Joint Investigation Committee (JIC) was convened in early 2020 and a team of professional investigators were hired. As a result of the investigation, JIC filed an ACM Code of Ethics violation complaint against Li, submitting as evidence the investigators' final report. COPE reviewed the evidence and determined that Li willfully violated scientific research integrity standards. Quite simply, Li orchestrated an attack on the ethical computing values expressed in the ACM Code of Ethics and most other codes of scientific conduct.

In one case, Li's actions involved a paper submitted to ISCA '19 authored by his graduate student, Huixiang Chen. Evidence exposed dozens of messages that "make clear that Dr. Li intentionally breached the peer review process for the paper in multiple ways: he repeatedly shared the reviewers' names and their scores of the paper with Chen; he manufactured support for the paper by asking Chen to draft messages for paper reviewers to post on the conference's software platform; he then passed those reviews on to the reviewers; and, in two instances, the reviewers posted the reviews that Chen had written at Li's direction." Further, the evidence showed that Li coerced Chen to publish the paper after Chen raised concerns that the work contained incorrect or falsified results.

In another case, Li shared his reviewer credentials for the 2017 ASPLOS conference paper submission system with some of his students to enable them to download confidential draft materials. Further, the investigation uncovered evidence that Li inappropriately approached reviewers in connection with other conferences.

Throughout the investigation, Li knowingly and intentionally proposed obvious false theories and spurious procedural arguments in order to thwart it.

In the end, the evidence showed that Li deliberately and repeatedly undermined the peer review process of at least three conferences, and he had actively encouraged others to engage in activities to support and facilitate these attacks. Li knowingly interfered with efforts to maintain the integrity of research and publication processes, and he violated the community standards expressed in ACM's Code of Ethics. His actions demonstrated a lack of respect for, and an unwillingness to abide by, the Code's Principles, and he intentionally organized an assault on them for his own benefit.

Based on the clear and convincing evidence that Li flagrantly violated several Principles of the Code, ACM terminated Li's involvement with all ACM-affiliated or ACM SIG-affiliated activities (including conference organizing or program committees) both in person or remotely. He is banned from publishing articles in ACM-affiliated conference proceedings or journals until 2036.

Principle 4.2 of the Code has us "[t]reat violations of the Code as inconsistent with membership in the ACM." The evidence presented by the JIC clearly indicated that Li's behavior was not only inconsistent with the Code, but that he actively worked to attack the core values and ethical practice expected for members of the ACM. Consequently, COPE further recommended that Li's ACM membership be revoked, and Council voted in agreement to officially cut all ties with him.

ACM policy generally prevents COPE from commenting publicly on cases it reviews, but in this case Council directed COPE to make a public statement. ACM's Code indicates that "the public good is the paramount consideration" in decision making and describes key ethical principles as essential elements of computing professionalism. The ACM is committed to upholding these Principles. By providing some details of this case, we hope to prevent future behavior that undermines these values. ACM members should promote the public good through computing and should actively and publicly resist behavior which threatens it.

Marty J. Wolf and **Don Gotterbarn** serve as co-chairs and **Michael Kirkpatrick** is a member of the ACM Committee on Professional Ethics.

ACM ON A MISSION TO SOLVE TOMORROW.

Dear Colleague,

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

For over 70 years, ACM has helped computing professionals be their most creative, connect to peers, and see what's next, and inspired them to advance the profession and make a positive impact.

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

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

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

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

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

Sincerely,

Gabriele Kotsis
President
Association for Computing Machinery

DOI:10.1145/3469104

Georgios Petropoulos

▸ Marshall Van Alstyne, Column Editor

Economic and Business Dimensions

A European Union Approach to Regulating Big Tech

Considering a new regulatory proposal for addressing digital market competition concerns.

The rise of big tech in the last two decades is remarkable. The so-called GAFAM (Google, Apple, Facebook, Amazon, and Microsoft) now top other companies in terms of world market capitalization—replacing oil, gas, and financial services. Potential abuse of market power has prompted European Union (EU) to regulate big tech.

The competitive landscape in technology markets in the 1990s and 2000s gave rise to platform markets dominated by Internet giants.[5,6] Concerns have grown that big tech firms will abuse their market power to protect and improve their market position at the expense of their competitors, consumer choice, and welfare.

Digital platform business models rely on the interaction of different user characteristics. Business users, advertisers, app developers, and external producers of goods and services use platform intermediaries' services to interact with individual users/consumers and vice versa. Advertisers are attracted by platforms that can efficiently match them with many consumers. App developers design software applications for app stores that put them in touch with significant demand. Consumers visit online marketplaces where they can find variety and quality of supply. The largest platforms become gatekeepers for given Internet activities such as online search, social media exchange, and e-commerce. They control the gates that agents of each side must pass through to reach the other side.

The largest platforms become gatekeepers for given Internet activities such as online search, social media exchange, and e-commerce.

To reduce abuse of platforms' market power that allows gatekeepers to extract unwarranted rents, regulators in Europe have responded through antitrust investigations. In the Google Shopping case,[2] Google Search was fined for self-preferencing, because by promoting its own services to consumers, it distorted competition by making it difficult for competitors to reach consumers. The practice of tying Microsoft's Internet Explorer to the Windows operating system was similarly found to be anticompetitive.[3] Amazon is currently under investigation for using non-public business data from independent sellers

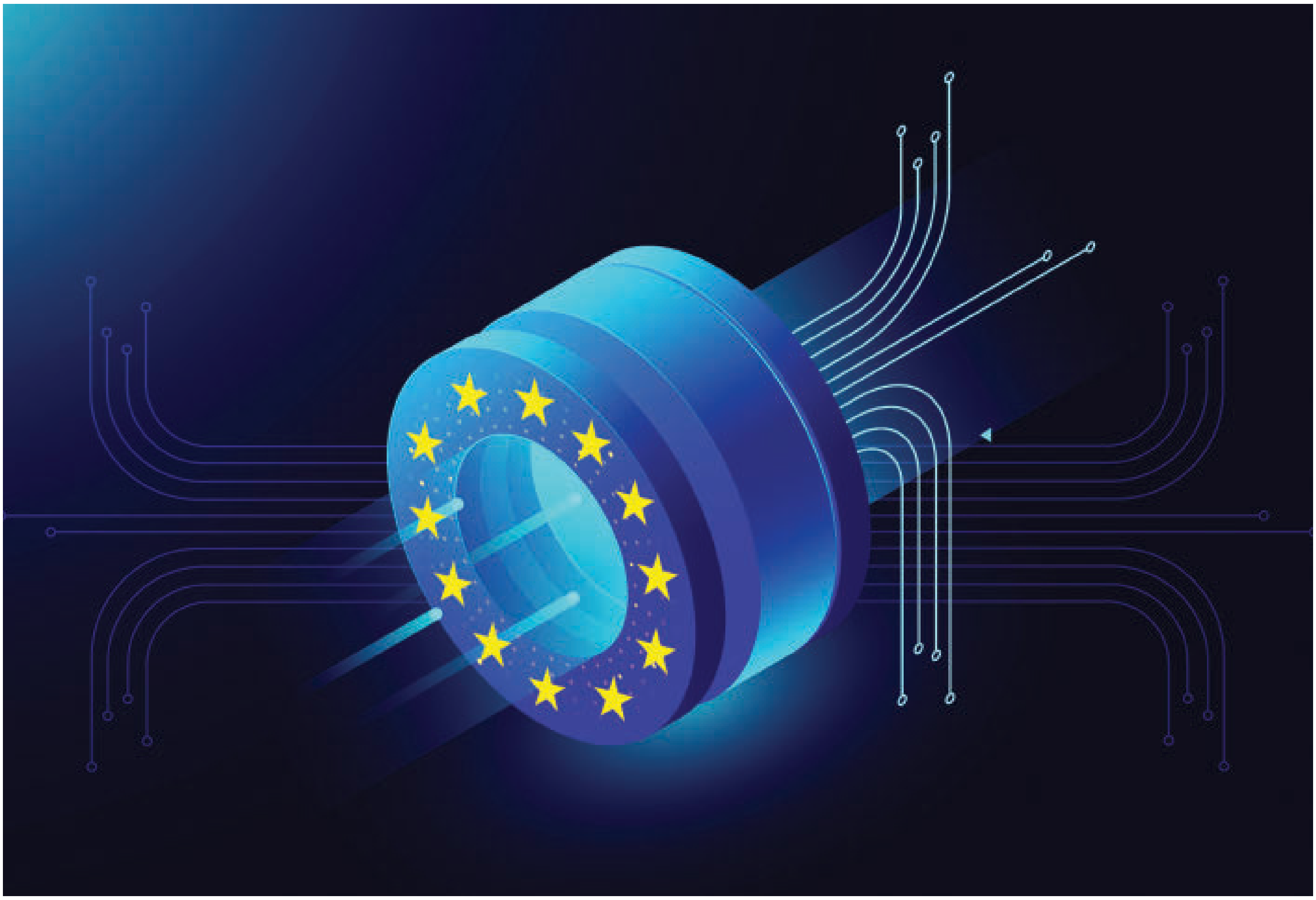


IMAGE BY FULLVECTOR

who sell on its marketplace to benefit Amazon's own retail business, directly competing with those third-party sellers.[a] Apple is currently under scrutiny for setting disproportionally high fees to businesses that seek to participate in its App Store.[b] Abuse of market power can also take place at the demand side. For example, in a controversial case, Germany's General Cartel Office concluded Facebook abused its market dominance by harvesting information from its users and combining user personal data from Facebook and WhatsApp without user consent.[c]

The main regulatory body in the EU is the European Commission, the executive branch of the Union. It is responsible for proposing legislation and implementing decisions and EU treaties. It is in charge of the EU market competition enforcement and it closely coordinates with national competition authorities at the EU member states that have a more country-specific focus on their enforcement duties.

The European Commission announced the Digital Markets Act (DMA) in late 2020 to regulate digital platforms like GAFAM. This new approach relies on before-the-fact investigation of business practices to assess wrongdoing. DMA imposes general obligations on what platforms can and cannot do when interacting with users. It prohibits self-preferencing, imposes serious limitations on bundling and tying practices, constrains platforms from using information provided by competitors to their own benefit, restricts excessive fees for market participants, and limits gatekeepers from combining of data across platforms without an opt-out option. In addition, the DMA also specifies the obligation of platforms to allow business users and consumers to transport platform-stored data to competitor firms in a continuous and real-time manner, following the principles of the General Data Protection Regulation (GDPR). It aims in this way to reduce the great data advantage of big platforms and the resulting information asymmetries.

The DMA also defines a framework for authorities to perform their enforcement tasks more effectively by accessing data and information located at platforms' infrastructure, and examining platforms' algorithmic systems. This is an important innovation in antitrust enforcement, as investigations in digital markets, are often in "unknown territory" without access to vital objective information to properly assess the impact of the business practice under question.

The DMA ultimately makes digital markets more contestable, reducing entry costs and increasing competition and consumer welfare. Incumbent platforms have more incentives to innovate and improve the quality of their services as they feel more threatened by new entrants and smaller competitors that find sufficient market space to develop their own businesses. They contribute to the fair allocation of the created value, and reduce big platform tendencies

[a] Press release by the European Commission (Nov. 10, 2020); https://bit.ly/3zaT0Gd

[b] Press release by the European Commission (Apr. 30, 2021); https://bit.ly/3g3wc3E

[c] Press release by the German General Cartel Office (Feb. 7, 2019); https://bit.ly/2RxSGjU

to extract disproportional rents.

But, there is room for improvement in achieving these goals. First, consumer choice and multihoming among online providers require frameworks that enable data portability and interoperability in practice. For this to happen, complementary rules must provide clarity and standards over effective data and information sharing. A mechanism is needed that allows free flow of data across disparate digital ecosystems, in keeping with data protection rules.[4] The DMA must be accompanied by such a mechanism to reduce the advantages of gatekeepers.

Second, DMA's obligations could consider specifics of the gatekeepers' business models. Business practices might incorporate efficiency gains for business users and consumers depending on the model in place.[1] For example, self-preferencing is welcome if it helps users quickly reach what they are looking for or encourages better service. In general, practices associated with benefits from economies of scale and scope might outweigh potential abuse in specific cases and therefore should be allowed.

The EU's new regulatory proposal thoroughly addresses competition concerns in digital markets. It arguably leads the way for global platform regulation. Improvements in its current form and development of complementary instruments can make it the state of the art in platform regulation and an example for other jurisdictions to follow.

References

1. Cabral, L. et al. The EU Digital Markets Act. Publications Office of the European Union, Luxembourg, 2021, doi:10.2760/139337, JRC122910.
2. European Commission decision of 27 June 2017 relating to a proceeding under Article 102 of the Treaty on the Functioning of the European Union and Article 54 of the EEA Agreement. Case AT.39740—Google Search (Shopping); https://bit.ly/3g0yJf0
3. European Commission Decision of 6 March 2013 relating to a proceeding on the imposition of a fine pursuant to Article 23(2)(c) of Council Regulation (EC) No 1/2003. Case COMP/39.530—Microsoft (Tying); https://bit.ly/2Sj5jQm
4. Parker, G., Petropoulos, G., and Van Alstyne, M.W. Digital Platforms and Antitrust (May 22, 2020); https://bit.ly/3v2mCCv
5. Report of the Digital Competition Expert Panel. Unlocking Digital Competition. UK HM Treasury (2019); https://bit.ly/2TQLGQ5
6. Stigler Committee on Digital Platforms. Final Report (Sept. 2019); https://bit.ly/3iysdOa

Georgios Petropoulos (gpetrop@mit.edu) is Marie Skłodowska-Curie fellow at the Massachussets Institute of Technology and Bruegel Digital fellow at the Digital Economy Lab of Stanford University, Stanford, CA, USA.

DOI:10.1145/3469115

Brett A. Becker

▸ Mark Guzdial, Column Editor

Education

What Does Saying That 'Programming Is Hard' Really Say, and About Whom?

Shifting the focus from the perceived difficulty of learning programming to making programming more universally accessible.

The commonly held belief that programming is inherently hard lacks sufficient evidence. Stating this belief can send influential messages that can have serious unintended consequences including inequitable practices.[4] Further, this position is most often based on incomplete knowledge of the world's learners. More studies need to include a greater diversity of all kinds including but not limited to ability, ethnicity, geographic region, gender identity, native language, race, and socioeconomic background.

Language is a powerful tool. Stating that programming is hard should raise several questions but rarely does. Why does it seem routinely acceptable—arguably fashionable—to make such a general and definitive statement? Why are these statements often not accompanied by supporting evidence? What is the empirical evidence that programming, broadly speaking, is inherently hard, or harder than possible analogs such as calculus in mathematics? Even if that evidence exists, what does it mean in practice? In what contexts does it hold? To whom does it, and does it not, apply?

Computer science has a reputation[3] and this conversation is part of that. Is programming inherently hard? Although worthy of discussion, this Viewpoint is not concerned with explicitly answering this question. It is concerned with *statements* such as "programming is hard" and particularly the direct and indirect *messages* such statements can convey. It explores who says this, why and how it is said, and what the ramifications are. Consider the statement "computer programming could be made easier."[7] Although this implies that programming is possibly more difficult than it needs to be, it clearly sends a different message than "programming is hard." This exemplifies how two fairly similar statements can convey very different messages and likely have different effects.

IMAGE BY REDPIXEL.PL

Who Says Programming Is Hard?

The belief that programming is hard seems to be widespread among teachers and researchers.[4] Academic papers frequently state that programming is hard anecdotally, as if just stating the obvious. Yet it is rarely discussed outside of motivating and justifying research. Although this approach is rarely challenged, when it is, the stakes are high. One critique points out this stance can lead to uncritical teaching practices, poor student outcomes, and may impact negatively on diversity and equity.[4] That work suggests the expectations of educators are unrealistic—not that programming is too hard.

The message of difficulty is also carried through more everyday mechanisms. It can be unknowingly or inadvertently perpetuated through our teaching habits, textbook language, terminology, the defensive climates in our classrooms,[3] tools, and programming languages themselves. A case in point is programming error messages that, across almost all languages, are notorious for causing confusion, frustration, and intimidation, and have been described as mysterious and inscrutable.[1]

The belief that programming is hard is not confined to academia. The concept of the "10x developer"—the elusive developer that is 10 times more productive than others—serves to communicate that programming is hard and few can be good at it. Even professionals have referred to programming as a black art,[8] a view that persists to present day for some. Hollywood typecasts embodying the hacker stereotype, staring at screens while 1s and 0s quickly stream by, present programming as a mystical, supernatural ability. It is possible that such portrayals have negative side effects in addition to their entertainment value.

Known Difficulties and Overlooked Successes

It is more accurate to say that certain aspects of programming are difficult or more challenging than others. This considerably dilutes the notion that programming is innately hard, as some aspects of many endeavors are more difficult than others. More pointed statements are also less likely to inflict collateral damage on general audiences and are less prone to misuse. Aspects of programming that are accepted to be challenging include knowledge transfer issues—including negative transfer—and developing a notional machine, among others.[3] Programming has several candidate threshold concepts[5] but so do aspects of many disciplines.

The belief that programming is hard seems to be widespread among teachers and researchers.

Programming is also the subject of many misconceptions.[5] This might be because it is hard. On the other hand, saying it is hard might be a convenient way of explaining these misconceptions. Other (and generally older) disciplines also have challenges with misconceptions. Many physics students struggle with the fundamentals of mechanics such as force vs. acceleration, speed vs. velocity, weight vs. mass, and the concept of inertia. Such comparisons are not that indirect. These well-established concepts underpin a more formulaic approach in much practice, similar to how the conceptual underpinnings of programming are expressed ultimately in code. Engineering also can suffer from a hard image. The notion that mathematics is hard already exists in many school systems and is often echoed by parents and other stakeholders, leading to negative implications including working against broadening participation. However, computing casts a very wide net in modern society; it is intertwined with much of daily life and influences chances of success in many ways. Similar could be said for mathematics but computing may seem more visible and tangible to many. It is also a very attractive and in demand career path for tomorrow's graduates.

There are examples that support programming not being hard. Success has been found in pair programming, peer instruction, worked examples, games, and contextualized approaches such as media computation.[3] These are often backed by empirical evidence[5] but are frequently overlooked when convenient. Block-based programming has become extremely successful, particularly with younger children. New modalities also demonstrate that computing is relatively young and rapidly changing. This pace of change itself may be one of the leading factors contributing to the perceived difficulty of programming.

Hard For Whom?

Given how many people are affected by computing technologies, another problem with statements that programming is hard is that most research and media reports are based on very narrow samples of Earth's population—largely from American, Commonwealth, and European contexts. There are entire countries, nearly entire continents, and countless groups whose experiences have not been rigorously studied and contrasted with others. Even in more frequently researched locales, our current views are not representative of many. This is rarely acknowledged or acted upon. Many sub-populations in terms of all manners of diversity and identity including ability, culture and ethnicity, geography, gender, native language, race, socioeconomic background, and many others are still underrepresented, everywhere.

Additionally, most data and experience currently come from computing students, yet computing is quickly becoming a mainstream discipline embedded in school curricula and for an increasing range of academic disciplines in higher education.[5] Declaring programming to be hard for relatively well-resourced Western computing students paints a bleak view for others. Even if programming was found to be uniquely hard for these students, this finding might only hold for the very limited, biased samples that have contributed to our current knowledge. Carelessly attempting to generalize such a finding would serve to shut the door on those who have not yet had their experiences counted. In effect, such practice is already happening when the message that programming is hard is perpetuated with little or no context.

Ramifications

There are several examples where explanations for observations in computing education have unintended negative consequences. For instance, to explain struggling students sitting next to high achievers, the "Geek Gene" hypothesis proved convenient. This states that programming is an innate ability. In other words, one generally cannot learn to be a great programmer; one is or is not, and most are not. There is evidence that computer scientists believe that innate ability is more important in computing than in other disciplines and this is known to be a barrier to broadening participation.[3] Although the Geek Gene hypothesis has met resistance[3,8] damage has already been done and might continue.

One need not look far for other contemporary examples of unintended messages having undesired effects. It was recently shown that competitive enrollment policies for university-level computing majors have a negative impact on student sense of belonging, self-efficacy, and perception of department.[6] Of course, these were not intended outcomes. Competitive enrollments were largely a response to growing student numbers and demand that could not be met. Nonetheless, this mechanism sent an unintended signal to students, resulting in undesired negative consequences. It is likely the perpetuation of the message that programming is hard has similar effects.

A recent series of NSF workshops revealed one of the most-heard comments made by non-computing educators when discussing computing curricula was "stop making computing/programming look scary."[2] Is that really the image that we intend to portray or is it just a byproduct of computing culture? If educators think that programming is scary, how can we expect students to think any differently? These messages may already have resulted in countless students abandoning computing, or not considering it in the first place. We will never know. It is likely that untold damage has already occurred and continues to accumulate.

The Future

The relatively short history of programming is filled with constant change, and what is taught can change often.[3] Character-based high-level programming has, overall, led the battle for adoption particularly at university and in industry, but there are many examples of programming that do not fit this mold. From Logo and Hypertalk, to prototype spoken-language programming languages, to block-based programming, and domain- and task-specific programming, what exactly constitutes programming can depend on who is asked and when they are asked. Today, low-code and no-code are emerging programming modalities, another sign that what constitutes programming is in constant flux. The number of programmers in non-computing contexts is also rapidly increasing.[1] Surely even if programming was deemed hard yesterday, that does not mean it will be tomorrow.

How can we change programming's notorious reputation? The answer is likely multifaceted and includes being aware of the true effects of the beliefs we hold and the messages we send. In addition, these should be based on evidence, informed by an appropriate diversity of people. We should have realistic expectations of students and focus on what we know is successful both in practice and in our messaging, including examining the intent of statements on the matter.

Conclusion

The notion that "programming is hard" is frequently reinforced in our classrooms, workplaces, academic literature, and the media. However, this position frequently reflects ideological views and lacks sufficient evidence. Statements that programming is hard can have obvious direct consequences. However, they can also convey more indirect messages—in effect sending signals that can have unintended consequences on students, educators, the community, and the discipline of computing itself. These are rarely considered.

Is programming hard or not? Current evidence is not compelling nor diverse enough to answer this question in general. More defensible (and likely honest) answers are: "it depends," and "both." Why then, is it so common to say that it is hard? Is it often said anecdotally because there is not that much evidence to support it? Because the evidence that does exist is difficult to understand? Could it be that it is just too convenient for motivating and justifying work? Is it that many want programming to seem hard, consciously, or unconsciously? Do tech companies and hiring managers depend on the image of programming being tough and elite? Is it too convenient for explaining phenomena whose true explanations remain elusive? Is it just an easy excuse for failure? Perpetuating this belief only serves to reinforce a shaky base of evidence that undermines any more rigorous evidence-based research. If we are going to make claims on the difficulty of programming, the community has a duty to provide robust empirical evidence from diverse contexts and state the findings responsibly.

Many current events and sociopolitical realities have caused us to question our educational practices recently. Considering the present global context, blanket messages that "programming is hard" seem outdated, unproductive, and likely unhelpful at best. At worst they could be truly harmful. We need to stop blaming programming for being hard and focus on making programming more accessible and enjoyable, for everyone. C

References

1. Becker, B.A. et al. Compiler error messages considered unhelpful: The landscape of text-based programming error message research. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education* (Aberdeen, Scotland UK) (ITiCSE-WGR '19). ACM, New York, NY, 2019, 177–210; https://bit.ly/2T97WUT
2. Birnbaum, L., Hambrusch, S. and Lewis, C. *Report on the CUE.NEXT Workshops.* Technical Report (2020); https://bit.ly/3x8ev8Q
3. Guzdial, M. Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics 8*, 6 (2015), 1–165.
4. Luxton-Reilly, A. Learning to program is easy. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (Arequipa, Peru) (ITiCSE '16). ACM, New York, NY, 2016, 284–289; https://bit.ly/3ivrKfM
5. Luxton-Reilly, A. et al. Introductory programming: A systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) (ITiCSE 2018 Companion). ACM, New York, NY, 2018, 55–106; https://bit.ly/3v1D9qh
6. Nguyen, A. and Lewis, C.M. Competitive enrollment policies in computing departments negatively predict first-year students' sense of belonging, self-efficacy, and perception of department. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) (SIGCSE '20). ACM, New York, NY, 2020, 685–691; https://bit.ly/2TTr3Tl
7. Sime, M.E., Arblaster, A.T., and Green, T.R.G. Structuring the programmer's task. *Journal of Occupational Psychology 50*, 3 (1977), 205–216; https://bit.ly/3w4NNxL
8. Tedre, M. From a black art to a school subject: Computing education's search for status. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (Trondheim, Norway) (ITiCSE '20). ACM, New York, NY, 2020, 3–4; https://bit.ly/3v436po

Brett A. Becker (brett.becker@ucd.ie) is an assistant professor in the School of Computer Science, University College Dublin, Dublin, Ireland.

DOI:10.1145/3469118

George V. Neville-Neil

Article development led by acmqueue
queue.acm.org

Kode Vicious
In Praise of the Disassembler

There is much to be learned from the lower-level details of hardware.

Dear KV,

I have read enough of your columns to see that one of your frequent themes is that source code is meant to be read by people, including one's future selves, and that how that code is processed by an interpreter such as Python or a compiler is less important than making the code clear to the next reader. You seem to be saying that our tools will work out what we mean and that we should treat the interpreter or compiler as a black box that magically turns our source into running code. I feel you are ignoring an important part of understanding software, which is what happens when your compiled code executes on a machine—after all, no computer executes C, C++, or Rust directly; they are running a compiled binary. What happens when you have a bug that appears in the binary only because of a mistake by the compiler, linker, assembler, or other part of the tool chain, which must occur from time to time. What then?

Dissembling at the Back End

Dear Dissembling,

Indeed, there have been many people and many movements within the software industry over the past 50 years that have shifted developers and development further away from machine code and assembly language—and not without good reasons. The abstractions of higher-level languages over the admittedly brief history of computing have allowed the explosion of software and services that nonprogrammers take for granted every day. Those of us who often work down in the bowels of technology know that these abstractions, and the movement of developers away from the machine, come with costs.

There are many problems in software systems that cannot be properly understood without a good understanding of the lower-level—some might argue the lowest-level—details of the machines we work on, and it shocks and angers me when I try to explain such things and get blank stares from people who have been in the industry for many years. The attitude can be summed up in a line I heard back in high school when I was learning my first assembly language on what was even then an ancient machine, the DEC-10. "You'll never need to use this because in the future all languages will be high-level like Fortran and Cobol" was the refrain of the math teachers who proctored our computer lab.

What is interesting about this quote is that it is wrong in two different ways. The first is that the languages they chose, though they are still in use today, are by far not the majority languages in software development, meaning that they were not the be-all and end-all that these teachers thought they were. The second fallacy is the idea that what I learned in my first brush with assembly and machine code would be useless in my career, when nothing has been further from the truth.

While it is true that the last piece of assembler I wrote by hand (and which wound up in a commercial product) was written 30 years ago, the lessons I learned by interacting directly with the hardware—without even the cushion of C (a.k.a. assembly with for loops)—remain with me to this day and have allowed me to track down difficult bugs, as well as significant performance problems. I have written in the past about understanding algorithms and how these expressions of problem solving relate to good performance[a] but the other side of this coin is understanding how the machine on which your algorithms run actually works. What does it take to understand the nether regions of computer systems? KV's advice comes in three parts: Start small. Start small. Start small.

Start at the small end of processor and computer hardware. A modern laptop, desktop, or server system is a fantastically complex piece of equipment that for the most part grew by the accretion of features that will utterly distract anyone who is new to this area of computing. You want to start with a small system, with few features, and with a small instruction set, so that you can, hopefully, jam nearly all the details into your head at once. Symmetric

[a] See https://bit.ly/3io2PuF

multiprocessing, multilevel caches, speculative execution of instructions, long pipelines, and all the rest of the innovations in hardware that have been added to improve performance as we have hit the wall at the end of Moore's Law are important to learn—later. Very few people start learning piano by sitting down to play Mozart or Fats Waller, and so you should not attempt to scale the heights of a superscaler processor on day one.

A good place to start in 2021 is with a small, cheap, embedded processor, and by this I definitely do not mean the Raspberry Pi or any of that ilk. The current Pi is based on a complex ARMv8 design. Do not start there. A better place to start is with the popular Atmel AVR chips, which are available on Arduino and other boards used by hobbyists and embedded-systems designers. These processors are eight-bit—yes, you read that correctly, eight-bit—systems much like the early microcomputers of the 1980s, and they have small memories, slow processing speeds, a small number of registers, and most importantly, a small and easy-to-remember set of assembly operations (opcodes).

These constraints actually help you learn the machine without a lot of extraneous distractions. Another advantage of this architecture is that instructions take either one or two clock cycles, depending on whether they are internal or I/O operations. Having instructions with a small, known cycle time makes it easier to think about the performance of the code you're looking at. Getting experience with performance in this way is key to being able to understand the performance of larger and more complex architectures. KV cannot emphasize enough that you want to start with an assembly language that is small and easy to understand. Go look at any data book for a big Intel, ARM, or other processor and you will see what I mean. The AVR instruction set fits on a single page.

Read small programs. I have given this advice to developers of languages at all levels, from high to low, but when you are trying to learn the lowest levels of the machine, it is even more important. The canonical "Hello, World" program taught to all C programmers results in a binary file with millions of instructions when it is statically linked. A description of what happens when you execute it is the subject of an excellent talk by Brooks Davis at the 2016 Technical BSDCan Conference.[b]

What does it take to understand the nether regions of computer systems?

The examples in most beginning programming tutorials for the Arduino—those that blink an LED—are the size of an example I would suggest. Later, you can develop a code editor that morphs into a mail reader and Web browser, which seems to be the course of all software projects over time, but for now just turn the light on and off. There are two ways to examine these small programs. The first is to look at the output of various compilers that build C or higher-level code for these embedded devices, such as the Arduino IDE, or LLVM- and GNU-based cross-compilers. You can look at the code by dumping it with an objdump program, or you can look at it in a debugger if you have one that understands the AVR instruction set. The debugger is the more interesting environment because you can both read the code and also execute it, stop it, inspect registers and memory, and the like. Both LLDB from LLVM and GDB from GNU have assembler modes, so you can even switch between a higher-level language and assembler.

Write small pieces of low-level code. That piece of assembly I wrote 30 years ago was only a couple of pages long, in part because what it did was simple (pulling audio data from a parallel port on a microcomputer) and because it was written using a powerful complex instruction set computer (CISC) assembly language (Motorola 68K). A CISC assembly language is closer to C than machine code and often has opcodes that do quite a bit on your behalf. The AVR can be considered in the family of reduced instruction set computer (RISC) processors, where each instruction is very simple, and complex operations must be built out of these. Much of the raw assembly that is still written today is along this model of a few pages of instructions.

When you are starting out you want to be able to hold the entire program in your head if at all possible. Once you are conversant with your first, simple assembly language and the machine architecture you are working with, it will be completely possible to look at a page or two of your assembly and know not only what it is supposed to do but also what the machine will do for you step by step. When you look at a high-level language, you should be able to understand what you mean it to do, but often you have no idea just how your intent will be translated into action—assembly and machine code is where the action is.

Developing these skills will take you far past blinking an LED on a hobby board. Being able to apply these same skills to larger machines, with more complex architectures, makes it possible to find all kinds of Heisenbugs,[c] optimize systems for power and performance, as well as understand the ramifications of low-level security attacks such as return-oriented programming (ROP) and buffer overflows. Without these skills, you are relegated to the cloudy upper layers of software, which is fine, until the tools or your hardware or the gods of software fail you.

KV

[b] See https://bit.ly/3gl3CK1

[c] See https://bit.ly/2TMk1j3

Related articles on queue.acm.org

Programming in Franglais
Rodney Bates
https://bit.ly/3cL5QBN

GNL Is Not Linux
Kode Vicious
https://bit.ly/3g47lNn

Coding for the Code
Friedrich Steimann and Thomas Kühne
https://bit.ly/3gh2rvj

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

DOI:10.1145/3445973

Ben Shneiderman

Viewpoint
Responsible AI: Bridging From Ethics to Practice

Recommendations for increasing the benefits of artificial intelligence technologies.

IMAGE BY ELENABSL/SHUTTERSTOCK

THE HIGH EXPECTATIONS of AI have triggered worldwide interest and concern, generating 400+ policy documents on responsible AI. Intense discussions over the ethical issues lay a helpful foundation, preparing researchers, managers, policy makers, and educators for constructive discussions that will lead to clear recommendations for building the reliable, safe, and trustworthy systems[6] that will be commercial success. This Viewpoint focuses on four themes that lead to 15 recommendations for moving forward. The four themes combine AI thinking with human-centered User Experience Design (UXD).

Ethics and Design. Ethical discussions are a vital foundation, but raising the edifice of responsible AI requires design decisions to guide software engineering teams, business managers, industry leaders, and government policymakers. Ethical concerns are catalogued in the Berkman Klein Center report[3] that offers ethical principles in eight categories: privacy, accountability, safety and security, transparency and explainability, fairness and non-discrimination, human control of technology, professional responsibility, and promotion of human values. These important ethical foundations can be strengthened with actionable design guidelines.

Autonomous Algorithms and Human Control. The recent CRA report[2] on "Assured Autonomy" and the IEEE's influential report[4] on "Ethically Aligned Design" are strongly devoted to "Autonomous and Intelligent Systems." The reports emphasize machine autonomy, which becomes safer when human control can be exercised to prevent damage. I share the desire for autonomy by way of elegant and efficient algorithms, while adding well-designed control panels for users and supervisors to ensure safer outcomes. Autonomous aerial drones become more effective as remotely piloted aircraft and NASA's Mars Rovers can make autonomous movements, but there is a whole control room of operators managing the larger picture of what is happening.

Humans in the Group; Computers in the Loop. While people are instinctively social, they benefit from well-designed computers. Some designers favor developing computers as collaborators, teammates, and partners, when adding control panels and status displays would make them comprehensible appliances. Machine and deep learning strategies will be more widely used if they are integrated in visual user interfaces, as they are in counterterrorism centers, financial trading rooms, and transportation or utility control centers.

Explainable AI (XAI) and Comprehensible AI (CAI). Many researchers from AI and HCI have turned to the problem of providing explanations of AI decisions, as required by the European General Data Protection Regulation (GDPR) stipulating a "right to explanation."[13] Explanations of why mortgage applications or parole requests are rejected can include local or global descriptions, but a useful complementary approach is to prevent confusion and surprise by making comprehensible user interfaces that enable rapid interactive exploration of decision spaces.

Combining AI with UXD will enable rapid progress to the goals of reliable, safe, and trustworthy systems. Software engineers, designers, developers, and their managers are practitioners who need more than ethical discussion. They want clear guidance about what to do today as they work toward deadlines with their limited team resources. They operate in competitive markets that reward speed, clarity, and performance.

This Viewpoint is a brief introduction to the 15 recommendations in a recent article in the *ACM Transactions on Interactive Intelligent Systems*,[8] which bridge the gap between widely discussed ethical principles and practical steps for effective governance that will lead to reliable, safe, and trustworthy AI systems. That article offers detailed descriptions and numerous references. The recommendations, grouped into three levels of governance structures, are meant to provoke discussions that could lead to validated, refined, and widely implemented practices (see the figure here).

TEAM: Reliable Systems Based on Sound Software Engineering Practices

These practices are intended for software engineering teams of designers, developers, and managers.

1) Audit trails and analysis tools: Software engineers could emulate the safety of civil aviation by making a "flight data recorder for every robot" to record actions, so that retrospective analyses of failures and near misses could track what happened and how dangers were avoided. Then analysts could make recommendations for improving design and training. Audit trails require upfront effort, but by reducing failures and improving performance they reduce injuries, damage, and costs.

Combining AI with UXD will enable rapid progress to the goals of reliable, safe, and trustworthy systems.

2) Software engineering workflows: Proposals for distinctive workflows for machine learning projects require expanded efforts with user requirements gathering, data collection, cleaning, and labeling, with use of visualization and data analytics to understand abnormal distributions, errors and missing data, clusters, gaps, and anomalies.

3) Verification and validation testing: The unpredictable performance of machine learning algorithms, means that algorithms need careful testing with numerous benchmark tasks. Since machine learning is highly dependent on the training data, different datasets need to be collected for each context to increase accuracy and reduce biases.

4) Bias testing to enhance fairness: Beyond algorithm correctness and data quality, careful testing will enhance fairness by lessening gender, ethnic, racial, and other biases.[1] Toolkits for fairness testing from researchers and commercial providers can seed the process, but involvement from stakeholders will do much to increase fairness and build connections that could be helpful when problems emerge.

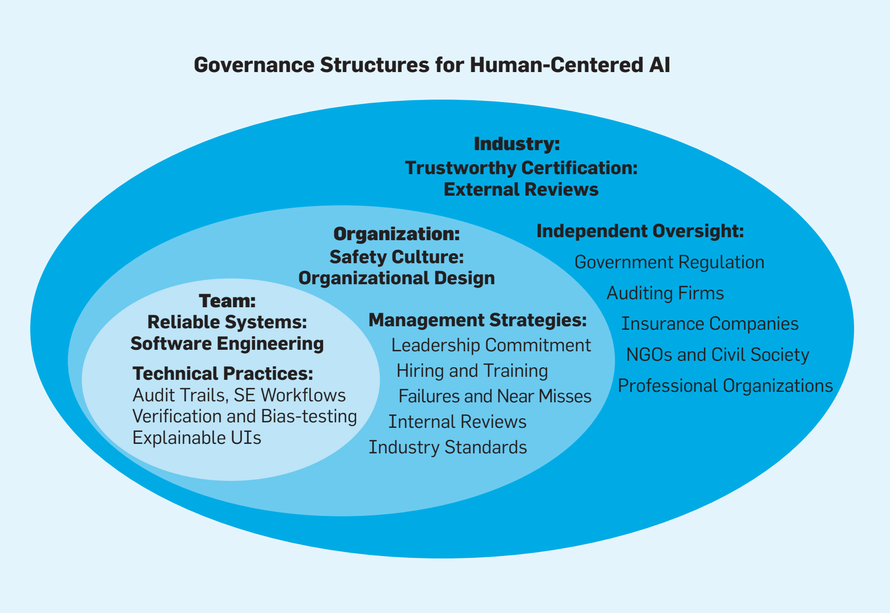
5) Explainable user interfaces: Software engineers recognize that explainable user interfaces enable more reliable development processes since algorithmic errors and anomalous data can be found more easily when explainability is supported. Exploratory user interfaces, often based on visualization are proving to be increasingly valuable in preventing confusion and understanding errors. Weld and Bansal[14] recommend that designers should "make the explanation system interactive so users can drill down until they are satisfied with their understanding."

Critics of these practices believe innovation is happening so quickly that these are luxuries that most software engineering teams cannot afford. Changing from the current practice of releasing partially tested software will yield more reliable and safer products and services.

ORGANIZATION: Safety Culture through Business Management Strategies

Management investment in an organizational safety culture requires budget and personnel, but the payoff is in reduced injuries, damage, and costs.

Governance structures to guide teams, organizations, and industry leaders.



6) Leadership commitment to safety: Leadership commitment is made visible to employees by frequent restatements of that commitment, positive efforts in hiring, repeated training, and dealing openly with failures and near misses. Reviews of incidents, such as monthly hospital review board meetings can bring much increased patient safety.

7) Hiring and training oriented to safety: When safety is included in job hiring position statements, that commitment becomes visible to current employees and potential new hires. Safety cultures may need experienced safety professionals from health, human resources, organizational design, ethnography, and forensics. Training exercises take time but the payoff comes when failures can be avoided and recovery made rapid.

8) Extensive reporting of failures and near misses: Safety-oriented organizations regularly report on their failures (sometimes referred to as adverse events) and near misses (sometimes referred to as "close calls"). Near misses can be small mistakes that are handled easily or dangerous practices that can be avoided, thereby limiting serious failures. NASA's Aviation Safety Reporting System (https://go.nasa.gov/2TdJhi1) and the Food and Drug Administration's Adverse Event Reporting System (https://bit.ly/3zl1A5B) provide models for public reporting of problems users encounter, while Bugzilla (https://bit.ly/3cRlkUT) is a useful model for technical bug reporting.

9) Internal review boards for problems and future plans: Commitment to a safety culture is shown by regularly scheduled monthly meetings to discuss failures and near misses, as well as to celebrate resilient efforts in the face of serious challenges. Review boards, such as hospital-based ones, may include managers, staff, and others, who offer diverse perspectives on how to promote continuous improvement. Google, Microsoft, Facebook, and other leading corporations have established internal review processes for AI systems.

Changing from the current practice of releasing partially tested software will yield more reliable and safer products and services.

10) Alignment with industry standard practices: Participation in industry standards groups such as those run by the IEEE, International Standards Organization, or the Robotics Industries Association show organizational commitment to developing good practices. AI-oriented extensions of the software engineering Capability Maturity Model[11] are being developed to enable organizations to develop carefully managed processes and appropriate metrics to improve software quality.

Skeptics worry that organizations are so driven by short-term schedules, budgets, and competitive pressures that the commitment to these safety culture practices will be modest and fleeting.[5] Organizations can respond by issuing annual safety reports with standard measures and independent oversight. It may take years for organizations to mature enough so they make serious commitments to safety.

INDUSTRY: Trustworthy Certification by Independent Oversight

The third governance layer brings industry-specific independent oversight to achieve trustworthy systems that receive wide public acceptance. The key to independent oversight is to support the legal, moral, and ethical principles of human or organizational responsibility and liability for their products and services. Responsibility is a complex topic, with nuanced variations such as legal liability, professional accountability, moral responsibility, and ethical bias. Independent oversight is widely used by businesses, government agencies, universities, non-governmental organizations, and civic society to stimulate discussions, review plans, monitor ongoing processes, and analyze failures. The goal of independent oversight is to promote continuous improvements that ensure reliable, safe, and trustworthy products and services.

11) Government interventions and regulation: Many current AI industry leaders and government policy makers fear that government regulation would limit innovation, but when done carefully, regulation can accelerate innovation as it did with automobile safety and fuel efficiency. A U.S. government memorandum for Executive Branch Departments and Agencies offered ten principles for "stewardship of AI applications,"[10,12] but then backed away by suggesting that: "The private sector and other stakeholders may develop voluntary consensus standards that concern AI applications, which provide nonregulatory approaches to manage risks associated with AI applications that are potentially more adaptable to the demands of a rapidly evolving technology."

12) Accounting firms conduct external audits for AI systems: Independent financial audit firms, which analyze corporate financial statements to certify they are accurate, truthful, and complete, could develop reviewing strategies for corporate AI projects. They would also make recommendations to their client companies about what improvements to make. These firms often develop close relationships with internal auditing committees, so that there is a good chance that recommendations will be implemented.

13) Insurance companies compensate for AI failures: The insurance industry is a potential guarantor of trustworthiness, as it is in the building, manufacturing, and medical domains. Insurance companies could specify requirements for insurability of AI systems in manufacturing, medical, transportation, industrial, and other domains. They have long played a key role in ensuring building safety by requiring adherence to building codes for structural strength, fire safety, flood protection, and many other features. Building codes could be a model for software engineers, as described in Landwehr's proposal for "a building code for building code."[6] He extends historical analogies to plumbing, fire, or electrical standards by reviewing software engineering for avionics, medical devices, and cybersecurity, but the extension to AI systems seems natural.

These recommendations are meant to increase reliability, safety, and trustworthiness while increasing the benefits of AI technologies.

14) Non-governmental and civil society organizations: NGOs have proven to be early leaders in developing new ideas about AI principles and ethics, but now they will need to increase their attention to developing ideas about implementing software engineering practices and business management strategies. An inspiring example is how the Algorithmic Justice League was able to get large technology companies to improve their facial recognition products so as to reduce gender and racial bias within a two-year period. This group's pressure was influential in the Spring 2020 decisions of leading companies to halt their sales to police agencies in the wake of the intense movement to limit police racial bias.

15) Professional organizations and research institutes: Established and new organizations have been vigorously engaged in international discussions on ethical and practical design principles for responsible AI. However, skeptics caution that industry experts and leaders often dominate professional organizations, so they may push for less restrictive guidelines and standards.[9] Ensuring diverse participation in professional organizations and open reporting, such as the Partnership on AI's Incident Database can promote meaningful design improvements. Academic research centers have been influential but their resources are often dwarfed by the budgets, systems, and datasets held by leading companies. Industry-supported research centers such as Open AI (https://openai.org) and the Partnership on AI (https://partnershiponai.org) could play a role in technology innovation and more effective governance.

These recommendations for teams, organizations, and industries are meant to increase reliability, safety, and trustworthiness while increasing the benefits of AI technologies. After all, the stakes are high: the right kinds of technology advance human values and dignity, while promoting self-efficacy, creativity, and responsibility. The wrong kinds of technology will increase the dangers from failures and malicious actors. Constructive adoption of these recommendations could do much to improve privacy, security, environmental protection, economic development, healthcare, social justice, and human rights. C

References

1. Baeza-Yates, R. Bias on the Web. *Commun. ACM 61*, 6 (June 2018), 54–61.
2. Computing Research Association. Assured Autonomy: Path toward living with autonomous systems we can trust. Washington, D.C. (Oct. 2020); https://bit.ly/3weGL9W
3. Fjeld, J. et al. Principled artificial intelligence: Mapping consensus in ethical and rights-based approaches to principles for AI. Berkman Klein Center Research Publication, (2020–1); https://bit.ly/358NYfN
4. IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems, First Edition. IEEE (2019); https://bit.ly/3gaBrig
5. Larouzee, J. and Le Coze, J.C. Good and bad reasons: The Swiss cheese model and its critics. *Safety Science, 126*, 104660 (2020); https://bit.ly/3izZ8Ca
6. Landwehr, C. We need a building code for building code. *Commun. ACM 58*, 2 (Feb. 2015), 24–26.
7. Shneiderman, B. Human-centered artificial intelligence: Reliable, safe and trustworthy. *International Journal of Human Computer Interaction 36*, 6 (2020), 495–504; https://bit.ly/3gaUetz
8. Shneiderman, B. Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy Human-Centered AI Systems, *ACM Transactions on Interactive Intelligent Systems 10*, 4, Article 26 (2020); https://bit.ly/3xisPff
9. Slayton, R. and Clark-Ginsberg, A. Beyond regulatory capture: Coproducing expertise for critical infrastructure protection. *Regulation & Governance 12*, 1 (2018), 115–130; https://bit.ly/3xbhjSK
10. U.S. White House. American Artificial Intelligence Initiative: Year One Annual Report. Office of Science and Technology Policy (2020); https://bit.ly/2Tl6sXT
11. von Wangenheim, C.G. et al. Creating software process capability/maturity models. *IEEE Software 27*, 4 (2010), 92–94.
12. Vought, R.T. Guidance for Regulation of Artificial Intelligence Applications. U.S. White House Announcement, Washington, D.C. (Feb. 11, 2019); https://bit.ly/35bhlxT
13. Wachter, S., Mittelstadt, B., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law and Technology 31* (2017), 841–887.
14. Weld, D.S. and Bansal, G. The challenge of crafting intelligible intelligence. *Commun. ACM 62*, 6 (June 2019), 70–79.

Ben Shneiderman (ben@cs.umd.edu) is an Emeritus Distinguished University Professor in the Department of Computer Science, Founding Director (1983–2000) of the Human-Computer Interaction Laboratory, and a Member of the U.S. National Academy of Engineering.

DOI:10.1145/3473614

Carlo Ghezzi

Viewpoint
Science Needs to Engage With Society: Some Lessons From COVID-19

Recent experiences toward communicating science to the general public.

THE SUDDEN DISRUPTION brought to our world by COVID-19 has put science and research at the center of the public attention. Scientists have been asked why the pandemic developed, how it may be countered and possibly defeated, and how its spread could be prevented. Almost all branches of science have been involved: from health to social sciences, from economics to technological sciences. Computer science (CS) was no exception. Its role—from tracing apps to modeling and simulation of virus spread—has been crucial. How did we manage this transition to the spotlight? What lessons can we learn? Hereafter, I will mainly focus on a key question: How effectively do we, as researchers, communicate with society? How should we equip ourselves to do it right? Why should we do it?

IMAGE BY INSPIRING/SHUTTERSTOCK

Scientific Dissemination

Researchers know how to communicate with peers. They learn how to do it since they enter a Ph.D. program, and continue to learn and improve throughout their career. Research is an intrinsically open process that relies on communication among peers. The main ambition of scientists is to achieve novel results and disseminate them to the world, contributing to the advance of human knowledge and welfare. Dissemination is mostly achieved through publications: research results have no value until they become public and can be challenged by further research and used to advance science or to exploit it. The term "publication" comes from Latin "publicare," meaning "make public." The pre-publication peer review process acts as a scientific filter. Publication exposes results to continuous validation through test of time.

Publication and engagement in debates with peers are key aspects of researchers' life. Career progress depends on how successful they are in producing and disseminating results within their community. However, scientists do not live in a closed world of peers. Their achievements can lead to various forms of public engagement. This is especially true for CS research, whose artifacts may lead to practical use and societal innovations. Researchers often cooperate with industry in joint programs or spin-off new industrial initiatives exploiting their results.

In my recent book, *Being a Researcher—An Informatics Perspective*,[1] I discuss research dissemination. I also discuss the importance of a generally more neglected and more sporadic form of public engagement through which scientists lead, or participate in, scientific debates with a broader audience, outside the circle of peers: with government, policy- and decision makers, and the general public. This participation exploded in COVID-19 time.

Public engagement became crucial since the scientific revolution in the 17th century, which proved that science is the main propeller of continuous changes and societal progress.[2] This caused continuous interaction between science and politics. In some cases, the political power even exerted control over scientific results. Today, politicians and governments often interact with scientists to ground their decisions on sound and factual data. Engagement in debates with the general public also became quite common, although still sporadic. At the end of World War II, several physicists initiated passionate discussions on the relation between research and war and became actively involved in pro-peace initiatives. Today scientists are often interviewed by media to comment on shocking examples of technological success or failures. In fewer cases, public engagement is a systematic effort to bring science to the broader public.

There are some good examples of more systematic efforts. The Jefferson Science Fellows program[a] in the U.S. has been launched to engage the American academic community with government. Scientists can spend a year in the U.S. Department of State or U.S. Agency for International Development advising on science and technology in relation with foreign policy and international development. The AAAS—American Association for the Advancement of Science—has launched the STPF Fellowship to support scientists who serve for one year engaging in public affairs in various branches of the U.S. federal government.[b]

There are also outstanding examples of top scientists who actively engaged in communicating science with the general public. The famous physicist (and Nobel prize recipient) Richard Feynman did documentaries and interviews on BBC[c] and wrote a book,[2] which provide an entertaining, yet rigorous, introduction to the essentials of physics. Computer scientist David Harel delivered widely acclaimed programs on Israeli radio and television; he published two broadly accessible yet informative books[4,5] on the fundamental notions and limitations of computability and algorithms. Another outstanding example is Michael Kearns's and Aaron Roth's recent book, *The Ethical Algorithm*,[6] which discusses the relation between algorithms and ethics—a crucial question in our society, where an unprecedented availability of data enables algorithms to automate human decisions and actions. Efforts of this kind are commendable, because understanding science makes citizens aware of its transformational role of society and empowers them to make more informed decisions.

Apart from some notable examples, the practice of public engagement is largely neglected, underrated, and even dismissed. Researchers do not learn how to debate science with the general public as part of their education. The existing reward system does not recognize it; promotion cases largely ignore it. Consequently, it is not surprising that when scientists do it, they mostly improvise, and the results are often poor and ineffective. Conversely, the ability to speak science to society is crucial, especially when research results have direct impact on society and human life, as in the time of COVID-19. It is especially and increasingly true for CS, due to its pervasive role in our society, through automation and autonomous decision making. Scientists cannot just focus on the development of technology but need to engage in discussions about its use and the ensuing potential disruptions.

Effective communication between researchers and the general public also demands a mature and competent audience. This raises additional serious concerns, since regrettably the level of scientific education has been decreasing in many countries. Most important, there is a lack of understanding of what science and the scientific method are all about, and hence the notion of validity of scientific results. Science does not produce absolute truths or perfect solutions; for most results, scientists can only say that all the evidence they have does not support the hypothesis that they are invalid. Likewise, most artifacts produced by technological research have limitations, they work under certain conditions, and may be further improved. Despite limitations and uncertainties, science is yet the best cognitive source we can rely upon at any given time. Regrettably, lack of scientific education generates a communication mismatch between researchers and the audience.

The convergence of two problems—poor dissemination of research by scientists and lack of basic scientific knowledge by the general audience—generates serious consequences. It is the main source of distrust of research and the cause of a phenomenon aptly called "death of expertise" by Tom Nichols.[7]

Science in the Time of COVID-19

Let us look back to what happened with the pandemic and what can be learned, to reflect on the role of science. More specifically, why and how we should communicate and interact with society at large and what should be done to make it possible.

COVID-19 brought science back on stage in the spotlight, emerging from the shaded "death of expertise" corner. Science and technology proved to be decisive weapons to fight against the virus. Scientific knowledge guided doctors and nurses to save human lives. The technology used to save patients is a product of research by doctors and engineers. The algorithms to model and predict virus spread and the monitoring apps that detect physical proximity with contagious individuals come from CS. Most important, the development of vaccines in a very short time frame has been a spectacular success of medical research.

COVID-19 has also been a showcase of the limits of science. At the beginning, scientists did not know the virus and could not provide adequate responses to defeat it. The initial lack of scientific knowledge has generated confusion in the general public and in the discussions with politicians and decision makers. Scientists played their part in generating further confusion. Inevitably, they were not prepared to provide a consolidated view of science on an unknown phenomenon that was threatening human lives worldwide. Ignorance of the virus led

[a] See https://bit.ly/3598wVj

[b] See https://bit.ly/3v54tEc

[c] See https://bbc.in/2Sgomef

some to initially underestimate its destructive power, while others seemed to be overly pessimistic. Sometimes, the two factions publicly engaged in disputes and gained visibility through the media. People got disoriented, and this was exacerbated by the lack of understanding of what science and the scientific method are. Scientists should have always distinguished between what is known, within the current frontiers of knowledge, and what is uncertain or poorly understood and needs further investigation.

The scale and complexity of COVID-19 also taught us another lesson. Often, real-life problems cut across several scientific areas: all of them can contribute to devise potential solutions, none of them are self-sufficient. Often, solutions do not depend on purely technical goals, but require setting parameters that depend on societal values. For example, distancing and lockdown may counter human relations and economy, technological solutions to control the spread of infections may counter privacy. Unfortunately, sometimes scientists seemed to only focus on their narrow area and defend (and overestimate) their own specific research interests instead of recognizing the need for multidisciplinary cooperation.

Some Lessons for the Future

Scientists should consider dissemination of research results to the general public and engagement in public debates as part of their duties. They should continue to be rewarded for their contributions to original, significant, and rigorous research, but they should also be rewarded for their efforts in contributing to the global understanding of the societal value of research.

This change, however, can only happen if the scientific community takes the lead and adopts some concrete actions. We should teach our students, in particular Ph.D. students, how to communicate science to the general public and engage with society. Successful achievements in public engagement should be counted in hiring and promotion cases. Academic organizations—like Informatics Europe and the CRA—might lead this process and provide recommendations for good practices that may be adopted by academic institutions. Professional and academic organizations—like ACM, IEEE, Informatics Europe, CRA—and institutions (for example, colleges or universities) might recognize outstanding examples through dedicated awards. Research funding programs should provide support for dissemination and outreach. More initiatives such as the aforementioned Jefferson Science or the STPF fellowships should be undertaken. These are only possible examples of important incremental steps, which may be rather naturally integrated in our existing processes and organization. Finally, we should continue to strive for better science education in schools at all levels, to form new generations of responsible citizens. Effective communication requires mature interlocutors at both sides.

We should learn from the pandemic to avoid falling back into the "death of expertise" pre-COVID-19 situation. We should mobilize energies to ask for better and broader science education for the young generations. We should learn how to speak science and broadly spread research results in an ethical way, avoiding both arrogance and oversimplification. We should avoid overselling our contributions. With respect to politicians, we should not be silent when decisions are made against scientific evidence. At the same time, we should humbly accept decisions that do not optimize our own specific viewpoint, when other societally relevant issues must also be considered. C

References

1. Ghezzi, C. Being a researcher: An informatics perspective. *Springer Nature* (2020).
2. Harari, Y.N. Sapiens: A brief history of humankind. *Signal* (2014).
3. Feynman, R.A. *Six Easy Pieces: Essentials of Physics Explained by Its Most Brilliant Teacher.* Basic Books, 22nd Edition, 2011.
4. Harel, D. *Computers Ltd: What They Really Can't Do.* Oxford University Press (2000).
5. Harel, D. Algorithmics: *The Spirit of Computing.* Addison-Wesley (1987).
6. Kerns, M. and Roth, A. *The Ethical Algorithm.* Oxford University Press (2020).
7. Nichols, T. *The Death of Expertise: The Campaign Against Established Knowledge and Why It Matters.* Oxford University Press (2017).

Carlo Ghezzi (carlo.ghezzi@polimi.it) is an emeritus professor at Politecnico di Milano, Italy.

I thank the anonymous reviewer for very constructive comments and suggestions, which helped me to make my point hopefully more clear and constructive.

DOI:10.1145/3448249

David Whalley, Xin Yuan, and Xiuwen Liu

Viewpoint
The Domestic Computer Science Graduate Students Are There, We Just Need to Recruit Them

Proven practices to recruit domestic computer science graduate students.

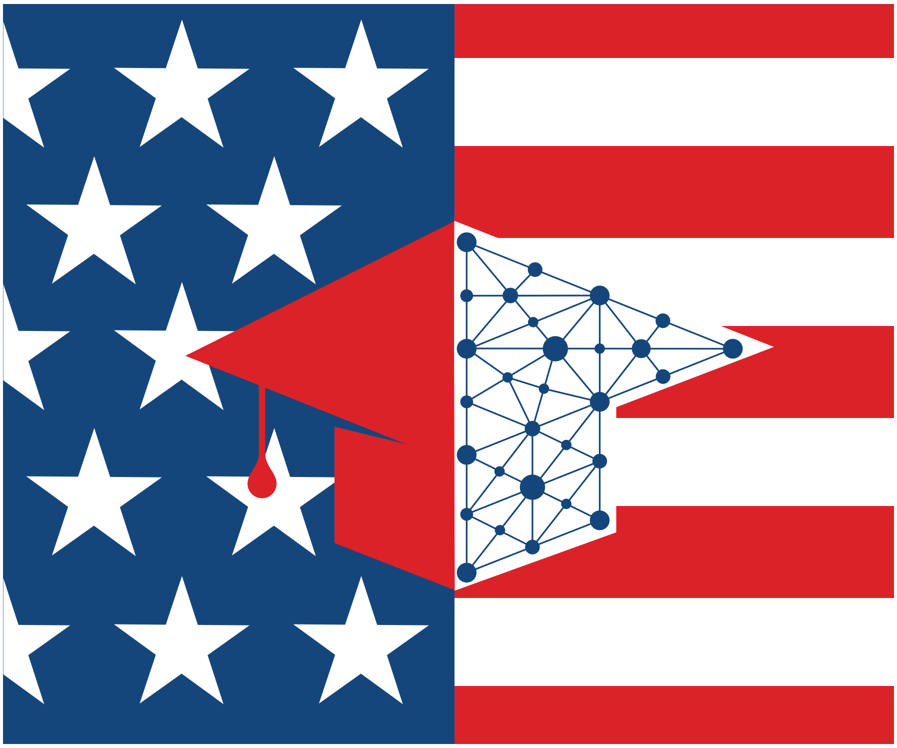


IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

IN THE "VARDI'S INSIGHTS" column, "Where Have All the Domestic Graduate Students Gone?", in September 2020 *Communications* (p. 5), Moshe Vardi stated: "Graduate programs admit so many international students not only because they have strong international applicants, but mainly because they do not have qualified domestic applicants." The authors of this Viewpoint agree with the points Vardi made in his column that the U.S. should welcome international computer science (CS) graduate students. However, we believe it is important to try to convince qualified domestic CS undergraduate (UG) students to attend graduate school as it can be beneficial to both the students and the nation to do so. In this Viewpoint, we share the approaches we have successfully utilized in the Florida State University (FSU) Department of Computer Science to increase the number of domestic CS graduate students.

In the past it had been the policy in the State of Florida that the State would provide a tuition waiver to all graduate students at Florida public state universities who were supported by a teaching or research assistantship. Between 1995 and 2005, the State of Florida made a sequence of changes to the way it would fund the state university system. One consequence of those shifts was that each university would need to take responsibility for providing tuition waivers. FSU decided it was important to not require any FSU graduate student on support to pay tuition even without the funding support from the State of Florida and decided to use its own funds to provide free tuition to these students. FSU was faced with a dilemma since many of its graduate students were international. Unlike domestic students, international students cannot qualify for in-state residency after 12 months and out-of-state tuition is much more expensive than in-state tuition. At FSU, this quickly led to a necessary balancing of tuition liability by optimizing the mix between domestic and international graduate assistants, an action that would require increasing the fraction of funded domestic graduate students.

The FSU Computer Science Department in the 1990s and early 2000s had a very large fraction of CS graduate students who were international students. Most of the domestic CS UG students in the U.S. have employment options after graduation that are not available to many CS UG students in developing countries. Most of FSU's CS international graduate students are from three countries: China, India, and Bangladesh. We have very few FSU CS international graduate students from Western Europe. From our discussions with our European colleagues, they face the same challenge of convincing their CS UG students to obtain a graduate degree.

We decided that we needed to educate our CS UG students about the merits of attending graduate school in order to recruit them into our graduate program. Our initial attempts included sending email to the CS UG students, having individual faculty talk to students about graduate school in senior classes, and holding advertised meetings where we provided pizza to encourage students to attend and learn about the merits of going to graduate school. These efforts had little success as they did not significantly increase the number of domestic applicants to our graduate program.

Conventional methods of distributing graduate school information to domestic CS undergraduate students are ineffective.

Effective Solutions

The next strategy we attempted was to have the FSU CS department chair individually meet with qualified CS UG students. The chair sends individual email messages to each FSU CS or computer engineering (CE) senior who meets a specified GPA requirement that is indicative of likely success in the FSU CS graduate program. The chair instructs each of these students to contact the department secretary to set up an appointment to meet with him/her and does not inform them of the reason for the meeting. Very few UG students will refuse a request to meet with the department chair. When the student arrives at the meeting, the chair and often one other faculty member (for example, the CS Director of Graduate/Undergraduate Studies) would together meet with each student. We compliment the student on his/her performance in their UG studies and ask the student about their plans after graduation. Most of the students will reply that they plan on getting a job. We ask them if they have considered going to graduate school. Most of the students reply that they had not considered this option. We then describe the merits of obtaining a CS graduate degree, including:

- CS graduates with an MS or Ph.D. can be hired by companies or government agencies that would normally not consider hiring BS CS graduates. One of us describes his own experience of the types of jobs that he was offered when completing his BS CS degree versus the types of jobs he was offered when completing his MS CS degree.
- MS CS graduates receive significantly higher pay on average than BS CS graduates and we provide the statistics on average salaries for FSU CS student graduates to show that getting an MS degree makes financial sense. If a student expresses any interest in getting a Ph.D., we point out that salaries at academic institutions tend to be lower than the Ph.D. salaries in industry and are often significantly lower for academic institutions with only UG students.
- MS CS and Ph.D. CS graduates obtain positions that often provide more fulfilling and enjoyable work. For instance, many companies will hire a BS CS graduate to be a test engineer while an MS CS graduate hired at that same company will develop software and/or

hardware for a product. We mention that many Ph.D. graduates decide to take academic positions despite receiving lower salaries because they find teaching and research more fulfilling than supporting the development or maintenance of a product in industry.

▸ We also describe the types of support that are offered by the FSU CS Department for graduate students, which include fellowships, research assistantships, and teaching assistantships. We explain that they will not have to pay any tuition if they are on support from the department. While we point out that a CS graduate student will not get wealthy from their graduate stipend, we show them that their pay is enough to support themselves without assistance from their parents or a student loan. We also point out that each FSU CS graduate student who is on support is assigned a desk in an office where they can study and perform their assigned duties.

▸ The FSU CS Department also established combined BS/MS programs, where a student can take up to four CS graduate courses and have these courses also count as CS undergraduate electives. We encourage all of the qualified CS UG students to enter these combined BS/MS programs to see how they would like taking graduate courses. The FSU CS faculty member who is the Director of Graduate Studies meets individually with these UG students in the combined BS/MS CS program each semester to provide advice about which graduate courses they should consider taking. Students who successfully complete one or more CS graduate courses gain confidence in their ability to obtain a CS graduate degree. We also point out that students who have taken three or more CS graduate courses by the time they complete their BS CS degree will very likely reduce the time they spend in their CS graduate studies by one semester.

Many CS faculty assume CS UG students are knowledgeable regarding the benefits of getting a graduate degree. Our experience with these individual meetings taught us that this is far from the truth: Many CS UG students, including high-achieving students, have very little knowledge about graduate school, including the benefits of getting a graduate degree, the typical activities of a graduate student, financial aid for graduate students, as well as the application process. For some reason, conventional methods of distributing graduate school information to domestic CS UG are ineffective. We suspect that some of these students may be the first in their family to attend a university and very often are the first in their family to enroll in a CS UG degree program. Given that CS domestic students usually have employment options when they graduate with a bachelor's degree, many of them will not even consider applying to a graduate program.

Recruiting domestic CS graduate students is possible, but it does take time and effort.

Over the years, these meetings with individual prospective graduate students have enjoyed great success. We believe these meetings not only benefited our department, but also the students. Note we are not advocating for recruiting UG students into the graduate program of the same school. Rather we emphasize that there are untapped qualified potential domestic applicants and that we must do a better job in reaching this group. If all U.S. universities can find effective methods to reach these potential applicants, the nation may not be in the situation of lacking domestic CS graduate students.

The FSU CS Department has also attempted to recruit domestic graduate students from other institutions. One approach that we have taken is to provide a travel reimbursement to visit the department for any admitted student with financial aid (fellowship, teaching assistantship, or research assistantship) to our CS graduate program that is currently residing in the lower 48 U.S. states. Most of these students are domestic students. We set up a schedule where the student will individually meet with FSU CS faculty during half-hour time slots. In contrast, many other departments will arrange a single time for all prospective students to visit their department. We have had many students state they were very impressed that our faculty spent the time to individually meet with them. Most of our CS faculty members are willing to spend the time to meet with these students as they realize that it is to their benefit to recruit as many well-qualified students into the FSU CS graduate program as possible.

One Step at a Time

We have found it is often difficult to convince CS UG domestic students to directly enroll in a Ph.D. program as that is a significant commitment of both time and effort. Obtaining an MS CS degree is viewed as much less formidable than obtaining a Ph.D. CS degree to the average CS UG student. We encourage our MS CS students to choose the MS thesis option so they can get some experience with CS research. A number of our FSU CS faculty members are willing to work with MS CS students on research projects as it is possible to get these students to accomplish some good research, and some of our best MS CS students do decide to stay for a Ph.D.

The FSU CS Department has also obtained grants from the NSF Cybercorps Scholarship for Service (SfS) program, which provides a number of scholarships for MS CS students. These SfS scholarship students are required to work for a government agency or a federally funded research and development center (FFRDC) after they graduate for a number of calendar years that is equivalent to the number of academic years of support they received from their scholarship. This SfS scholarship program requires the student either be a U.S. citizen or a U.S. permanent resident. We find the financial benefits of the SfS program are an excellent incentive for our students to obtain an MS CS degree. We do encourage students to not enroll in the SfS program if they are considering getting the Ph.D. degree as the service requirement may delay them from enrolling in a Ph.D. program.

The FSU CS Department also encourages our faculty members to actively recruit CS UG domestic students, often the top performers in their classes, into the FSU CS graduate program. Many of the FSU CS faculty members apply for an NSF Research Experience for Undergraduates (REU) supplement after obtaining an NSF grant or apply for other types of UG research grants. Few CS UG students will turn down an opportunity to get paid for working on a CS research project. CS UG students who become excited about working on a research project are much more likely to enroll in a CS graduate program and eventually get a Ph.D. degree.

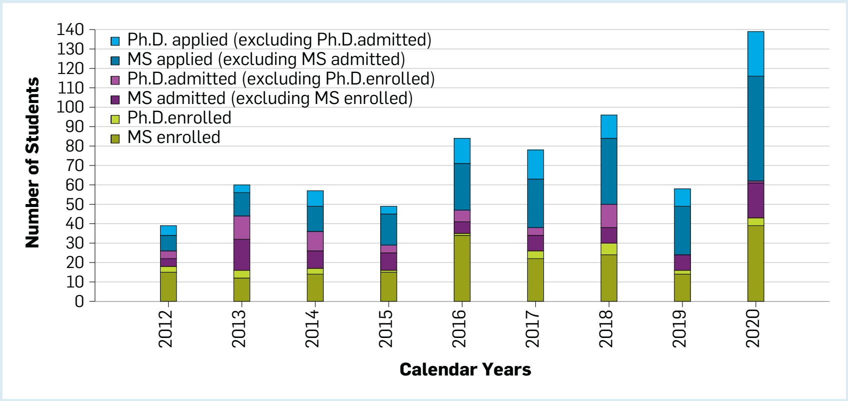
FSU CS Domestic Graduate Student Data

We realize the number of CS domestic graduate student applications can be affected by many factors. During recessions, domestic students are more likely to apply to graduate school as there are fewer employment opportunities. The economies in other countries can have an indirect impact on the number of domestic students who are awarded financial aid. In recent years, there has been a significant decline in the number of FSU CS graduate student applicants from India, which we believe is due, in part, to the improved employment opportunities in their economy. The timeliness of processing the graduate student applications to admit students and award financial aid can vary depending upon the efficiency of the CS graduate student admissions and financial aid committee and the office staff member who processes the CS graduate student applications. Both the composition of this committee and the CS staff member who processes the graduate student applications can change over time. Graduate student stipends can affect whether or not an applicant decides to enroll in a graduate program. The number of CS UG students affects the number of domestic CS graduate applicants, as we can get more applicants when there is a larger pool of CS UG students qualified to enter a CS graduate program. We found it became easier to recruit CS domestic students over the years as role models are important. Once a CS UG domestic student's friends start to attend a CS graduate program and these CS UG students are exposed to more domestic CS graduate teaching assistants, then they are much more likely to apply and attend a CS graduate program themselves.

Despite the effects of these various other factors, we believe we can still show that our efforts to recruit CS domestic graduate students has had a positive impact. The accompany figure shows the FSU CS domestic graduate student data for each calendar year from 2012–2020, which had similar improving economic conditions each year. We decided to report this information in calendar years as we tend to have individual meetings with prospective CS domestic graduate students in the Fall semesters. Note that the number of FSU CS UG students has been significantly increasing from 2012, but has become more stable since 2016. Similar increases in CS UG students have been seen across the nation. Having more CS UG students affects the number of prospective graduate students. From 2012–2015, the numbers of FSU CS graduate students applied and enrolled are significantly lower than those in more recent years. We believe this was in part due to the individual meetings with prospec-

tive graduate students from our own department being suspended in Fall semesters 2011–2013, which affected the number of FSU domestic CS graduate applicants from 2012–2014. From 2014–2020, we resumed our normal individual meetings with prospective graduate students from our own department. However, in 2014, the current graduate coordinator staff member to process the graduate student applications resigned and a replacement was not hired for several months, which resulted in some applications not being processed on time and caused some students to not complete the application process or enroll for the 2015 cycle. Thus, we also had a lower number of domestic CS graduate student applicants in 2015. One can see a significant improvement in applications and enrolled CS domestic graduate students from 2016 to 2018, which demonstrates the effectiveness of individually meeting with prospective graduate students. In fall 2018, the FSU CS department chair decided to indicate to the students he wished to discuss the benefits of attending graduate school rather than not revealing the reason for the meeting in his email message to each prospective CS graduate student. This resulted in a significant decline of the number of students meeting with the FSU CS department chair. This decline in meetings directly correlates to the decline in the number of FSU domestic CS graduate student applications and newly enrolled students in 2019. The FSU CS department chair did not reveal the reason for meeting with prospective CS graduate students in his email message in fall 2019, resulting in a large increase in the number of CS domestic graduate students in 2020. We believe this increase was also in part due to fewer job opportunities during the COVID-19 pandemic leading to more FSU CS domestic graduate student applicants. We also had fewer FSU CS graduate students enrolled from other countries as many U.S. embassies and consulates were closed during the pandemic causing our international graduate student applicants being unable to obtain a student visa. The indirect impact was that there was less competition for teaching or research assistant positions for FSU CS domestic graduate student applicants.

FSU CS domestic graduate student enrollment, admission, and application data.



We recognize the data shown in figure here is not perfectly correlated with our efforts to recruit CS domestic graduate students due to the effect of other factors previously described in this section. However, we know from our individual meetings with prospective graduate students that many of the students tell us they never considered going to graduate school and do eventually enroll in our CS graduate program or other CS graduate programs in the U.S.

Conclusion

Through these efforts previously described in this Viewpoint, the FSU CS Department for many years has been able to recruit a majority of students that are domestic among the supported graduate students that enter the FSU CS graduate program each year. While the majority of our domestic CS graduate students are MS students, a reasonable fraction is Ph.D. students. We believe we need to do a better job of convincing our MS CS domestic students to stay for the Ph.D. program.

We usually have the CS department chair and one other faculty member individually meet with our CS UG students to discuss the merits of attending graduate school. It takes approximately 20–30 minutes to meet with each of these students to discuss their plans and to describe the merits of attending graduate school. Viewpoint authors David Whalley and Xin Yuan were FSU CS department chairs previously and Xiuwen Liu is the current FSU CS department chair. All three of these department chairs made the commitment to meet personally with FSU CS UG students to discuss the merits of attending graduate school. This is quite a time commitment given we typically meet with 50+ students each year.

Recruiting domestic CS graduate students is possible, but it does take time and effort. We feel it is not only beneficial to the FSU CS department to recruit domestic CS graduate students, but it is also very beneficial to these students and the nation that we convince qualified domestic CS undergraduate students to attend a CS graduate program to help them reach their potential.

Engineering, mathematics, and the physical sciences are also fields that have a high fraction of international graduate students in the U.S. We believe the strategies we have utilized could increase the number of domestic graduate students in these other fields as well since our strategies are not specific to CS students. ■

David Whalley (whalley@cs.fsu.edu), **Xin Yuan** (xyuan@cs.fsu.edu), and **Xiuwen Liu** (liux@cs.fsu.edu) are all professors in the Florida State University computer science department in Tallahassee, FL, USA.

Sam Huckaba, the FSU Dean of the College of Arts and Sciences, provided some historical information regarding FSU graduate tuition waivers. The authors appreciate the efforts by the FSU Institutional Research for providing the data displayed in the figure and also appreciate the suggestions by the reviewers of this Viewpoint. This Viewpoint was supported in part by the U.S. National Science Foundation (NSF) under grants DUE-1259462, CCF-1533846, DGE-1565215, DRL-1640039, CICI-1738912, CRI-1822737, CCF-1823417, CCF-1900788, and SHF-2007827. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and may not reflect the views of the NSF.

DOI:10.1145/3464903

Article development led by acmqueue
queue.acm.org

A survey for practitioners.

BY RAMYA SRINIVASAN AND AJAY CHANDER

Biases in AI Systems

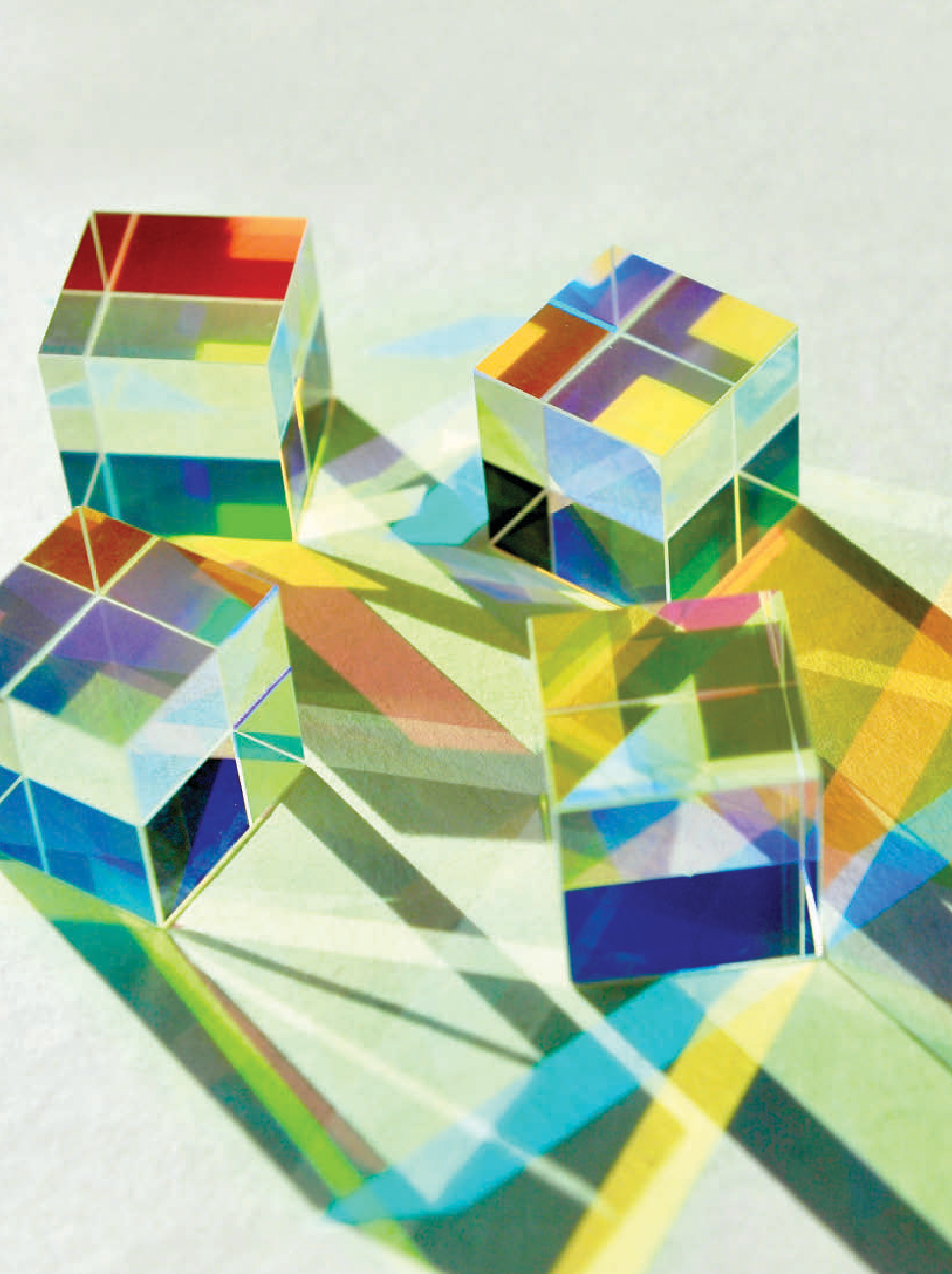
A CHILD WEARING sunglasses is labeled as a "failure, loser, nonstarter, unsuccessful person." This is just one of the many systemic biases exposed by ImageNet Roulette, an art project that applies labels to user-submitted photos by sourcing its identification system from the original ImageNet database.[7] ImageNet, which has been one of the instrumental datasets for advancing AI, has deleted more than half a million images from its "person" category since this instance was reported in late 2019.[23] Earlier in 2019, researchers showed how Facebook's ad-serving algorithm for deciding who is shown a given ad exhibits discrimination based on race, gender, and religion of users.[1] There have been reports of commercial facial-recognition software (notably Amazon's Rekognition, among others) being biased against darker-skinned women.[6,22]

These examples provide a glimpse into a rapidly growing body of work that is exposing the bias associated with AI systems, but biased algorithmic systems are not a new phenomenon. As just one example, in 1988, the U.K. Commission for Racial Equality found a British medical school guilty of discrimination because the algorithm used to shortlist interview candidates was biased against women and applicants with non-European names.[17]

With the rapid adoption of AI across a variety of sectors, including in areas such as justice and health care, technologists and policy makers have raised concerns about the lack of accountability and bias associated with AI-based decisions. From AI researchers and software engineers to product leaders and consumers, a variety of stakeholders are involved in the AI pipeline. The necessary expertise around AI, datasets, and the policy and rights landscape that collectively helps uncover bias is not uniformly available among these stakeholders. As a consequence, bias in AI systems can compound inconspicuously.

Consider, for example, the critical role of machine learning (ML) developers in this pipeline. They are asked to: preprocess the data appropriately, choose the right models from several available ones, tune parameters, and adapt model architectures to suit the requirements of an application. Suppose an ML developer was entrusted with developing an AI model to predict which loans will default. Unaware of bias in the training data, an engineer may inadvertently train models using only the validation accuracy. Suppose the training data contained too many young people who defaulted. In this case, the model is likely to make a similar prediction about young people defaulting when applied to test data. There is thus a need to educate ML developers about

IMAGE BY SUPERPUAY/SHUTTERSTOCK



the various kinds of biases that can creep into the AI pipeline.

Defining, detecting, measuring, and mitigating bias in AI systems is not an easy task and is an active area of research.[4] A number of efforts are being undertaken across governments, nonprofits, and industries, including enforcing regulations to address issues related to bias. As work proceeds toward recognizing and addressing bias in a variety of societal institutions and pathways, there is a growing and persistent effort to ensure that computational systems are designed to address these concerns.

The broad goal of this article is to educate nondomain experts and practitioners such as ML developers about various types of biases that can occur across the different stages of the AI pipeline and suggest checklists for mitigating bias. There is a vast body of literature related to the design of fair algorithms.[4] As this article is directed at aiding ML developers, the focus is not on the design of fair AI algorithms but rather on practical aspects that can be followed to limit and test for bias during problem formulation, data creation, data analysis, and evaluation. Specifically, the contributions can be summarized as follows:

- *Taxonomy of biases in the AI pipeline.* A structural organization of the various types of bias that can creep into the AI pipeline is provided, anchored in the various phases from data creation and problem formulation to data preparation and analysis.
- *Guidelines for bridging the gap between research and practice.* Analyses that elucidate the challenges associated with implementing research ideas in the real world are listed, as well as suggested practices to fill this gap. Guidelines that can aid ML developers in testing for various kinds of biases are provided.

The goal of this work is to enhance awareness and practical skills around bias, toward the judicious use and adoption of AI systems.

Biases in the AI Pipeline

A typical AI pipeline starts from the data-creation stage: collecting the data; annotating or labeling it; and preparing or processing it into a format that can be consumed by the rest of the pipeline. Let's analyze how different types of bias can be introduced in each of these steps.

Data-creation bias. Specific types of biases can occur during the creation of datasets.

Sampling Bias

The bias that arises in a dataset that is created by selecting particular types of instances more than others (and thereby rendering the dataset under-representative of the real world) is called *sampling bias*. This is one of the most common types of dataset biases. Datasets are often created with a particular set of instances. For example, image datasets prefer street scenes or nature scenes.[25] A face-recognition algorithm may be fed with more photos of light-skinned faces than dark-skinned faces, thereby leading to poor performance in recognizing darker-skinned faces.[6] Thus, sampling bias can result in poor generalization of learned algorithms.

Measurement Bias

Measurement bias is introduced by errors in human measurement, or because of certain intrinsic habits of people in capturing data. As an example, consider the creation of image and video datasets, where the images or videos may reflect the techniques used by the photographers. For example, some photographers might tend to take pictures of objects in similar ways; as a result, the dataset may contain object views from certain angles only. In their 2011 paper "Unbiased Look at Dataset Bias," Torralba and Efros refer to this type of measurement bias as *capture bias*.[25]

Another source of measurement bias could be a result of the device used to capture datasets. For example, cameras used to capture images may be defective, leading to poor-quality images and thereby contributing to biased results. These types of biases are broadly categorized as *device bias*.

A third type of measurement bias can occur when proxies are used instead of true values in creating the dataset. For example, arrest rates are often used instead of crime rates; doctor visits and medications are used as indicators of medical conditions, and so on.

Label Bias

Label bias is associated with inconsistencies in the labeling process. Different annotators have different styles and preferences that get reflected in the labels created. A common instance of label bias arises when different annotators assign differing labels to the same type of object (for example, *grass* vs. *lawn*, *painting* vs. *picture*).[25]

Yet another type of label bias can happen when the subjective biases of evaluators affect labeling. For example, in a task of annotating emotions experienced in a text, the labels could be biased by the subjective preferences of annotators such as their culture, beliefs, and introspective capabilities.[24] *Confirmation bias*,[21] which is the human tendency to search for, interpret, focus on, and remember information in a way that confirms one's preconceptions, is closely related to this type of label bias. Thus, labels may be assigned based on prior belief rather than objective assessments.

A third type of label bias can arise from the peak end effect. This is a type of memory-related cognitive bias in which people judge an experience based largely on how they felt at its peak (that is, its most intense point) and at its end, rather than based on the total sum or average of every moment of the experience.[15] For example, some annotators may give more importance to the last part of a conversation (rather than the entire conversation) in assigning a label.[24]

Negative Set Bias

Torralba and Efros define *negative set bias* as being introduced in the dataset as a consequence of not having enough samples representative of "the rest of the world."[25] The authors state that "datasets define a phenomenon (for example, object, scene, event) not just by what it is (positive instances), but also by what it is not (negative instances)." As a consequence, the learned classifiers can perform poorly in detecting negative instances.

Biases related to problem formulation. Biases can arise based on how a problem is defined. Consider the following example presented in *MIT Technology Review* by Karen Hao.[13] Suppose a credit card company wants to predict a customer's creditworthiness using

AI. In order to do so, creditworthiness must be defined in a manner that can be "predicted or estimated." The problem can be formulated based on what the company wants, say, to maximize its profit margin or to maximize the number of loans that get repaid; however, "those decisions are made for various business reasons other than fairness or discrimination," says Cornell University's Solan Barocas, who specializes in fairness.

Framing Effect Bias

The previous creditworthiness example can be thought of as a type of *framing effect bias*.[21] Based on how the problem is formulated and how information is presented, the results obtained can be different and perhaps biased. Another notable example is the COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) debate[8] concerning the definition of fairness between Northpointe (now known as Equivant), which came up with COMPAS scores for assessing risk of recidivism, and *ProPublica*, which claimed that the COMPAS system was biased. *ProPublica* claimed that Northpointe's method was biased against black defendants as the group was associated with a higher false-positive rate. There are several metrics of fairness, and *ProPublica* stated that Northpointe's system violated equalized odds and equality of opportunity fairness criteria. Northpointe's main defense was that scores satisfied fairness from the viewpoint of predictive rate parity.[4] Thus, bias can arise based on the way a problem and its success metrics are defined.

Biases related to the algorithm/data analysis. Several types of biases can occur in the algorithm or during data analysis.

Sample Selection Bias

Sample selection bias is introduced by the selection of individuals, groups, or data for analysis in such a way that the samples are not representative of the population intended to be analyzed.[9] In particular, sample selection bias occurs during data analysis as a result of conditioning on some variables in the dataset (for example, a particular skin color, gender, among others), which in turn can create spurious correlations. For example, in analyzing the effect of motherhood on wages, if the study is restricted to women who are already employed, then the measured effect will be biased as a result of conditioning on employed women.[9] Common types of sample selection bias include Berkson's paradox[20] and sample truncation.[9]

Based on how the problem is formulated and how information is presented, the results obtained can be different and perhaps biased.

Confounding Bias

Bias can arise in the AI model if the algorithm learns the wrong relations by not taking into account all the information in th e data or if it misses the relevant relations between features and target outputs.[20] *Confounding bias* originates from common causes that affect both inputs and outputs. Consider a scenario wherein admissions to a graduate school are based on the person's previous grade point average. There might be other factors, however, such as ability to get coaching, which in turn may be dependent on sensitive attributes such as race; and these factors may determine the grade point average and admission rates.[16] As a result, spurious relations between inputs and outputs are introduced and thus can lead to bias.

A special type of confounding bias is the *omitted variable*, which occurs when some relevant features are not included in the analysis. This is also related to the problem of model underfitting.

Another type of confounding bias is the *proxy variable*. Even if sensitive variables such as race and gender are not considered for decision making, certain other variables used in the analysis might serve as "proxies" for those sensitive variables. For example, zip code might be indicative of race, as people of a certain race might predominantly live in a certain neighborhood. This type of bias is also commonly referred to as *indirect bias* or *indirect discrimination*.

Design-Related Bias

Sometimes, biases occur as a result of algorithmic limitations or other constraints on the system such as computational power. A notable entry in this category is *algorithm bias*, which can be defined as bais that is solely induced or added by the algorithm. In their 1996 paper "Bias in Computer Systems," Friedman and Nissenbaum[10] provide an example: Software that relies on randomness for fair distributions of results is not truly random; for example, by skewing selections toward items at the end or beginning of a list, the results can become biased.

Another type of design-related bias is *ranking bias*.[18] For example, a search engine that shows three results per screen can be understood to privilege the top three results slightly more than the next three.[10] Ranking bias is also closely related to presentation bias,[18] which is derived from the fact that you can receive user feedback only on items that have been presented to the user. Even among those that are shown, the probability of receiving user feedback is further affected by where the item is shown.[2]

Biases related to evaluation/validation. Several types of biases result from those inherent in human evaluators, as well as in the selection of those evaluators (sample treatment bias).

Human Evaluation Biases

Often, human evaluators are employed in validating the performance of an AI model. Phenomena such as confirmation bias, peak end effect, and prior beliefs (for example, culture) can create biases in evaluation.[15] Human evaluators are also constrained by how much information they can recall, which can result in *recall bias*.

Sample Treatment Bias

Sometimes, test sets selected for evaluating an algorithm may be biased.[3] For example, in recommendation systems, some specific viewers (for example, those speaking a certain language) may be shown an advertisement, and some may not. As a consequence, the observed effects will not be representative of true effects on the general population. The bias introduced in the process of selectively subjecting some sets of people to a type of treatment is called *sample treatment bias*.

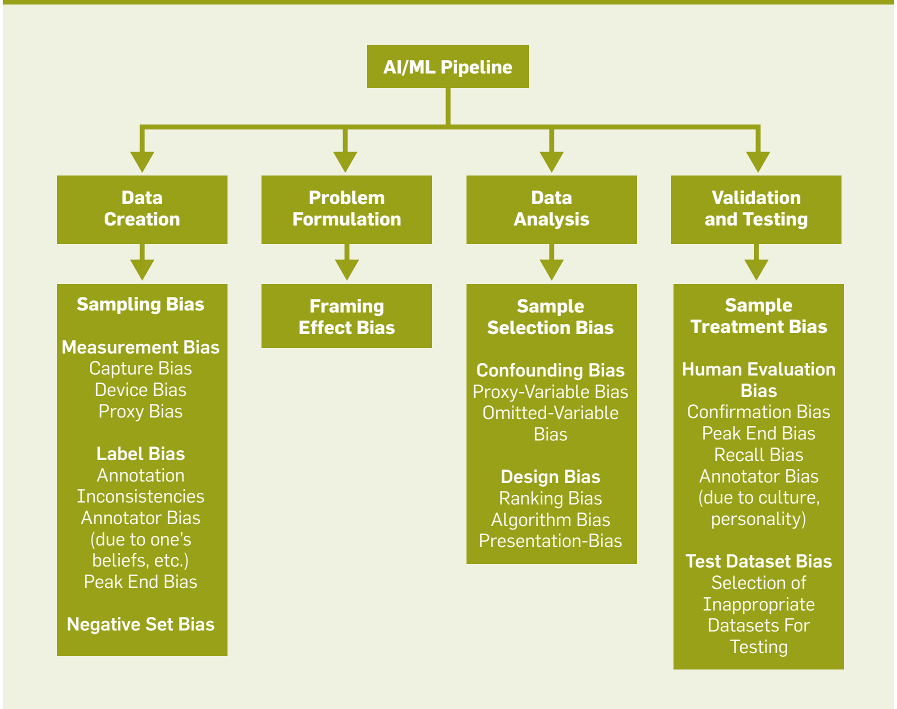
Validation and Test Dataset Biases

Biases can also be induced from sample selection and label biases in the validation and test datasets.[25] In general, biases associated with the dataset-creation stage could show up in the model-evaluation stage as well. Additionally, evaluation bias can result from the selection of inappropriate benchmarks/datasets for testing.

The accompanying figure provides an illustration of the taxonomy of biases along the various stages of the AI pipeline as discussed in the previous sections.

Despite significant research efforts within the AI community to address bias-related challenges, several gaps impede the collective progress. Next, we highlight some of these gaps.

Taxonomy of bias types along the AI pipleline.



Gaps Between Research and Practice

Methods to counter dataset bias issues have been proposed, as have new datasets with an emphasis on maintaining diversity. For example, the diversity-in-faces dataset consists of almost a million images of people pulled from the Yahoo! Flickr Creative Commons dataset, assembled specifically to achieve statistical parity among categories of skin tone, facial structure, age, and gender. In their 2019 paper, “Excavating AI,” Crawford and Paglen, however, question the use of cranio-metrical features used in creating this dataset, as these features could also be proxies for racial bias.[7] The authors further provide a critical review of issues pertaining to several benchmark datasets.

“Fairness in machine learning” is an active area of research. There are also conferences and workshops dedicated to the theme. A complete overview of fairness in machine learning is beyond the scope of this survey. For an extensive overview of various algorithmic definitions of fairness and methods to achieve fairness in classification, consult Barocas et al.[4] There are also open-source tools such as IBM's AI Fairness 360[5] that facilitates detection and mitigation of unwanted algorithmic bias. Despite these efforts, there are notable gaps, as noted by Gajane and Pechenizkiy in their 2018 paper, “On Formalizing Fairness in Prediction with Machine Learning.[11]

Filling the gap. Practice guidelines have been proposed for reducing the potential bias in AI systems. These include “Factsheets for Datasets” from IBM, and “Datasheets for Datasets,” an approach for sharing essential information about datasets used to train AI models.[12] In their 2019 paper, Mitchell et al. suggest the use of detailed documentation of released models in order to encourage transparency.[19]

Holstein et al. identify areas of alignment and disconnect between the challenges faced by teams in practice and the solutions proposed in the fair ML research literature.[14] The authors urge that future research should focus on supporting practitioners in collecting and curating high-quality datasets. The authors further see a need for creating domain-specific educational resources, metrics, processes, and tools. In that spirit, this article aims to be an educational resource for ML developers in understanding various sources of biases in the AI pipeline.

Guidelines for ML Developers

While it may not be possible to eliminate all sources of bias, with certain precautionary measures, some bias issues can be reduced. Here are some key messages that could aid ML developers in identifying potential sources of bias and help in avoiding the introduction of unwanted bias:

▸ Incorporation of domain-specific knowledge is crucial in defining and detecting bias. It is important to understand the structural dependencies among various features in the dataset. Often, it helps to draw a structural diagram illustrating various features of interest and their interdependencies. This can then help in identifying the sources of bias.[20]

▸ It is also important to understand which features of the data are deemed sensitive based on the application. For example, age may be a sensitive feature in determining who gets a loan, but not necessarily in determining who gets a medical treatment. Furthermore, there may be proxy features that, although not thought to be sensitive features, may still encode sensitive information so as to render biased predictions.

▸ As far as possible, datasets used for analysis should be representative of the true population under consideration. Thus, care has to be taken in constructing representative datasets.

▸ Appropriate standards have to be laid out for annotating the data. Rules have to defined so as to obtain consistent labels from annotators as much as possible.

▸ Identifying all features that may be associated with the target feature of interest is important. Omitting variables that have dependencies with the target feature leads to a biased estimate.

▸ Features that are associated with both input and output can lead to biased estimates. In such cases, it is important to eliminate these sources of confounding biases by appropriate data conditioning and randomization strategies in selecting input.[20]

▸ Restricting data analysis to some truncated portions of the dataset can lead to unwanted selection bias. Thus, in choosing subsets of data for analysis, care must be taken not to introduce sample selection bias.

▸ In validating the performance of a model such as in A/B testing, care has to be taken to guard against the introduction of sample treatment bias. In other words, in testing the performance of a model, the test conditions should not be restricted to a certain subset of the population (for example, showing recommendation results to people of a certain locality only), as the results would be biased.

Conclusion

This article provides an organization of various kinds of biases that can occur in the AI pipeline starting from dataset creation and problem formulation to data analysis and evaluation. It highlights the challenges associated with the design of bias-mitigation strategies, and it outlines some best practices suggested by researchers. Finally, a set of guidelines is presented that could aid ML developers in identifying potential sources of bias, as well as avoiding the introduction of unwanted biases. The work is meant to serve as an educational resource for ML developers in handling and addressing issues related to bias in AI systems. C

References

1. Ali, M., Sapiezynsk, P., Bogen, M., Korolova, A., Mislove, A., Rieke, A. Discrimination through optimization: how Facebook's ad delivery can lead to biased outcomes. In *Proceedings of the ACM on Human-Computer Interaction 3* (2019); https://dl.acm.org/doi/10.1145/3359301.
2. Amatriain, X. What does the concept of presentation-feedback bias refer to in the context of machine learning? Quora, 2015; https://www.quora.com/What-does-the-concept-of-presentation-feedback-bias-refer-to-in-the-context-of-machine-learning.
3. Austin, P.C., Platt, R.W. Survivor treatment bias, treatment selection bias, and propensity scores in observational research. , 2 (2010), 136–138; https://www.jclinepi.com/article/S0895-4356(09)00247-9/fulltext.
4. Barocas, S., Hardt, M., Narayanan, A. Fairness and machine learning: limitations and opportunities, 2019; https://fairmlbook.org.
5. Bellamy, R.K.E. et al. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. 2018, arXiv; https://arxiv.org/abs/1810.01943.
6. Buolamwini, J., Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of Machine Learning Research 81* (2018), 1–15; http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf.
7. Crawford, K., Paglen, T. Excavating AI: The politics of images in machine learning training sets. The AI Now Institute, New York University, 2019; https://www.excavating.ai.
8. Dressel, J., Farid, H. The accuracy, fairness, and limits of predicting recidivism. *Science Advances 4*, 1 (2018); https://advances.sciencemag.org/content/4/1/eaao5580.
9. Elwert, F., Winship, C. Endogenous selection bias: The problem of conditioning on a collider variable. *Annual Review of Sociology 40* (2014), 31-53; https://www.annualreviews.org/doi/full/10.1146/annurev-soc-071913-043455.
10. Friedman, B., Nissenbaum, H. Bias in computer systems. In *ACM Trans. Information Systems 14*, 3 (1996), https://dl.acm.org/doi/10.1145/230538.230561.
11. Gajane, P., Pechenizkiy, M. On formalizing fairness in prediction with machine learning. In *Proceedings of the Intern. Conf. Machine Learning, Fairness Accountability and Transparency Workshop*, 2018; https://www.fatml.org/media/documents/formalizing_fairness_in_prediction_with_ml.pdf.
12. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., Crawford, K. Datasheets for datasets. In *Proceedings of the 5th Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2018; https://www.microsoft.com/en-us/research/uploads/prod/2019/01/1803.09010.pdf.
13. Hao, K. This is how AI bias really happens—and why it's so hard to fix. *MIT Technology Review*; https://www.technologyreview.com/2019/02/04/137602/this-is-how-ai-bias-really-happensand-why-its-so-hard-to-fix/.
14. Holstein, K., Vaughan, J.W., Daumé III, H., Dudik, M., Wallach, H. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 SIGCHI Con. Human Factors in Computing Systems*, 1–16; https://dl.acm.org/doi/10.1145/3290605.3300830.
15. Kahneman, D. Evaluation by moments: past and future. *Choices, Values and Frames.* D. Kahneman and A. Tversky, Eds. Cambridge University Press, New York, 2000.
16. Kilbertus, N., Ball, P.J., Kusner, M.J., Weller, A., Silva, R. The sensitivity of counterfactual fairness to unmeasured confounding. In *Proceedings of the 2019 Conf. Uncertainty in Artificial Intelligence*; http://auai.org/uai2019/proceedings/papers/213.pdf.
17. Lowry, S., Macpherson, G. 1988. A blot on the profession. *British Medical J.* Clinical Research Ed. 296, 6623 (1988), 657; https://www.bmj.com/content/296/6623/657.
18. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A. A survey on bias and fairness in machine learning. 2019, arXiv; https://arxiv.org/abs/1908.09635.
19. Mitchell, M. et al. Model cards for model reporting. In *Proceedings of the 2019 AAAI/ACM Conf. AI, Ethics, and Society*; arXiv; https://arxiv.org/abs/1810.03993.
20. Pearl, J., Mackenzie, D. *The Book of Why: The New Science of Cause and Effect.* Basic Books, 2018.
21. Plous, S. *The Psychology of Judgment and Decision Making.* McGraw-Hill, 1993.
22. Raji, I., Buolamwini, J. Actionable auditing: investigating the impact of publicly naming biased performance results of commercial AI products. In *Proceedings of the 2019 AAAI/ACM Conf. AI, Ethics, and Society*, 429–435; https://dl.acm.org/doi/10.1145/3306618.3314244.
23. Small, Z. 600,000 images removed from AI database after art project exposes racist bias. *Hyperallergic*, 2019; https://hyperallergic.com/518822/600000-imagesremoved- from-ai-database-after-art-project-exposesracist- bias/.
24. Srinivasan, R., Chander, A. Crowdsourcing in the absence of ground truth—a case study. In *Proceedings of the 2019 Intern. Conf. Machine Learning Workshop on Human in the Loop Learning*; https://arxiv.org/abs/1906.07254.
25. Torralba, A., Efros, A.A. Unbiased look at dataset bias. In *Proceedings of the 2011 IEEE Con. Computer Vision and Pattern Recognition*, 1521–1528; https://ieeexplore.ieee.org/document/5995347.

Ramya Srinivasan is an AI researcher with Fujitsu Research of America. Her background is in the areas of computer vision, machine learning, explainable AI, and AI ethics.

Ajay Chander leads R&D teams in imagining and building new human-centered technologies and products. His work has spanned transparent AI, AI life assistants, digital healthcare and wellness, software tools design, security, and computational behavior design. He has received ACM's Most Influential Paper of the Decade award.

DOI:10.1145/3453182

Article development led by acmqueue
queue.acm.org

What was once a way to bring audio and video to the Web has expanded into more use cases than we could ever imagine.

BY NIKLAS BLUM, SERGE LACHAPELLE, AND HARALD ALVESTRAND

WebRTC: Real-Time Communication for the Open Web Platform

IN THIS TIME of pandemic, the world has turned to Internet-based, real-time communication (RTC) as never before. The number of RTC products has, over the past decade, exploded in large part because of cheaper high-speed network access and more powerful devices, but also because of an open, royalty-free platform called WebRTC.

In fact, over the past year, there has been a 100-fold increase of video minutes received via the WebRTC stack in the anonymous population that has opted into Google Chrome's statistics. WebRTC can be found in most Internet meeting services, social networks, live-streaming experiences, and even cloud-based gaming products.

WebRTC provides RTC capabilities to browsers and native apps. An open source implementation and tutorials for this platform can be found at https://webrtc.org. It includes audio and video codecs, and signal-processing functions such as bandwidth estimation, noise suppression, and echo cancellation.

This widely deployed communications platform powers audio/video calling, conferencing, and collaboration systems across all major browsers, both on desktop and mobile devices. This has enabled billions of users to interact. WebRTC has vastly expanded and facilitated the ability to create and deploy real-time, interactive services for startups and large-scale companies, and it can be found in commercial products and open source projects alike.

The idea for WebRTC originated in late 2009, more than a year after the launch of Google's Chrome browser. The Chrome team looked for functionality gaps between the desktop and the Web. While most of the discrepancies were already being addressed by ongoing projects, no solution existed for real-time communications. At the time, only Adobe's Flash and Netscape's NPAPI (Netscape Plugin API) provided RTC. Flash's offering was somewhat low quality and required a server license. Plug-ins are quite tricky for users to install, and few developers have the resources to handle deploying and updating plug-ins that work with three different browsers across several operating systems.

At about this time Google identified a company, Global IP Solutions (aka GIPS), that had the low-level components required for RTC. The GIPS components were licensed by several large customers and were present in products from Google, Skype, AOL, Yahoo!, Cisco, and others. By combining these audio and video components with a JavaScript interface, Google believed it

IMAGE BY QUSHE/SHUTTERSTOCK

could solve the big "hole" in its Web offerings and spur innovation in the RTC market. If a few lines of JavaScript code were all you needed to add RTC to a Web app—and with no licensing, integration of components, or deep knowledge of RTC required—who knew what could happen?

GIPS was based in Sweden and the U.S. and had engineers in both Stockholm and San Francisco. Luckily for Google, its audio and video Hangouts product was already being worked on in Stockholm, and having the GIPS engineers join in further reinforced the Stockholm office's strength as an RTC specialist within Google.

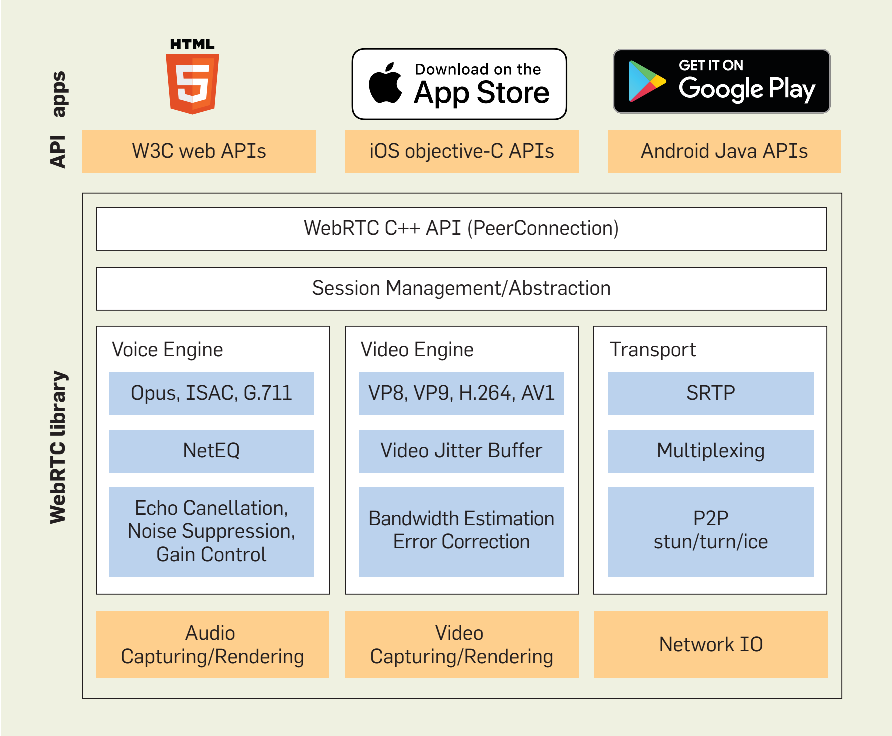
When the acquisition was completed in January 2011, the newly formed Chrome WebRTC team focused on integrating the code into Chrome and open sourcing all the key components at webrtc.org. From the beginning the plan was to build something open for the Web that would make RTC available for everyone.

Architecture and Functionality

A WebRTC peer may be a user endpoint (Web browser, native app, and so on) or a server that acts as an intermediary between two or more endpoints. While many WebRTC services rely on a client-server architecture, many others are deployed in a peer-to-peer (P2P, aka connection-less) architecture.

WebRTC is both an API and a protocol. The WebRTC protocol is a set of rules for two WebRTC agents to negotiate bidirectional secure real-time

Figure 1. WebRTC library componenets.



communication. The WebRTC API[23] then allows developers to use the WebRTC protocol.

The WebRTC API is specified only for JavaScript. The protocol to establish a connection between two WebRTC peers is a collection of other technologies, which can be split into signaling, connection management, security, and media transfer. These four steps usually happen sequentially. The prior step must be successful for the subsequent one to begin. Each step is actually made up of many other protocols.

As part of the WebRTC standards, many existing technologies that have been around since the early 2000s are combined and adapted for use in browsers and mobile applications.[17]

Figure 1 provides a high-level overview of the main components and technologies in WebRTC.

Android and iOS APIs are implementation-specific and not part of the standard, but they follow the same principles as the JavaScript APIs (webrtc.org open source implementation[18]). Audio and video capturing/rendering and network integration are specific to different operating systems.

PeerConnection API. The RTCPeerConnection API[21] is the central part of the WebRTC specification dealing with connecting two applications on different endpoints to communicate using a peer-to-peer protocol. The communication between peers can be video, audio, or arbitrary binary data (later we will discuss clients supporting the DataChannel API).

In order to discover how two peers can connect, both clients need to provide a STUN (session traversal utilities for NAT)[9] or a TURN (traversal using relays around NAT) server[3] configuration.[11] Their role is to provide an ICE (interactive connectivity establishment) candidate to each client, which is then transferred to the remote peer. This transferring of ICE candidates and exchange of other configuration information, such as media capabilities, is commonly called *signaling*.

Audio/video processing. WebRTC allows you to send and receive streams that include audio and/or video content. Streams can be added and removed at any time during a call; they can be either independent or bundled together. A common collaboration use case for RTC is to capture a computer's desktop content as a video feed and then include audio/video from the computer's Webcam and microphone. The WebRTC protocol in general is codec agnostic. The underlying transport has been designed to support any codec format; however, the WebRTC user agent capabilities with regard to media codecs have been subject to standardardization and are well defined.

The media functionality for processing audio and video provides the core of any WebRTC implementation. For audio communications and recording, Opus, G.711μ-law/A-law algorithms, and DTMF (dual-tone multi-frequency) have been defined as mandatory codecs.[16] The IETF standardization committees have agreed that WebRTC endpoints need to support the VP8 video codec and H.264 Constrained Baseline for processing video.[13]

Buffers in WebRTC implementations manage variability in packet arrival times, also called *jitter*, over the connection between peers. The logic of the buffering, managing of retransmission requests, and concealing data packets that have been lost or timed out is at the core of the signal processing work in WebRTC. These algorithms are constantly being developed and have seen major improvements over the past 10 years. The work greatly contributes to obtaining the best possible media quality when communicating over the Internet, especially when peers are connected to networks with different throughput levels and quality.

Security and media transport. WebRTC connections must be encrypted. This is both a core part of the design and part of the standardization. Two existing protocols, DTLS[12] (Datagram Transport Layer Security) and SRTP[2] (Secure Real-time Transport Protocol), have been adopted for this.

DTLS allows you to negotiate a session and then exchange data securely between two peers. SRTP is designed for exchanging media; it does not have a handshake mechanism and is bootstrapped with the external keys exchanged via DTLS:

1. DTLS does the handshake over the connection provided by ICE. During the DTLS handshake, both sides offer a certificate.

2. The SRTP session is created from the keys generated by DTLS.

3. With these steps completed successfully, SRTP-encrypted media can be exchanged between WebRTC peers.

Media flows between WebRTC

peers are by default based on UDP (User Datagram Protocol), meaning that the protocol has to handle unreliable delivery. To achieve the highest possible quality, the stack needs to make trade-offs between latency and quality. Generally speaking, the more latency you are willing to tolerate, the higher-quality video you can expect. For real-time voice communication, ITU-T (International Telecommunication Union-Telecommunications Standardization Sector) has defined the E-model,[7] which says that users start being dissatisfied when the mouth-to-ear delay becomes greater than 250 ms.

Congestion control is the mechanism by which WebRTC figures out what quality is achievable, given the latency constraints. Practically speaking, congestion control is being used by a bandwidth estimator adapting the media-encoding parameters for bit rate and video resolutions or audio frame sizes. This lowers the quality but assures that media keeps flowing when users have low or varying bandwidth available.

In the early days of WebRTC, it took, even under good conditions, on average 40 seconds or more to establish a connection and reach video quality of 720 pixels (HD) resolution. By setting aggressive goals, the time was pushed down to 100 ms, thanks to a collaboration with researchers from the Polytechnic University of Bari. This collaboration led to a new congestion-controller design;[4] Figure 2 shows the result of launching the congestion-control algorithm.

Data channels. In addition to sending real-time audio and video data, WebRTC allows sending and receiving arbitrary data via so-called data channels. Use cases for data channels range from file transfer, gaming, and IoT (Internet of Things) services to P2P CDNs (content delivery networks). The peer-to-peer data API[20] allows the creation of data channels. It extends the RTCPeerConnection API. SCTP (Stream Control Transmission Protocol)[15] is used as the underlying protocol to transport data channels. It includes channel multiplexing, reliable delivery with TCP-like retransmission mechanism, congestion avoidance, and flow control.

Standardization

At IETF 78 (summer 2010) in Maastricht, Google's nascent WebRTC team had an informal lunch with engineers from Microsoft, Apple, Mozilla, Skype, Ericsson, and others to gauge the interest in building such an RTC platform for the Web. A quickly organized one-day workshop[14] was held with the goal of understanding how such a standard should be written and defined. This led to intense activity in the W3C (World Wide Web Consortium) and the IETF, resulting in the formation of two working groups in May 2011: the IETF's RTCWeb[5] and the W3C's WebRTC,[27] both with participation from across the industry.

WebRTC in 2020

The adoption of WebRTC has come a long way. Most modern services that use voice or video are either based on the WebRTC protocols or have the ability to use them in addition to the native protocols the service originally deployed with. Cisco's Webex service, for example, has a WebRTC client that lets people participate in conferences directly from their browsers without downloading additional software. Newer services, such as whereby.com and Jitsi, have been natively based on WebRTC from the outset. Even when no Web browser is involved, major services use WebRTC for video transmission. For example, WebRTC enables the Amazon Ring product to view security camera and doorbell footage. Increasingly, new IoT products that stream voice and/or video are basing their network stacks on the WebRTC protocols.[3]

2020 was a year unlike any before. The need for RTC has been highlighted by Covid-19, as people across the globe have found new ways to work, educate, and connect with loved ones via video

Figure 2. Ramp-up time to 1Mbps video bit rate.

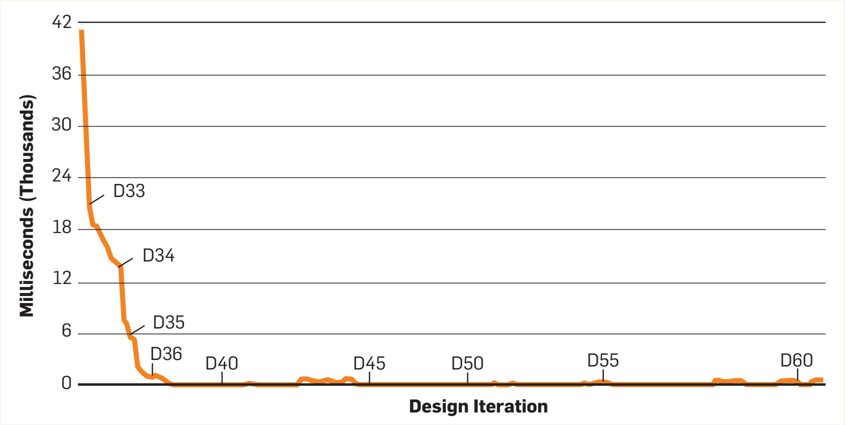
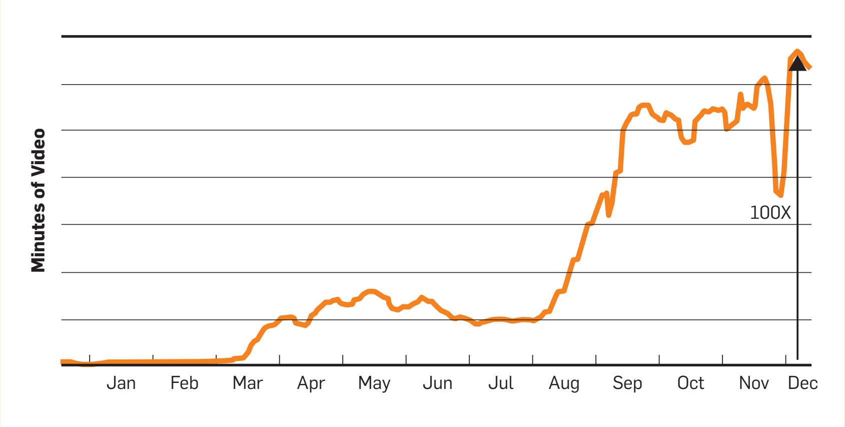


Figure 3. Video minutes in Chrome over WebRTC in 2020.



chat. WebRTC has suddenly become one of the most important sets of technologies allowing Web browsers to make voice, video, and real-time data calls. It has allowed for an ecosystem of interoperable communications apps to flourish: Since the beginning of March 2020, Chrome has seen a 100-fold increase in received video streams via WebRTC, excluding incognito and users opted by default out of sharing stats (see Figure 3).

These successes would not have been possible without all the supporters that make an open source community. An important element of this success is all the code contributors, testers, bug filers, and corporate partners who helped make this ecosystem a reality.

Outlook

Google is a founding member of AOMedia (Alliance for Open Media) and has been active in defining the AV1 video bitstream for the RTC use case. As AV1 has become a standard, the video codec is being integrated into WebRTC. Chrome version 89 is shipping an AV1 software encoder providing AV1-to-Web applications for RTC. AV1 provides another 30%–50% bit-rate savings at the same quality compared with VP9, and is expected to offer another level of bandwidth efficiency and quality for video-calling services. Because of the complexity of the codec, hardware support will be of great importance to make it ubiquitously available. AV1 will be critical in facilitating RTC services to scale further and in allowing for higher-quality video experiences in the future.

WebRTC goes beyond voice and video communication. Emerging gaming, low-latency video streaming, AR/VR (augmented reality/virtual reality), and mixed-reality services are equally benefiting from and demanding low-latency media. For example, WebRTC enables the Stadia gaming service to bring cloud-based, low-latency, high-quality experiences to Web browsers and televisions.

These use cases push the latency barrier, resulting in the need for further transport protocol optimizations. The corresponding standardization effort to cover this need is WebTransport,[6,26] focusing on optimizing for super-low-latency client-server media streaming via the QUIC protocol.

As new use cases for WebRTC emerge, the WebRTC standardization is evolving into what is called WebRTC NV (Next Version).[25] NV will not be a completely new API but will allow access to the lower-level media pipeline inside PeerConnections. Media will become accessible using the Streams[19] and WebCodecs APIs.[22] A first step in this direction is the already implemented Insertable Streams API[24] that provides the foundation for full E2EE (end-to-end encryption) multiparty conferencing in browsers.[8]

WebRTC's reach into mobile devices started through the native (that is, non-Web) integration into mobile social media, messaging, and video calling apps. With emerging 5G networks, video calling will become even more of a commodity.

WebRTC's open architecture also allows for interesting innovations using machine learning and artificial intelligence to augment call quality and hide the effects of noise[10] or network disruptions.[1]

What started as a way to bring audio and video to the Web has expanded into more use cases than could be imagined—from simple video calling to AR/VR experiences, cloud-based gaming, and massively scalable live streaming services; and from simple point-to-point video chat to multiuser conversations where quality is augmented through advanced machine-learning models. Most importantly, WebRTC is growing from enabling useful experiences to being essential in allowing billions to continue their work and education, and keep vital human contact during a pandemic. The opportunities and impact that lie ahead for WebRTC are intriguing indeed.

References

1. Barrera, P., Stimberg, F. Improving audio quality in Duo with WaveNetEQ. Google AI Blog (Apr. 1, 2020); https://ai.googleblog.com/2020/04/improving-audio-quality-in-duo-with.html.
2. Baugher, M., McGrew, D., Naslund, M., Carrara, E., Norrman, K. The Secure Real-time Transport Protocol, IETF RFC 3711, 2004; https://tools.ietf.org/html/rfc3711.
3. Gross, G. WebRTC technologies prove to be essential during pandemic. IETF interview with Adam Roach (Dec. 8, 2020); https://www.ietf.org/blog/webrtc-pandemic/.
4. Holmer, S., Lundin, H., Carlucci, G., De Cicco, L., Mascolo, S.; H. Alvestrand, eds. A Google congestion control algorithm for real-time communication, 2015; https://tools.ietf.org/html/draft-alvestrand-rmcat-congestion-03.
5. IETF. Real-time communication in Web-browsers (RTCWeb) working group; https://datatracker.ietf.org/wg/rtcweb/documents/.
6. IETF. WebTransport (webtrans), 2021; https://datatracker.ietf.org/wg/webtrans/about/.
7. International Telecommunication Union-T. G.107: The E-model: A computational model for use in transmission planning, 2015; https://www.itu.int/rec/T-REC-G.107-201506-I/en.
8. Ivov, E. This is what end-to-end encryption should look like! (Apr. 12, 2020). Jitsi blog; https://jitsi.org/blog/e2ee/.
9. Petit-Huguenin, M., Salgueiro, G., Rosenberg, J., Wing, D., Mahy, R., Matthews, P. Session Traversal Utilities for NAT. IETF RFC 8489, 2020; https://tools.ietf.org/html/rfc8489.
10. Protalinski, E. Google Meet noise cancellation is rolling out now—here's how it works. VentureBeat (June 8, 2020); https://venturebeat.com/2020/06/08/google-meet-noise-cancellation-ai-cloud-denoiser-g-suite/.
11. Reddy, T., Johnston, A., Matthews, P., Rosenberg, J. Traversal using relays around NAT (TURN): Relay extensions to session traversal utilities for NAT (STUN). IETF RFC 8656, 2020; https://tools.ietf.org/html/rfc8656.
12. Rescorla, E., Modadugu, N. Datagram Transport Layer Security, version 1.2. IETF RFC 6347, 2012; https://tools.ietf.org/html/rfc6347.
13. Roach, A.B. WebRTC video processing and codec requirements. IETF RFC 7742, 2016; https://tools.ietf.org/html/rfc7742.
14. RTC-Web Workshop. 2010; http://rtc-web.alvestrand.com/.
15. Stewart, R., Ed. Stream Control Transmission Protocol. IETF RFC 4960, 2007; https://tools.ietf.org/html/rfc4960.
16. Valin, J.M., Bran, C. WebRTC audio codec and processing requirements. IETF RFC 7874, 2016; https://tools.ietf.org/html/rfc7874.
17. WebRTC for the Curious. What is WebRTC? (Sept. 19, 2020); https://webrtcforthecurious.com/docs/01-what-why-and-how/.
18. WebRTC.org implementation. Google Git; https://webrtc.googlesource.com/src/.
19. W3C. Streams API (Nov. 29, 2016); https://www.w3.org/TR/streams-api/.
20. W3C. Peer-to-peer Data API (Dec. 15, 2020); https://www.w3.org/TR/webrtc/#peer-to-peer-data-api.
21. W3C. RTCPeerConnection interface (Dec. 15, 2020); https://www.w3.org/TR/webrtc/#rtcpeerconnection-interface.
22. W3C. WebCodecs (Dec. 8, 2020); https://wicg.github.io/web-codecs/.
23. W3C. WebRTC 1.0: Real-time communication between browsers. W3C Proposed Recommendation (Dec. 15, 2020); https://www.w3.org/TR/webrtc/.
24. W3C. WebRTC insertable media using Streams (Sept. 1, 2020); https://w3c.github.io/webrtc-insertable-streams/.
25. W3C. WebRTC Next Version use cases (Nov. 30, 2020); https://www.w3.org/TR/webrtc-nv-use-cases/.
26. W3C. WebTransport (Dec. 9, 2020); https://w3c.github.io/webtransport/.
27. W3C. Web Real-Time Communications working group; https://www.w3.org/groups/wg/webrtc.

Niklas Blum is a group product manager at Google. He leads the strategy and execution for the audio/video calling experience in Google's video communication products, including Google Meet, Google Duo, and Chrome/WebRTC. He has spent 15-plus years in the communications space.

Serge Lachapelle is director of product management at Google. He has spent more than 20 years in the video communications industry, starting as the cofounder of Marratech AB, which was acquired by Google in 2007. At Google, Lachapelle started many video-calling initiatives, including Gmail Video Chat, Google Hangouts, WebRTC, Google Duo, and Google Meet.

Harald Alvestrand is the standards coordinator for the WebRTC project at Google. He has been an evangelist for open communication across company borders for over 35 years. He has been a member of the board at ICANN, chair of the IETF, and area director of the IETF Applications Area.

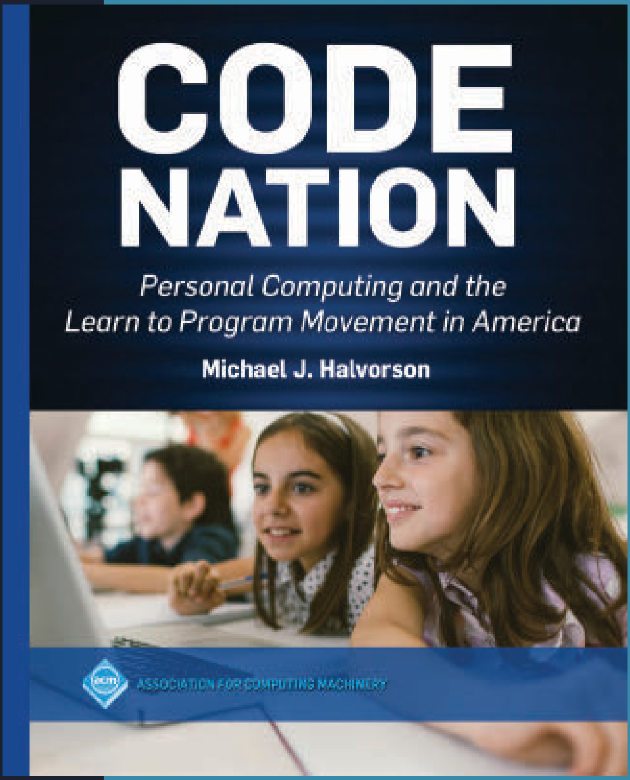
ACM BOOKS

Collection II

Code Nation explores the rise of software development as a social, cultural, and technical phenomenon in American history. The movement germinated in government and university labs during the 1950s, gained momentum through corporate and counterculture experiments in the 1960s and 1970s, and became a broad-based computer literacy movement in the 1980s. As personal computing came to the fore, learning to program was transformed by a groundswell of popular enthusiasm, exciting new platforms, and an array of commercial practices that have been further amplified by distributed computing and the Internet. The resulting society can be depicted as a "Code Nation"—a globally-connected world that is saturated with computer technology and enchanted by software and its creation.

Code Nation is a new history of personal computing that emphasizes the technical and business challenges that software developers faced when building applications for CP/M, MS-DOS, UNIX, Microsoft Windows, the Apple Macintosh, and other emerging platforms. It is a popular history of computing that explores the experiences of novice computer users, tinkerers, hackers, and power users, as well as the ideals and aspirations of leading computer scientists, engineers, educators, and entrepreneurs. Computer book and magazine publishers also played important, if overlooked, roles in the diffusion of new technical skills, and this book highlights their creative work and influence.

Code Nation offers a "behind-the-scenes" look at application and operating-system programming practices, the diversity of historic computer languages, the rise of user communities, early attempts to market PC software, and the origins of "enterprise" computing systems. Code samples and over 80 historic photographs support the text. The book concludes with an assessment of contemporary efforts to teach computational thinking to young people.



CODE NATION

Personal Computing and the Learn to Program Movement in America

Michael J. Halvorson

ASSOCIATION FOR COMPUTING MACHINERY

Code Nation

Personal Computing and the Learn to Program Movement in America

Michael J. Halvorson

ISBN: 978-1-4503-7757-7
DOI: 10.1145/3368274

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

DOI:10.1145/3470569

A panoramic view of a popular platform for C program analysis and verification.

BY PATRICK BAUDIN, FRANÇOIS BOBOT, DAVID BÜHLER, LOÏC CORRENSON, FLORENT KIRCHNER, NIKOLAI KOSMATOV, ANDRÉ MARONEZE, VALENTIN PERRELLE, VIRGILE PREVOSTO, JULIEN SIGNOLES, AND NICKY WILLIAMS

The Dogged Pursuit of Bug-Free C Programs: The Frama-C Software Analysis Platform

THE C PROGRAMMING language is a cornerstone of computer science. Designed by Dennis Ritchie and Ken Thompson at Bell Labs as a key element of Unix engineering, it was rapidly adopted by system-level programmers for its portability, efficiency, and relative ease of use compared to assembly languages. Nearly 50 years after its creation, it is still widely used in software engineering.

But C is a difficult language to wield. Its native design choices give developers much freedom—a reason for its popularity—but that can often clash with the requirements of modern development practices, such as strong typing, encapsulation, or genericity. Given its ubiquity in software engineering, this has had noticeable safety and, more recently, cybersecurity impacts. The use of verification techniques, and in the case of systems with high-confidence requirements, formal methods, can address these shortcomings.

Indeed, formal methods are a set of techniques based on logic, mathematics, and theoretical computer science which are used for specifying, developing, and verifying software and hardware systems. By relying on solid theoretical foundations, formal methods can provide strong guarantees about those systems. In particular, program analysis techniques focus on the program code *after* it has been written or even compiled. Such techniques are called *sound* if their results are correct with respect to the behavior of the program under analysis.

Unfortunately, implementing such techniques for C programs is difficult. Indeed, the same issues that make C programming very error-prone also tend to complicate the task for formal methods-based analyzers. It is very easy to write an illegal program whose behavior is *undefined* by the C standard: it can, for instance, provoke a crash or sometimes silently corrupt memory and lead to arbitrary results.

» key insights

- The C programming language remains popular for system-level programming and embedded code in many critical domains. Verification and validation is crucial to making software-dependent services reliable and secure.
- Relying on solid theoretical foundations, formal methods can offer strong guarantees about the software in those systems.
- Frama-C, a collaborative, open source platform for code analysis and verification based on formal methods, offers several C code analysis plug-ins.
- Frama-C attracted a large community of academic and industrial users, thanks to its open source license and a regular stream of releases since 2008.

IMAGE BY ANDRIJ BORYS ASSOCIATES

Examples of such behavior include division by 0, illegal memory access, and reading uninitialized variables. In particular, the fact that C allows direct access, through casts and pointer arithmetic, to the sequence of bytes that contain the concrete representation of an object in memory is a major impediment to any attempt to reason on these objects at a more abstract level. Yet, many functions from the C standard library, starting with memcpy for copying an object to another location in memory, will trigger such low-level access, not to mention their presence in many parts of user-defined code.

Frama-C[26] is a C code analysis platform that attempts to tackle this complex issue. It is developed at CEA List with a few key ideas at its core. First and foremost, it acknowledges the fact that there is no silver bullet in software verification: no single technique will ever be able to succeed in assessing all properties a user can be interested in. Thus, the platform should foster collaborations between various techniques, by letting individual analyzers exchange information about the properties they can handle as well as the hypotheses they make during the analysis (in the hope that another analyzer may be able to validate them).

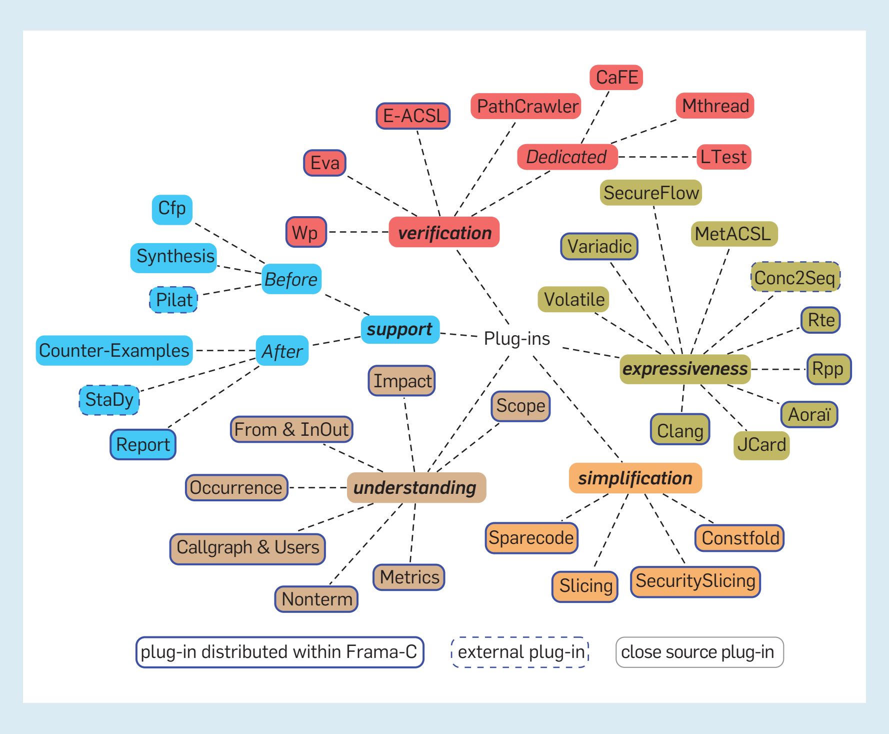
In a similar manner, the platform was meant to be easily extensible, in particular, by third-party developers. This is also reflected by the choice of the LGPL license for open-source releases of the tool, which allows the development of proprietary plug-ins as long as any change made to the core platform is contributed back.

Finally, Frama-C is meant to be usable by software engineers who are not necessarily experts in formal methods. This implies providing as much automation as possible, as well as assessing the performances of the platform on real-world case studies. The purpose of this article is to provide a panorama of the platform, its key design choices, and its uses.

Since its first public release in 2008, Frama-C has been continuously evolving. An active R&D is conducted to bring well-established program analysis techniques, such as abstract interpretation or weakest precondition calculus, to the level of industrial-strength tools.

In parallel, novel techniques are developed for specific analysis tasks; for example, for specification and verification of specific kinds of properties coming into focus with the increasing complexity of modern software or enhancing existing techniques with new approaches. This article illustrates efforts of both kinds.

Figure 1. Frama-C plug-in gallery.



Platform Overview

Frama-C allows users to analyze a given C program, better understand or even simplify it, and assess properties about it. The user can, for instance, explore the program structure and compute some metrics on it. Program properties can be explicitly expressed as annotations written in the formal specification language ACSL (described later). They can be validated by partial, dynamic verification or formally verified by rigorous, static verification.

Frama-C is not a single tool, but a framework that groups together several tools, each provided as a plug-in. Frama-C 21-Scandium, the latest open-source release (at time of writing), contains 27 plug-ins. Frama-C offers an extensible and collaborative setting: anyone can develop and provide new plug-ins, which can collaborate with each other in different ways.

Different plug-ins for different analyses. Figure 1 shows a selection of Frama-C plug-ins. Verification plug-ins are the most important ones. The value-analysis plug-in Eva focuses on detecting undefined behaviors (often called *runtime errors*) and tries to prove their absence. For example, for the code `if(*p<0) *p = -(*p);` where `p` is of type `int*`, it has to check that reading and writing `*p` is safe, and that `-(*p)` does not overflow, that is, `*p` $\neq -2^{31}$, because the type int (over 32 bits) can only express values $-2^{31} \ldots (2^{31} - 1)$. For a division, it has to check that the denominator is not 0. Eva does not require additional annotations: potential runtime errors can be deduced from the code.

On the contrary, proving program-specific, *functional* properties requires first to specify them as ACSL annotations. For the previous example, such annotations can state that it computes the absolute value of `*p`. Then, such properties can be proved using the deductive verification plug-in WP. It can also require additional proof-guiding annotations or even a user-guided, *interactive* proof.

Sometimes, when such properties are not (yet) proved, the user can automatically verify them at runtime for a given execution using E-ACSL. The user can also automatically generate test inputs and check for these inputs that the program behaves as expected using PathCrawler. A few other plug-ins are specialized, such as CaFE for temporal properties, Mthread for concurrency properties, and LTest for test automation.

Several plug-ins are aimed at *supporting the verification process*, either before or after the run of verification plug-ins. Cfp[1] prepares the analysis with Eva for a given library function specified with an ACSL contract by inferring a suitable analysis context. Synthesis automatically generates a function body implementing a given function contract. Pilat[16] infers necessary proof-guiding annotations for loops (as polynomial loop invariants) for a proof with WP. In case of a proof failure, StaDy and Counter-Examples aim at generating a counter-example. Report summarizes what has (or has not yet) been verified.

Other plug-ins help verification engineers to better understand the analyzed code: From, InOut, Impact, Scope, and Occurrence detail dependency and scope information related to memory locations. Callgraph and Users provide information about function calls, while Nonterm warns about non-terminating code. Metrics provides some code metrics.

A few plug-ins are program transformers that simplify the analyzed code. Constfold performs constant propagation, while Slicing removes pieces of code that are irrelevant with respect to a specific criterion. Sparecode and SecuritySlicing[32] perform specialized simplifications, removing non-executable, *dead* code or code irrelevant to confidentiality/integrity properties.

Last but not least, several plug-ins extend the expressiveness of other analyzers. Frama-Clang and JCard target C++ and JavaCard code; Volatile and Variadic specifically deal with volatile memory locations and variadic functions; while RTE, Aoraï, RPP, MetACSL, Conc2Seq, and SecureFlow automatically generate ACSL properties from higher-level or implicit specifications.

Plug-in collaboration. No program-analysis technique is perfect by nature: many program-analysis problems are *undecidable*. In other words, it is impossible to create a tool capable of solving them for all programs. However, some approaches and tools are more efficient for particular kinds of properties or programs than others. Frama-C promotes analyzer collaboration to leverage the benefits and strengths of different tools. It can be used to decompose verification work and comes in two different flavors: *sequential* and *parallel*.

Frama-C plug-ins are based on a kernel that provides key services to both end users and plug-in developers.

Sequential collaboration uses the result of one analyzer as the input to another one. It can also generate annotated C code that encodes a verification problem in such a way that another analyzer can understand it. The plug-ins allow such collaborations in the "Support" and "Expressiveness" categories of Figure 1. Several examples are provided below.

Parallel collaboration uses several analyzers to verify program properties, with each analyzer verifying a subset of properties. For instance, Eva can verify the absence of undefined behaviors, while WP can prove functional properties. Eventually, the few remaining properties may be checked at runtime by E-ACSL. Frama-C ensures the consistency of partial results emitted by the analyzers and summarizes what has been verified and what remains to be.[14]

Platform architecture. Frama-C plug-ins are based on a *kernel* that provides key services to both end users and plug-in developers. The kernel contains three main components: (1) basic services (such as program parsing) that build a normalized representation (called Abstract Syntax Tree, or AST) of the analyzed program, (2) specialized services (for example, exploring and manipulating the program AST, including ACSL annotations) for code analyses, and (3) general-purpose libraries. Altogether, they provide a large API, providing useful services to analyzers and facilitating plug-in development. This makes it possible to develop, within a few days, a brand-new prototype analyzer supporting most C constructs.

ACSL Specification Language

For specifying C code, Frama-C offers ACSL, the ANSI/ISO C Specification Language.[a] ACSL clauses (*annotations*) are written in special comments //@... or /*@... */. While ACSL is a fairly rich

[a] https://github.com/acsl-language/acsl/releases/tag/1.14

Figure 2. Example of ACSL function contract.

```
1  /*@ requires \valid(p);
2      requires *p > INT_MIN;
3      assigns *p;
4      ensures ( \old(*p) ≥ 0 ⇒ *p == \old(*p) ) ∧
5        ( \old(*p) < 0 ⇒ *p == -\old(*p) );
6  */
7  void pabs(int *p){
8    if (*p < 0)
9      *p = -(*p);
10 }
```

Figure 3. Example of ACSL loop contract.

```
1 int i = 0, j = 10, k = 12;
2 /*@ loop invariant 0 ≤ i ≤ 10;
3     loop invariant i+j == 10;
4     loop assigns i,j;
5 */
6 while (i < 10) { i++; j--; }
7 //@ assert j == 0;
8 //@ assert k == 12;
```

language, we give only a very brief description in this article. Interested readers can refer to existing tutorials[b] for an in-depth presentation.

As mentioned previously, Frama-C can be used to check that no input leads to a runtime error (RTE) in a given program. Such checks can be generated by the RTE plug-in as ACSL assertions, for verification by other plug-ins (Eva, WP, or E-ACSL). An ACSL assertion (**assert** clause) can be put anywhere in the code to indicate that a property must hold at this particular point. For the code example **if**(*p<Ø) *p = –(*p); we considered earlier, RTE generates the following (simplified) assertions:

```
//@ assert \valid ( p ) ;
if(*p<Ø) {
//@ assert * p>INT_MIN ;
*p = - (*p ) ;
}
```

These assertions indicate precisely the required properties: (i) pointer p is *valid*, that is, *p can be safely read/written, and (ii) *p should be greater than the minimal value of type int. As we will show later, lines 13–14 of Figure 4 show an assertion to prevent a division by 0.

Obviously, such properties only ensure the absence of undefined behaviors. They do not mean the program behaves as intended. To verify its *functional* properties—the intended behavior—it is necessary to have a precise, formal description of what this intended behavior is. Such a description can also be expressed in ACSL.

A key ingredient of ACSL is the notion of *function contract*, which can be traced back to Eiffel and Meyer's *Design by Contract*.[31] Basically, a (function) contract defines some constraints on the state in which the function might be called (the *precondition*), and in exchange provides some guarantees about the state in which it returns control to its caller (the *postcondition*). It is also important to define which parts of the state (that is, which variables or memory locations) can be modified during the execution of the function (the *frame rule*). Thanks to it, the caller knows that everything that is not in the frame is left untouched.

Figure 2 shows a possible contract for a simple function with the considered conditional statement. The requires clause expresses the precondition (lines 1–2), denoted Pre$_{\text{pabs}}$. It states that function pabs expects to be called with an argument p that is a valid pointer, and the pointed value is greater than the minimal value of type int.

This precondition guarantees the absence of runtime errors in the function. The ensures clause (lines 4–5) expresses the postcondition Post$_{\text{pabs}}$, which states that the resulting value of *p is the absolute value of its initial (old) value. Furthermore, pabs is supposed to modify only *p, as indicated by the assigns clause on line 3.

Another important ingredient of ACSL is a *loop contract*. Placed in front of a loop, it contains clauses providing additional information to reason about loop behavior. It includes a loop invariant, stating properties that hold when entering the loop for the first time and are preserved after each loop step. Hence, by induction, they also hold at the end of the loop, regardless of the number of steps.

As for functions, loops also have frame rules, introduced by **loop assigns**. Figure 3 illustrates these annotations (see lines 2–4) on a very simple loop manipulating i and j together in order to keep their sum constant, while leaving variable k untouched. Lines 7–8 contain two assertions that hold after the loop. We will illustrate below how loop contracts help to reason for programs with loops.

ACSL uses first-order logic formulas, with integer and real arithmetic. Unlike the bounded C types, ACSL's integer and real types are unbounded. In particular, this makes it easier to write annotations stating the absence of arithmetic overflow. For instance, assuming *x* and *y* are C variables of type int (hence also their sum), the following assertion will guarantee that their sum can be safely computed in C, without triggering an overflow:

```
/*@ assert INT_MIN ≤ x + y ≤ INT_MAX ; */
```

Finally, ACSL features several built-in predicates for stating properties over the pointers manipulated by the program.

Core Platform Analyses

Frama-C has four core plug-ins: Eva, which focuses on detecting undefined behaviors; WP, which aims to prove functional properties; E-ACSL, which checks properties at runtime; and PathCrawler, which generates test cases.

Chasing undefined behaviors with Eva. The Eva plug-in provides a configurable and automatic analysis of the whole program, intended to prove the absence of undefined behaviors. This term refers to instructions for which the C standard imposes no requirements, leading to crashes and more generally unpredictable execution flow. They can, in particular, result in security vulnerabilities, and attackers frequently exploit such illegal instructions to steal data or execute malware.

Eva detects most undefined behaviors, such as invalid memory accesses,

[b] https://github.com/fraunhoferfokus/acsl-by-example/raw/master/ACSL-by-Example.pdf, https://allan-blanchard.fr/publis/frama-c-wp-tutorial-en.pdf

uninitialized memory reads, divisions by zero, signed integer overflows, undefined bit shifts, and invalid pointer comparisons. It can also treat as erroneous some behaviors that are allowed by the standard but often unwanted by developers, such as unsigned integer overflows or exceptional floating-point values (for example, infinities).

Eva is based on a technique called *abstract interpretation*. The goal of the analysis is to compute a set of possible values for each variable at each program point. Since computing these sets precisely is undecidable (as we explained in the Plug-In Collaboration section), Eva uses *abstractions* to over-approximate them.

For instance, if the set of values D_v^l of a variable v at program point l is $\{1, 3, 5, \ldots, 97, 99\}$, it can be approximated as the integer interval [1, 99]. If v takes value v_0 at point l for some execution, v_0 will necessarily belong to the approximated set D_v^l. The contrary is not true: the set D_v^l can contain values that v never takes at point l in practice. Thus, the computed abstractions build a sound over-approximation of all possible behaviors. As a consequence, Eva is sound: its analysis is exhaustive and reports *all* undefined behaviors that could happen in an execution of a program.

Let us illustrate how Eva analyzes the toy example of Figure 4a, which gives the body of function main. It expects the user to type a character (line 3). In the majority of cases, the else branch is activated and the program executes without errors. But if the user types '*', the program executes the branch and tries to divide by 0 on line 14. We use line numbers to refer to program points l.

At line 2, the sets of values computed by Eva (shown in comments on line 2) for the four variables contain only the special "Uninitialized" value. After the assignments of line 4 (resp., 7), the new domains of x and y are shown on line 5 (resp. 8), the others being unchanged. On line 10, the domains coming from both branches are merged. Therefore, the computed over-approximated set of values for `sum` on line 12 is $D_{sum}^{12} = \{0, 1, 2\}$, even if the value 1 is not possible in practice. Since $0 \in D_{sum}^{12}$, the assertion on line 13 cannot be proved, and Eva reports a potential division by 0. This is a *true alarm*: the division by 0 can happen, and Eva detects it.

Eva can detect other runtime errors similarly. For instance, if the assignment of y is removed on line 4, Eva computes $D_y^5 = \{\text{Uninit}\}$ and $D_y^{10} = \{\text{Uninit}, 1\}$, and reports an alarm for reading an uninitialized variable on line 11.

Approximations often lead to false alarms: correct code can also be flagged as a potential error. It can be seen in the example of Figure 4b, where the computed over-approximated set of values for sum on line 12 is again $D_{sum}^{12} = \{0, 1, 2\}$, while only value 1 occurs in practice. Based on this over-approximated set, Eva cannot prove the assertion on line 13 and reports a potential division by 0 while it can never happen; this is a false alarm. To avoid it, the user can use trace partitioning; that is, make the analysis consider both paths separately to glean a more precise analysis. Eva will continue the analysis of both branches without merging their values on line 10: in both cases (with $D_x^{10} = \{0\}, D_y^{10} = \{1\}$, and $D_x^{10} = \{1\}, = \{0\}$) Eva will compute $D_{sum}^{12} = \{1\}$ and prove the absence of the error.

To limit the burden of false alarms while maintaining a reasonable analysis time, a balance must be reached between precision and efficiency. Typically, Eva is used in an iterative process, where the analyst configures the abstractions and partitioning and uses the result of one analysis to finely tweak the next one. To make complex settings easily accessible for non-expert users, Eva provides a meta-option -eva-precision `N`, with `N` between 0 and 11, which conveniently adjusts a dozen underlying options (including trace partitioning[30]). Any $N \geq 1$ avoids the false alarm for Figure 4b.

Eva provides various means of expressing abstractions (called *abstract domains*) that can be enabled and tuned on a case-by-case basis. The default abstract domain represents integer values as small discrete sets or intervals with a linear congruence information, floating-point values as intervals following the IEEE 754 standard, and pointers as possible offsets for each potential base address. It accurately represents arrays, structures, and unions. Various additional domains (such as *gauges*,[39] *numerors*,[25] numerical domains provided by Apron[c]) bring more expressiveness but slow down the analysis.

Studying the results. Eva's main output is an exhaustive list of potential undefined behaviors, or *alarms*, expressed as ACSL assertions. Each alarm should be reviewed to determine if it reveals a real bug or is a false alarm caused by the analysis approximations. False alarms might be disproved by other Frama-C plug-ins. It is possible to inspect (see Figure 5), at each program point and for each call stack, the values computed for each variable and expression.

Eva is tightly integrated with other tools of the platform, providing them with detailed information about its results. These results are used by many other plug-ins. Notably, Studia highlights all statements reading or writing a given memory location, allowing the user to jump between the sink of a bug (where it can be observed) and its source (the actual culprit). InOut computes the memory zones read and written by a function, summarizing its dependen-

[c] http://apron.cri.ensmp.fr/library/

Figure 4. Eva illustrated on two toy examples, (a) and (b).

```
1  int x, y, sum, res;
2  // $D_x^2 = D_y^2 = D_{sum}^2 = D_{res}^2 = \{Uninit\}$
3  if(getchar()=='*'){
4    x = 0; y = 0;
5  // $D_x^5 = \{0\}$, $D_y^5 = \{0\}$
6  }else{
7    x = 1; y = 1;
8  // $D_x^8 = \{1\}$, $D_y^8 = \{1\}$
9  }
10 // $D_x^{10} = \{0, 1\}$, $D_y^{10} = \{0, 1\}$
11 sum = x + y;
12 // $D_{sum}^{12} = \{0, 1, 2\}$
13 //@ assert sum ≠ 0;
14 res = 10/sum;
```

(a)

```
1  int x, y, sum, res;
2  // $D_x^2 = D_y^2 = D_{sum}^2 = D_{res}^2 = \{Uninit\}$
3  if(getchar()=='*'){
4    x = 0; y = 1;
5  // $D_x^5 = \{0\}$, $D_y^5 = \{1\}$
6  }else{
7    x = 1; y = 0;
8  // $D_x^8 = \{1\}$, $D_y^8 = \{0\}$
9  }
10 // $D_x^{10} = \{0, 1\}$, $D_y^{10} = \{0, 1\}$
11 sum = x + y;
12 // $D_{sum}^{12} = \{0, 1, 2\}$
13 //@ assert sum ≠ 0;
14 res = 10/sum;
```

(b)

cies. Finally, Metrics estimates the analysis code coverage and reports the statements proven unreachable by Eva.

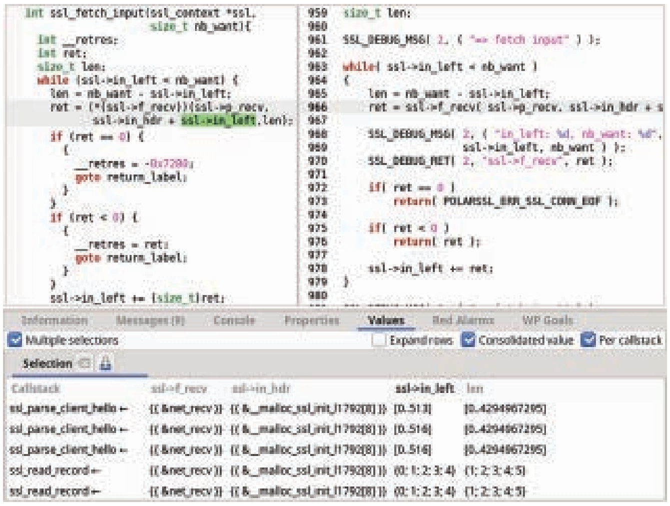
Usage. Eva handles the subset of C99 commonly used in embedded software. Dynamic allocation is supported, but often leads to imprecise results. The analysis is fully context-sensitive: function calls are inlined, and recursive functions are not supported. Eva has been highly optimized for years to achieve scalability on large programs and has already been successfully applied to verify safety-critical codes, especially in the nuclear industry.[33]

Proving functional properties with WP. *Deductive verification* aims at proving that functional properties of a program hold in all cases. It is usually performed in a *modular* way, function by function, where the caller's proof can rely on the callee's contract, proved separately. The WP plug-in is a modern and effective implementation of this approach for C and ACSL.

Let us illustrate this approach on the code of Figure 6a (say, giving the body of function main) that calls the function of Figure 2. After line 2 of Figure 6a, pointer `q` refers to `x`, thus `x` and `*q` are aliases. On this code, WP deduces the first assertion from the value –42 of `*q` before the call and the postcondition of `pabs` (lines 4–5 in Figure 2). Since other variables cannot be modified by the call of `pabs` (line 3 in Figure 2), WP also proves the second assertion of Figure 6a.

However, a call to a function guarantees to ensure its postcondition after the call only if its precondition is respected before the call. Thus, WP must check that the precondition of `pabs` (lines 1–2 in Figure 2) is respected before the call: here, indeed, pointer `q` is valid and the pointed value is not `INT_MIN`. The precondition cannot be proved for the code of Figure 6b, where the pointed value is `INT_MIN`. Its proof also fails for the code of Figure 6c, where `q` refers to the first cell of an array of two integers; hence `q+2` is invalid: dereferencing it would be an out-of-bounds access (that is, an undefined behavior).

Figure 5. Frama-C graphical interface allows any C expression to be inspected for its possible runtime values, as computed by Eva. Pointers, structured and scalar values, are expressed in a concise but precise notation. Each callstack is separated, with filtering and grouping capabilities.



```
int ssl_fetch_input(ssl_context *ssl,
                    size_t nb_want){
  int __retres;
  int ret;
  size_t len;
  while (ssl->in_left < nb_want) {
    len = nb_want - ssl->in_left;
    ret = (*(ssl->f_recv))(ssl->p_recv,
             ssl->in_hdr + ssl->in_left,len);
    if (ret == 0) {
      {
        __retres = -0x7280;
        goto return_label;
      }
    }
    if (ret < 0) {
      {
        __retres = ret;
        goto return_label;
      }
    }
    ssl->in_left += (size_t)ret;
```

```
size_t len;

SSL_DEBUG_MSG( 2, ( "=> fetch input" ) );

while( ssl->in_left < nb_want )
{
    len = nb_want - ssl->in_left;
    ret = ssl->f_recv( ssl->p_recv, ssl->in_hdr + s

    SSL_DEBUG_MSG( 2, ( "in_left: %d, nb_want: %d",
                   ssl->in_left, nb_want ) );
    SSL_DEBUG_RET( 2, "ssl->f_recv", ret );

    if( ret == 0 )
        return( POLARSSL_ERR_SSL_CONN_EOF );

    if( ret < 0 )
        return( ret );

    ssl->in_left += ret;
}
```

☑ Multiple selections □ Expand rows ☑ Consolidated value ☑ Per callstack

Callstack	ssl->f_recv	ssl->in_hdr	ssl->in_left	len
ssl_parse_client_hello ←	{{ &net_recv }}	{{ &__malloc_ssl_init_l1792[0] }}	[0..513]	[0..4294967295]
ssl_parse_client_hello ←	{{ &net_recv }}	{{ &__malloc_ssl_init_l1792[8] }}	[0..514]	[0..4294967295]
ssl_parse_client_hello ←	{{ &net_recv }}	{{ &__malloc_ssl_init_l1792[8] }}	[0..514]	[0..4294967295]
ssl_read_record ←	{{ &net_recv }}	{{ &__malloc_ssl_init_l1792[8] }}	{0; 1; 2; 3; 4}	{1; 2; 3; 4; 5}
ssl_read_record ←	{{ &net_recv }}	{{ &__malloc_ssl_init_l1792[8] }}	{0; 1; 2; 3; 4}	{1; 2; 3; 4; 5}

Figure 6. WP illustrated on toy examples (a), (b), and (c).

```
1  int x=-42, y=36;
2  int *q=&x;
3  // Pre_pabs holds
4  pabs(q);
5  //@ assert x==42;
6  //@ assert y==36;
```

(a)

```
1  int x=INT_MIN;
2  int *q=&x;
3  // Value of *q
4  // is INT_MIN.
5  // Pre_pabs fails
6  pabs(q);
```

(b)

```
1  int a[2]={-42,0};
2  int *q=&a[0];
3  // Pointer q+2
4  // is invalid.
5  // Pre_pabs fails
6  pabs(q+2);
```

(c)

In the modular approach, the function contract of the callee must be proved separately. For the code of Figure 2, WP successfully proves that the implementation of `pabs` respects its contract.

Deductive verification for programs with loops usually relies on loop contracts that must be specified by the user. To illustrate it on a toy example, consider the code of Figure 3. For the loop contract, WP must verify that the loop invariant is indeed true before the loop and is preserved by each new loop iteration, and that the loop frame rule is indeed true. Thanks to the loop invariant, at line 7, WP knows that $0 \leq i \leq 10$, $i + j = 10$, and since the execution exited the loop, $i \geq 10$. From these conditions, it deduces that $i = 10$ and therefore $j = 0$, that proves the assertion on line 7. The assertion on line 8 is deduced from the frame rule (line 4) since the value of k cannot be changed by the loop.

To perform deductive verification, WP relies on Hoare logic and weakest precondition calculus. At a high level, WP compiles C code and ACSL contracts into mathematical theorems (called *verification conditions* and expressed as first-order logic formulas) that provide sufficient conditions to entail the validity of the expected functional properties. These theorems use various mathematical theories (including integer and real arithmetic, anonymous functions, arrays, and records). They are then sent to automated theorem provers (or SMT solvers, such as Alt-Ergo, Z3, or CVC4) to be checked for validity. Alternatively, one can also use a proof assistant like Coq.

Naive implementations of weakest precondition calculus are known to have exponential costs and cannot be used on complex programs. Moreover, modeling the semantics of C memory access with aliasing and low-level encoding of data is known to be a challenge for automated reasoning. WP has been developed since 2008 with an industrial target in mind and benefits from well-known modern techniques to make it efficient.

WP implements a generic, backward calculus engine to produce verification conditions by weakest precondition calculus.[28] It is parameterized by a memory model, defining a specific representation of memory locations in the resulting verification conditions. WP features various memory models which combine known techniques[21] to propose different balancing between efficiency and expressiveness, and some heuristics to select which model(s) to apply on a given program.

WP offers several backends for discharging generated verification conditions with automated SMT solvers and proof assistants, either natively or via the Why3[22] platform. The complexity of the generated verification conditions is dramatically reduced by Qed,[13] a generic and extensible simplification engine of WP, helping to discharge some corner cases of theories that are still issues for mainstream SMT solvers. Finally, WP features an extensible proof tactic engine to interactively split complex proofs into smaller ones, possibly executing custom decision procedures.

Altogether, these features make WP an efficient implementation of deductive verification to prove functional properties of C/ACSL programs. A recent industrial use case in avionics[9] reports that 98.5% of the 3,315 C functions were proved by WP, where only 2.3% functions required the interactive termination of some proofs.

No single technique will succeed in assessing all properties a user can be interested in. Thus, the platform fosters collaborations between various techniques by any test case.

Checking properties at runtime with E-ACSL. Runtime assertion checking is the process of verifying specifications (historically, assertions) at runtime, that is, when the program is being executed. It was popularized by the programming language Eiffel in the late 1980s to support defensive programming. At the turn of the millennium, this approach was adopted by dedicated, formal specification languages for mainstream programming languages, such as JML for Java or Spec# for C#.

In the context of C and Frama-C, ACSL would be the language of choice for runtime assertion checking. However, being primarily designed for deductive verification, it needed adjustments for runtime checking. In addition, verifying expressive properties at runtime for a language like C in a sound and efficient way is challenging and requires original solutions.

Specification language adjustments. As explained previously, ACSL is based on mathematical logic. In particular, it contains several constructs that have no computational meaning, such as lemmas and axioms or unbounded quantifications. Therefore, they were removed from the executable subset of ACSL dedicated to runtime assertion checking: the E-ACSL specification language.[d]

Another important issue of ACSL, with respect to runtime checking, is its logic-based semantics, which assigns a (possibly unspecified) value to each construct. For instance, the predicate $0/0 \equiv 0/0$ is necessarily true in ACSL by reflexivity of equality. This semantics helps formal reasoning made by the WP plug-in and associated provers. However, it is problematic at runtime, since terms such as 0/0 cannot be safely executed. Consequently, E-ACSL considers that the semantics of such terms is actually undefined (relying on Chalin's strong validity principle[11] and three-valued logic). Undefined terms and predicates must never be executed.

Compiling formal properties into executable code. Compiling E-ACSL annotations into C code is the purpose of the E-ACSL plug-in[38] of Frama-C. The instrumented code it produces checks the annotations at runtime and reports failures. For instance, using the E-ACSL plug-in to check the code of Figure 6a at runtime confirms the annotations (including the assertions, the precondition, and postcondition of `pabs`) are verified, while for Figure 6b,c, the failing preconditions are detected and reported to the user.

At a first glance, the compilation process may look quite easy. For instance, the E-ACSL assertion `/*@ assert z ≠ 0;*/` is compiled to the C assertion `assert(z ≠ 0);`. However, in general it is not always so simple to generate code that is both sound and efficient, as shown below on two illustrative cases.

Arithmetic. For the E-ACSL assertion `/*@ assert x+1 ≤ INT_MAX;*/`, it would be unsound to generate the C code `assert(x+1 ≤ INT_MAX);` since at runtime `x+1` might overflow, while in the ACSL specification, as we explained above, it is computed over (unbounded) mathematical integers. Consequently, E-ACSL generates specific

[d] http://frama-c.com/download/e-acsl/e-acsl.pdf

code[e] to precisely perform the computations and to remain sound. To remain efficient, it still generates more efficient machine arithmetic-based code when it is sound to do so. For instance, assuming that the type of *x* is int on a standard 64-bit architecture, the previous assertion is compiled to `assert((long)x+1L ≤ (long) INT_MAX);` to execute the addition and comparison without overflow over the larger C type `long`.

Memory properties. Memory properties, such as `\valid(p)`, are an important feature of the specification language. To soundly and efficiently evaluate such properties, memory-related operations (allocations, deallocations, and assignments) in the original code that are relevant for the memory properties of interest are recorded in a dedicated data structure.

Generating test cases with PathCrawler. Another dynamic analysis plug-in of Frama-C is PathCrawler.[40] Given a C program and a specific function in it, PathCrawler generates unit test cases for this function. Basically, it explores a subset of program paths and tries to generate test inputs for each of them. PathCrawler follows the so-called *concolic* test-generation technique—also called *Dynamic Symbolic Execution*, since it combines symbolic execution of the program with a usual (*concrete*, that is, non-symbolic) execution of the compiled code.

Symbolic execution represents the execution of a program path symbolically, with undetermined values of program inputs. It relies on the path predicate defining the values of the input variables that activate the chosen path. PathCrawler relies on the Colibri constraint solver, also developed at CEA List, to find a set of concrete values satisfying the path predicate, that is, test inputs for the path. A concrete execution of the generated test on an instrumented version of the program is used to confirm the executed path and to optimize the test-generation process.

To ensure the test inputs are realistic and avoid detecting bugs which would never arise in legitimate function calls, the user can provide a precondition limiting the admissible input values. If the user also provides an *oracle* function that compares the outputs produced by the test with the expected behavior, then PathCrawler will automatically report a "Pass" or "Fail" verdict for each test.

Since 2009, PathCrawler has an online version[f] (see Figure 7) that allows the user to provide a C file (or choose one of the available examples), generate test cases, and explore the results.

Tell Frama-C What You Want to Verify

Together with the expressive power of ACSL specifications, the core analyzers presented in the previous section allow the verification of a very large class of properties about C programs. However, the bare ACSL language sometimes makes it difficult to express other kinds of properties. In that case, various specialized plug-ins exist to ease the task of writing the formal specification of a property of interest.

In many cases, such a plug-in offers a dedicated domain-specific language (DSL) for writing the property. The plug-in operates by instrumenting the code under analysis with additional ACSL annotations and/or C instructions so that the verification of standard ACSL annotations on the instrumented code with the core analyzers implies that the original DSL formulas hold on to the original code.

Verify sequences of events: Aoraï and CaFE. It is often necessary to verify that a set of events during a program execution follows a particular order, for example:

A call to function `send_private_data()` must always be preceded by a call to function `authenticate()` returning ∅, without a call to function `logout()` in-between.

Such properties (often expressed in temporal logic[12]) can be verified for any given execution using an automaton. In our example, it consists of three states encoding the current status of the execution: user non-authenticated (initial state), user authenticated (and not yet logged out), or error (that is, private data sent without being authenticated). The transitions between states naturally follow the observed function calls, except that the error state cannot be left. The first two states are accepting (that is, the property is respected as long as the execution ends in one of them), while the last one means the property fails for the given execution.

Two Frama-C plug-ins are dedicated to such properties. Aoraï[26] simply adds C variables representing the states, together with the appropriate transition functions and ACSL annotations ensuring we end up in an accepting state. Checking the validity of these annotations is then left to one of the main analysis plug-ins described in the previous section. CaFE, a more recent plug-in, can handle additional properties, including nested function calls. CaFE is based on a refined version of classical temporal logic, CaRet,[2] and relies on model-checking techniques.[12]

Verify relational properties: RPP. Contrary to an ACSL contract, which specifies what is supposed to happen during a single call to the corresponding function, relational properties examine the relations that may exist between several executions of either the same or different functions. An interesting example of this class of properties is non-interference: given a partition of the variables into public and private ones, one wants to ensure that any two executions starting in states where public variables have the same values always end up in states where public variables have the same values. In other words, the public result should not depend upon the values of private variables.

Figure 8a illustrates a function `noleak` that respects this property: the public variable `pub` does not depend on the secret variable `sec`. This property is not true for function `leak`, where `pub` depends on `sec`.

RPP[8] is a Frama-C plug-in that offers an extension of ACSL to formally specify relational properties (involving any number of executions of any number of functions). RPP then uses a form of self-composition[6] to generate a wrapper function (composing the executions of the functions involved in the relational property) with an ACSL contract, such that its proof implies the relational property for the original code. For instance, the wrapper of Figure 8b simulates two executions of `leak` with equal public values (line 2), but this equality after these executions (line 3)—the non-interference—cannot be proved: the pub-

[e] Based on GNU Multiple Precision Arithmetic Library: https://gmplib.org/

[f] http://pathcrawler-online.com

lic result depends on a secret variable.

An important benefit of RPP's transformation is that it also allows the use of a proven relational property as a hypothesis in subsequent proofs, following the modularity of the standard deductive verification approach.

Enforce global properties: MetACSL. It is often the case that one wants to enforce a given property across the whole program. For instance, we may associate a confidentiality level with each memory location and check that a read access is never performed from a location with a higher level than that of the current user, or that a write is never performed into a lower-level location. While these kinds of properties could, in theory, be expressed with standard ACSL annotations, they would spread everywhere in the program. In practice, it can be difficult to write them all by hand without making a mistake and to convince ourselves that the set of annotations is indeed complete.

The recently started MetACSL plug-in[37] seeks to alleviate this issue by automatically generating these ACSL annotations from a single, higher-level property expressed in a small DSL, extending ACSL to indicate the contexts in which the property must hold. It has been tested over various examples to establish security properties (confidentiality and integrity) and is currently being assessed over more realistic case studies.

Prove concurrent programs: Conc2Seq. While most of its plug-ins focus on sequential program analysis, Frama-C also offers Conc2Seq, an experimental plug-in for deductive verification of concurrent programs.[7] Similarly to the CSec approach,[g] it performs a dedicated code transformation of a given concurrent program into a sequential one. This simulates concurrent executions of the code in several threads by interleaving the executions of indivisible (atomic) blocks in various ways, defined non-deterministically.

Conc2Seq also automatically transforms specifications of the initial program into specifications for the resulting program. The variables of various threads are represented by arrays in the simulating program, so that the user can add guiding annotations relating these variables between them to help the proof. Thanks to this transformation, the WP plug-in can be used to verify the resulting sequential program. If the proof of the annotations for it is successful, the initial concurrent program respects its specification.

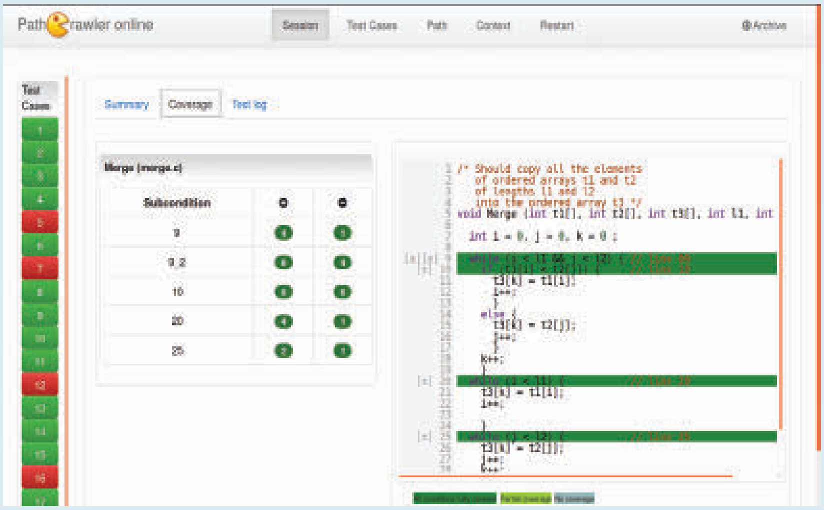
Specify test objectives: LAnnotate. Test objectives offer another example of specific annotations that can be added for analysis using Frama-C core plug-ins. Various structural test coverage criteria (for example, functions, statements, decisions, or branches, conditions, conditions-decisions) can be treated in a unified way provided that the corresponding test objectives are expressed in the code in the generic form of elementary coverage targets. Such a coverage target, also called a "label,"[4] is basically a predicate inserted in a particular location.

A label is covered by a test when the execution of the test reaches this location and satisfies the predicate. For a given test-coverage criterion, the LAnnotate plug-in[4] inserts the corresponding labels and other plug-ins that can be used to reason about them. In particular, PathCrawler supports the label-coverage criterion (and therefore, all coverage criteria that can be expressed using labels) and offers an efficient test generation for labels. Other usages of labels are examined next.

Go Beyond Raw Analyses Results

The previous section presented several analyses that apply core plug-ins after a relatively lightweight adaptation (often via instrumentation) of properties of interest into properties they can directly handle. For more complex analysis problems, this is not sufficient. The target properties can require an advanced code, specification transformation, or even a dedicated reasoning. Their analysis can still rely on some of the core analyzers but has to extend or adapt them in a more significant way. We present a few examples here.

Figure 7. Results of a test-generation session with PathCrawler online illustrating condition coverage of the generated test cases. Failed test cases are shown in red. For each test case, its inputs, outputs, and the activated path can be inspected. Any gaps in the coverage of the function are also explained.



Merge (merge.c)

Subcondition	○	●
9	●	●
9.2	●	●
10	●	●
20	●	●
25	●	●

```
/* Should copy all the elements
   of ordered arrays t1 and t2
   of lengths l1 and l2
   into the ordered array t3 */
void Merge (int t1[], int t2[], int t3[], int l1, int

  int i = 0, j = 0, k = 0 ;

  while (i < l1 && j < l2) {     // line 09
    if (t1[i] < t2[j]) {         // line 10
      t3[k] = t1[i];
      i++;
      }
    else {
      t3[k] = t2[j];
      j++;
      }
    k++;
    }
  while (i < l1) {               // line 20
    t3[k] = t1[i];
    i++;

    }
  while (j < l2) {               // line 25
    t3[k] = t2[j];
    j++;
    k++;
```

Figure 8. (a) A C code, and (b) RPP transformation for leak.

```
int sec, pub;
void noleak(){
 pub=pub+10;
 sec=sec+pub; }
void leak(){
 pub=pub+sec; }
```

(a)

```
int sec1, pub1, sec2, pub2;
/*@ requires pub1==pub2;
    ensures  pub1==pub2; */
void wrapper_leak(){
 pub1=pub1+sec1; /* 1st call */
 pub2=pub2+sec2; /* 2nd call */ }
```

(b)

[g] http://www.southampton.ac.uk/~gp1y10/cseq/cseq.html

Counter-examples for unproven annotations: StaDy. Manual analysis of proof failures during deductive program verification can be a very complex and time-consuming task. Such failures can be due to an error in the code or in the specification itself, a missing or weak specification for a called function or a loop, lack of time, or the incapacity of the prover to finish a particular proof. Using a combination of deductive verification (with WP) and test generation (with PathCrawler), the StaDy plug-in[35] helps to classify proof failures into several categories and provides a counter-example illustrating the issue.

The translation of ACSL annotations (preconditions, postconditions, and so on) into their counterparts supported by test generation is not straightforward. For example, to support unbounded integers in ACSL annotations during both concrete and symbolic execution, operations with unbounded integers are translated in two different ways: directly into unbounded integers supported by the constraint solver for symbolic execution and using a dedicated library for execution of unbounded integers for a concrete execution.

While StaDy was mainly designed for use with WP, it can also be applied to alarms reported by Eva. Such alarms being reported as unproven assertions, StaDy can be applied to generate counterexamples for some of them (thus showing that they are not false alarms) and facilitate the analysis of alarms by the verification engineer. This is another illustration of the benefits of sharing the same specification language between different analyzers.

Infeasible test objectives: LUncov. Previously, we illustrated how generic test objectives—or labels—allow PathCrawler to support test-case generation for various test coverage criteria. An important issue in testing relates to infeasible (that is, uncoverable) test objectives that cannot be covered by any test case. Infeasible test objectives lead to an imprecise computation of coverage for a given test suite and a waste of resources for trying to cover them. Detection of infeasible test objectives—which is in general undecidable—is thus an important task in testing.

An efficient approach to identify infeasible test objectives is to use static analysis. This is the purpose of the LUncov plug-in[4]. It translates a label with predicate p into an assertion with the negated predicate $\neg p$ at the same location. The label is uncoverable if and only if the resulting assertion is always true. LUncov implements various analysis techniques relying on value analysis using Eva, and weakest precondition calculus using WP. In particular, it provides an advanced combination of both tools, where Eva is used to compute the domains of program variables and then shares this information with WP to make it more precise.

Frama-C is intensively used for teaching. Indeed, it became difficult to keep track of all universities where the toolset is used in various program analysis or verification courses.

Program simplification: Slicing. *Slicing* is a program transformation technique that takes as input a program and a so-called *slicing criterion* (for example, to preserve the value of a given variable at a given program point) and outputs a simplified C program that preserves the property defined by the slicing criterion. The pieces of code necessary to ensure the preservation property (or for a correct compilation) of the resulting program are kept, while all other, irrelevant instructions are removed. Slicing helps the end-user to focus on a particular point of interest. It also facilitates other analyses by reducing the size of the code they must deal with.

The Frama-C slicing tool proposes numerous slicing criteria including preserving read and written memory locations at particular program points, function calls, return values, ACSL annotations or statements. It soundly relies on Eva to compute aliasing and dependency information. Therefore, it may over-approximate its results by keeping pieces of code that are actually not relevant for the selected criterion. However, it never removes anything relevant.

Information flow: SecureFlow. Information flow properties denote properties of the dependencies between the outputs and the inputs of the program. The most common example, presented above, is non-interference. It expresses the absence of information leak. The SecureFlow plug-in[3] lets the user annotate each declaration with a public or private attribute and uses a dataflow analysis to verify the absence of information leak. It also relies on Eva's results for determining which locations (hence, with which confidentiality level) pointers might refer to.

More Than a Toolset: An Ecosystem

Frama-C community. Since its initial release in May 2008, Frama-C has built an active community of users and plug-in developers. Its open source license (LGPL 2.1) played an important role in this development, facilitating its integration into many Linux distributions (the oldest package, from Debian, dates back to 2009 and Frama-C 3.0 Lithium), and into the main repository of the opam package manager that handles software written in OCaml, as is the case with Frama-C. As of July 2020, opam reports around 200 monthly downloads (via opam) of the latest release, Frama-C 21.1 Scandium, which was released in June 2020.

Naturally, these public releases are accompanied with various communication channels,[h] including a mailing list, a bug tracker, and a dedicated StackOverflow tag. Frama-C's blog[i] is also a good way to inform users about what is going on in the platform.

An important part of Frama-C development is funded through collaborative projects, most of which are supported by the French government and the European Union. Apart from CEA itself, these projects usually gather a mix of academic and industrial partners to explore new research directions while keeping sure that they are relevant to real-world problems. Among these projects, we can mention the French RNTL project CAT and its successor U3CAT,[j] funded by ANR, both of which have been fundamental for building the grounding blocks of the platform. Later on, European projects Stance and Vessedia help broaden Frama-C's target properties to cybersecurity.

Teaching with Frama-C. Frama-C is intensively used for teaching. Indeed, it became difficult to keep track of all universities where the toolset is used in various program analysis or verification courses. In France, where the platform was born and is developed, there are dozens of departments relying on Frama-C for teaching every year. Just a few examples include Ecole Polytechnique, CentraleSupélec, École Normale Supérieure, ENSIIE, almost all universities in and around Paris, as well as in Besançon, Bordeaux, Bourges, Grenoble, Lille, Lyon, Orléans, Rennes, Toulouse, and many others. Frama-C is also increasingly used in other countries, including Austria, Brazil, China, Germany, Portugal, Russia, U.K., and the U.S. Among the analyzers of the open source distribution, Eva, WP, and E-ACSL are the most popular plug-ins for teaching. PathCrawler is also actively used for teaching thanks to its online version, PathCrawler-online, allowing the user to explore advanced test-generation results. Finally, Frama-C has often been used for training in industrial companies and for tutorials on program verification at premier international conferences, such as ASE, FM, iFM, ISSRE, POPL, SAC, TAP, and QSIC.

Collaborations and industrial applications. Long-time partnerships began with Frama-C's precursor, Caveat, which was developed in the 1990s in close collaboration with the teams at Airbus and leveraged automated reasoning capabilities from Inria's Alt-Ergo solver. As Caveat went into industrial production, the development around Frama-C continued to nourish these collaborations and engaged them in assisting with design decisions. Notably, this took the form of a domain-specific language for low-level specifications that compiles into ACSL for deductive verification with WP, or into a system similar to E-ACSL for runtime verification. This system, NWOW,[9] has been deployed at large scale for the development of onboard critical software, and will be extended to other applications.

Other partnerships started in the mid-2000s, with Électricité de France (EDF) and Areva for energy production systems. In particular, EDF reported[33] that Frama-C's Value Analysis plug-in (predecessor of Eva) improved the analysis of a 39kLoC nuclear power plant shutdown system, allowing the demonstration of the absence of intrinsic runtime errors. After some experimentation with different tools, Frama-C was chosen to analyze the code. Today, an ongoing collaboration with EDF focuses on the analysis of larger code bases. R&D efforts between EDF, Framatome, and CEA study further usage of Frama-C for other safety-critical software.

Frama-C has also been used for verifying software in other industrial domains, notably by Fraunhofer FOKUS[36] and Mitsubishi for rail, and Brazil's TIA for space applications.[18] The 2010s saw a broadening of this base, and an extension from safety-critical software into cybersecurity. The capabilities of Frama-C were used by NASA in air traffic management,[24] SRI International in gamified cybersecurity,[20] Bureau Veritas in marine and offshore,[27] and Thales and ANSSI in communication.[19]

Test generation with PathCrawler was recently evaluated by MERCE (Mitsubishi Electric R&D Centre Europe). After developing additional tooling around PathCrawler, MERCE evaluated automatic test generation over industrial code of about 80,000 lines. In this experiment, 86% of functions were successfully covered in eight hours. MERCE estimated that automatic test generation with PathCrawler could bring an effective benefit factor of more than 230 for test input generation in the company. Those very good results are encouraging for an adoption of the technology in the business units.[5]

Beyond applications, the extensibility of the platform also allowed tool developers to abstract from the groundwork of code parsing and data structure design, and to focus on new types of verification. Early on, Inria experimented with deductive verification in the Jessie plug-in, later on extended and adapted by ISPRAS in AstraVer.[29] Adelard investigated lightweight concurrency, while teams at Atos implemented dataflow conformity capabilities[15] and prototyped IDE integrations. The field of cybersecurity also proved fertile in academic developments, giving rise to the Stac plug-in from Verimag[10] or the Celia plug-in from Université Paris Diderot.[17] Finally, the mid-2010s modernization brought about with the Eva plug-in allowed for another level of extensibility, at the level of its abstract domains. This was quickly adopted to interface with developments in this field from Verimag, including the Apron domain library and the VPL verified polyhedron library.[23]

Similarly, in the context of European projects Stance[k] (FP7) and Vessedia[l]

[h] https://frama-c.com/support.html

[i] https://blog.frama-c.com

[j] https://frama-c.com/u3cat.html

[k] https://cordis.europa.eu/project/rcn/105816/brief/en

[l] https://www.vessedia.eu/

(H2020), Search Lab developed Frama-C plug-ins dedicated to generating counter-examples in the spirit of StaDy but based on external test-case generators, namely Search Lab's own tool Flinder[m] and later the AFL fuzzer.[n] In these projects, Dassault Aviation also designed a methodology based on Eva, E-ACSL, and two home-made plug-ins to detect security vulnerabilities and deploy runtime countermeasures when necessary.[34] It has been experimented on a few modules of Apache.

Conclusion

Since its first public release more than 13 years ago, the Frama-C framework has demonstrated its ability to successfully address very diverse verification tasks. One of the main factors of this success is undoubtedly the key design idea of a modular analysis platform, where developing a specialized plug-in and having it communicate with others should be as easy as possible. Another important aspect is the fact that the development of Frama-C has been fueled by collaborative projects that strive to maintain a balance between exploring new research directions and targeting existing industrial code. This holds true to this day, with lines of research toward new programming languages (C++, Rust), cybersecurity and privacy properties, verifying AI-based applications or using AI in verification among others. We hope the readers will try out Frama-C[o] and will find it useful for their verification activities.

[m] https://www.flinder.hu

[n] http://lcamtuf.coredump.cx/afl/

[o] see http://www.frama-c.com/download.html

References

1. Alberti, M. and Signoles, J. Context generation from formal specifications for C analysis tools. In *Proc. of the 2017 Conf. on Logic-based Program Synthesis and Transformation.*
2. Alur, R., Etessami, K., and Madhusudan, P. A temporal logic of nested calls and returns. In *Proc. of the 2004 Conf. on Tools and Algorithms for the Construction and Analysis of Systems.*
3. Barany, G. and Signoles, J. Hybrid information flow analysis for real-world C code. In *Proc. of the 2017 Conf. on Tests and Proofs.*
4. Bardin, S., Chebaro, O., Delahaye, M., and Kosmatov, N. An All-in-One Toolkit for Automated White-Box Testing. In *Proc. of the 2014 Conf. on Tests and Proofs.*
5. Bardin, S., Kosmatov, N., Marre, B., Mentré, D., and Williams, N. Test case generation with PathCrawler/LTest: How to automate an industrial testing process. In *Proc. of the 2018 Conf. on Leveraging Applications of Formal Methods, Verification and Validation.*
6. Barthe, G., D'Argenio, P., and Rezk, T. Secure information flow by self-composition. *Mathematical Structures in Computer Science 6* (2011).
7. Blanchard, A., Kosmatov, N., Lemerre, M., and Loulergue, F. Conc2Seq: A Frama-C Plugin for Verification of Parallel Compositions of C Programs. In *Proc. of the 2016 Conf. on Source Code Analysis and Manipulation.*
8. Blatter, L., Kosmatov, N., Gall, P., and Prevosto, V. RPP: Automatic proof of relational properties by self-composition. In *Proc. of the 2017 Conf. on Tools and Algorithms for the Construction and Analysis of Systems.*
9. Brahmi, A., Carolus, M., Delmas, D., Essoussi, M., Lacabanne, P., Lamiel, V., Randimbivololona, F., and Souyris, J. Industrial use of a safe and efficient formal method based software engineering process in avionics. In *Proc. of the 2020 Conf. on Embedded Real Time Softw. and Systems.*
10. Ceara, D., Mounier, L., and Potet, M. Taint dependency Ssquences: A characterization of insecure execution paths based on input-sensitive cause sequences. In *Proc. of the 2010 Int. Conf. on Softw. Testing, Verification and Validation.*
11. Chalin, P. A sound assertion semantics for the dependable systems evolution verifying compiler. In *Proc. of the 2007 Int. Conf. on Softw. Engineering.*
12. Clarke, E., Emerson, E., and Sistla, A. Automatic verification of finite-state concurrent systems using temporal logic specifications. *Trans. Programming Languages and Systems* (1986).
13. Correnson, L. Computing what remains to be proved. *In Proc. of the 2014 Conf. on NASA Formal Methods.*
14. Correnson, L. and Signoles, J. Combining analyses for C program verification. *Int. Workshop. on Formal Methods for Industrial Critical Systems (2012)..*
15. Cuoq, P., Delmas, D., Duprat, S., and Lamiel, V. Fan-C, a Frama-C plug-in for data flow verification. In *Proc. of the 2012 Conf. on Embedded Real Time Softw. and Systems.*
16. de Oliveira, S., Bensalem, S., and Prevosto, V. Polynomial invariants by linear algebra. In *Proc. of the 2016 Conf. on Automated Technology for Verification and Analysis.*
17. Dragoi, C., Enea, C., and Sighireanu, M. Local shape analysis for overlaid data structures. In *Proc. of the 2013 Int. Symp. on Static Analysis.*
18. e Silva, R., Arai, N., Burgareli, L., de Oliveira, J., and Pinto, J.
Formal verification with Frama-C: A case study in the space software domain. *Trans. Reliability* (2016).
19. Ebalard, A., Mouy, P., and Benadjila, R. Journey to a RTEfree X.509 parser. In *Proc. of the 2019 Symp. sur la Sécurité des Technologies de l'information et des Communications.*
20. Fava, D., Signoles, J., Lemerre, M., Schäf, M., and Tiwari, A. Gamifying program analysis. In *Proc. of the 2015 Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning.*
21. Filliâtre, J. and Marché, C. Multi-prover verification of C programs. In *Proc. of the 2004 Int. Conf. on Formal Methods and Softw. Engineering.*
22. Filliâtre, J. and Paskevich, A. Why3—Where programs meet provers. In *Proc. of the 2013 European Symp. on Programming.*
23. Fouilhé, A., Monniaux, D., and Périn, M. Efficient generation of correctness certificates for the abstract domain of polyhedra. In *Proc. of the 2013 Int. Symp. on Static Analysis.*
24. Goodloe, A., Muñoz, C., Kirchner, F., and Correnson, L. Verification of numerical programs: From real numbers to floating point numbers. In *Proc. of the 2013 Conf. on NASA Formal Methods.*
25. Jacquemin, M., Putot, S., and Védrine, F. A reduced product of absolute and relative error bounds for floating-point analysis. In *Proc. of 2018 Int. Symp. on Static Analysis.*
26. Kirchner, F., Kosmatov, N., Prevosto, V., Signoles, J., and Yakobowski, B. Frama-C: A software analysis perspective. *Formal Asp. Comput.* (2015).
27. Kirchner, F., Sadmi, F., Flanc, S., Duboc, L., Marteau, H., Prevosto, V., and Vedrine, F. Safer marine and offshore software with formal-verification-based guidelines. In *Proc. of the 2016 Conf. on Embedded Real Time Softw. and Systems.*
28. Leino, K. Efficient weakest preconditions. *Information Processing Letters* (2005).
29. Mandrykin, M. and Khoroshilov, A. High-level memory model with low-level pointer cast support for Jessie intermediate language. *Programming and Computer Softw.* (2015).
30. Mauborgne, L. and Rival, X. Trace partitioning in abstract interpretation based static analyzers. In *Proc. of the 2005 European Symp. on Programming.*
31. Meyer, B. *Design by Contract.* Prentice Hall, 1991.
32. Monate, B. and Signoles, J. Slicing for security of code. In *Proc. of the 2008 Conf. on Trusted Computing and Trust in Information Technologies.*
33. Ourghanlian, A. Evaluation of static analysis tools used to assess software important to nuclear power plant safety. *Nucl. Eng. Technol.* (2015).
34. Pariente, D. and Signoles, J. Static analysis and runtime assertion checking: Contribution to security countermeasures. In *Proc. of the 2017 Symp. sur la Sécurité des Technologies de l'Information et des Communications.*
35. Petiot, G., Kosmatov, N., Botella, B., Giorgetti, A., and Julliand, J. How testing helps to diagnose proof failures. *Formal Asp. Comput.* (2018).
36. Prevosto, V., Burghardt, J., Gerlach, J., Hartig, K., Pohl, H., and Völlinger, K. Formal specification and automated verification of railway software with Frama-C. In *Proc. of the 2013 Int. Conf. on Industrial Informatics.*
37. Robles, V., Kosmatov, N., Prevosto, V., Rilling, L., and Gall, P. MetAcsl: Specification and verification of high-level properties. In *Proc. of the Conf. on Tools and Algorithms for the Construction and Analysis of Systems.*
38. Signoles, J., Kosmatov, N., and Vorobyov, K. E-ACSL, a runtime verification tool for safety and security of C programs. Tool Paper. In *Proc. of the 2017 Int. Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardization for Runtime Verification Tools.*
39. Venet, A. The Gauge domain: Scalable analysis of linear inequality invariants. In *Proc. of the 2012 Conf. on Computer Aided Verification.*
40. Williams, N., Marre, B., Mouy, P., and Roger, M. PathCrawler: Automatic generation of path tests by combining static and dynamic analysis. In *Proc. of the 2005 European Dependable Computing Conf.*

Supplemental material for this article is available online at https://dl.acm.org/doi/10.1145/3470569

Patrick Baudin is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

François Bobot is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

David Bühler is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

Loïc Correnson is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

Florent Kirchner is head of the department at the Université Paris-Saclay, CEA, List, Palaiseau, France.

Nikolai Kosmatov is a researcher at Thales Research and Technology, Palaiseau, France.

André Maroneze is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

Valentin Perrelle is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

Virgile Prevosto is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

Julien Signoles is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

Nicky Williams is a researcher at the Université Paris-Saclay, CEA, List, Palaiseau, France.

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

DOI:10.1145/3431283

Using clever video curation and processing practices to extract video training signals automatically.

BY TALI DEKEL AND NOAH SNAVELY

Unveiling Unexpected Training Data in Internet Video

ONE OF THE most important components of training machine-learning models is data. The amount of training data, how clean it is, its diversity, how well it reflects the real world—all can have a dramatic effect on the performance of a trained model. Hence, collecting new datasets and finding reliable sources of supervision have become imperative for advancing the state of the art in many computer vision and graphics tasks, which have become highly dependent on machine learning. However, collecting such data at scale remains a fundamental challenge.

In this article, we focus on an intriguing source of training data—online videos. The Web hosts a large volume of video content, spanning an enormous space of real-world visual and auditory signals. The existence of such abundant data suggests the following question: *How can we imbue machines with visual knowledge by directly observing the world through raw video?* There are a number of challenges faced in exploring this question.

First, while the scale of data is useful for building diverse training datasets, the problem becomes one of curation—characterizing the type of videos suited for the task at hand and automating the process of filtering irrelevant and noisy videos at scale. Second, raw videos come with no annotations or labels, except possibly noisy tags and descriptions, and so deriving reliable supervision signals for a given machine-learning task is a fundamental challenge.

For many problems, human labels are the most accurate source of supervision. Indeed, there have been ongoing efforts to generate an ImageNet-equivalent dataset for videos—large-scale, real-world video datasets with ground truth annotations. Such datasets have been mostly generated by collecting videos and manually or semi-manually gathering accurate, human-labeled annotations for various tasks. Examples include activity/action-recognition datasets[25,40] and video classification datasets.[1,24] Several datasets also contain more complex manual annotations, such as hand contact between objects in a video,[18] spatially localized labels such as bounding boxes,[22,36] dense object segmentation maps,[6,35] and some feature audio-based labels.[20] Such human-annotated video datasets are continually growing in number, size, and richness of labels and have been widely used in the research community. However, collecting and annotating natural videos is extremely challenging, requiring great effort to devise dedicated, interactive annotation tools and to perform careful analysis of the precision and quality of the labels.

Moreover, for many tasks, manual annotation may not be feasible. For example, annotating accurate depth maps from a video or ensuring sub-pixel optical flow between two frames in a video, are difficult if not impossible tasks for humans. Synthetic data can address some of these limitations by giving full control over data generation. But this approach assumes access to realistic 3D models of a wide range of scenes, and also requires learned models to generalize from synthetic data to real scenes—an uncertain proposition.

In this article, we focus on a different route—learning from videos in a *self-supervised* manner, that is, without any human labels. In particular, we show that in some cases, and sometimes unexpectedly, certain types of such raw videos can unveil powerful training signals that fit directly with a specific task. Nevertheless, determining the type of videos needed and automatically deriving such supervision signals is often non-obvious and challenging.

Our article highlights three of our recent papers that tackle such challenges for three distinct computer vision and graphics tasks: (1) taking a pair of images of a scene and predicting 3D geometry for synthesizing new views of that scene; (2) predicting dense depth maps from video in challenging scenarios, where both the camera and the people in the scene are freely moving; and (3) an audio-visual speech separation model that takes an ordinary video as input and isolates and enhances the speech of a particular speaker while suppressing all other sounds. Each of these works involves discovery of powerful supervision signals in raw videos and shows insightful creation of new datasets via clever automatic video curation and processing algorithms. The models trained on these new datasets achieve state of the art results and have been successfully applied to real-world test scenarios.

The key idea of our work, called looking-to-listen, is to use visual signals to help process the audio signal.

Related work. The work highlighted in this article falls under the umbrella of self-supervised methods that learn from unlabeled video. Such methods use training signals that are either readily available in the videos or can be fully and automatically computed from the video data. For example, video frames have been used as supervision for learning correspondences, for tracking objects,[45,46] and for various synthesis tasks, such as generating future frames in a video[26,43] or frame interpolation.[29]

But the bulk of self-supervised methods do not obtain direct supervision for the task at hand, but rather supervise an auxiliary task, which in turn allows the model to learn useful video representations.[2,11,16,30,38,44,46,47] For example, learning the temporal ordering of frames,[16,30] or learning the arrow of time of a video[47] (that is, whether a given video is playing forward or backward), allows systems to learn useful representations for action recognition. Despite the rapid progress of such methods, their performance is still inferior compared with supervised methods.

In this article, we review work that can mine Internet video for direct supervision for the task at hand but in a fully automatic manner; that is, the model is trained by directly regressing to the desired unknowns. This approach—deriving "labels" in a self-supervised manner yet training the model in a supervised manner—allows us to achieve state of the art results for various complex tasks.

Learning to Estimate 3D Geometry from Real-Estate Footage

When our 3D world is projected onto 2D images, geometric information is lost. The 3D position of objects in the scene, their 3D structure, or even their depth ordering are unknown. We show how Internet video can be of unexpected use for predicting the lost 3D geometric information from 2D image data.

For instance, consider the problem of computing a depth map from two images of a scene, as illustrated in Figure 1. Normally, if we wanted to apply supervised learning to this task, we would need to collect a dataset of images with their corresponding ground-truth depth maps, for instance, by taking a Microsoft Kinect sensor and scanning a large number of scenes.[12] However, such data collection is cumbersome and limited—for instance, Kinect sensors produce noisy, incomplete depth maps and do not work outdoors. However, if we change our perspective and make creative use of existing data, we can find surprisingly useful sources of geometric supervision from real-world online video.

In particular, one application of geometry estimation from images is *view synthesis*—taking a set of known images of a scene and synthesizing new, unobserved views of the same scene with quality suitable for computer graphics applications such as virtual reality.

We can formulate this view synthesis problem as a machine-learning problem as follows: given three images of a static scene, each from a different and known viewpoint, we select two of the images and use them as input to a deep neural-based model. We then ask the model to predict the 3D geometry of that scene (for example, in the form of a depth map) from the input image pair, and then use that estimated geometry to render that scene from the perspective of the third camera viewpoint.

The machine-learning model is judged by how well the rendered image matches the actual image. If the predicted 3D-scene model is the output of a convolutional neural network, then we can train that network using the signal arising from that comparison with the ground-truth third image. If we have many such triplets of images across many different scenes, then we can train a network that can generalize to predict good 3D representations from images of any number of new scenes.

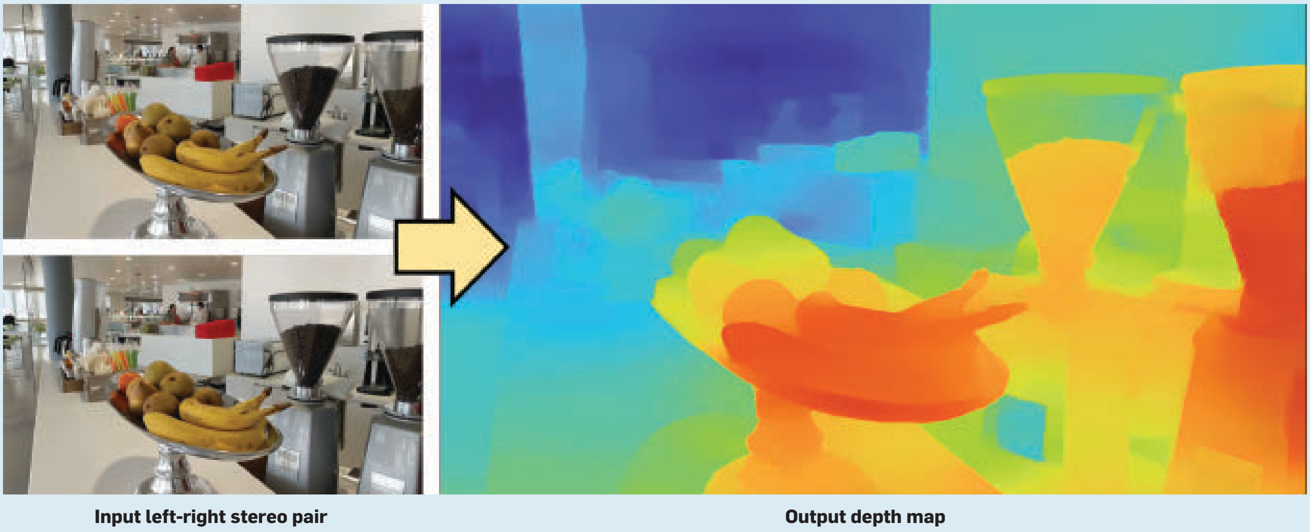
Hence, the problem of training such a network reduces to the problem of finding a large and diverse collection of image triplets of static scenes captured from known viewpoints. Previous work has also observed that 3D representations can be learned from imagery alone, but such work has used very relatively small amounts of data from, for example, lightfield cameras,[23] or has involved proprietary data, such as Google Street View.[17] Can we gather suitable data for this task from Internet videos?

At first, this task seems difficult: most online videos feature dynamic scenes (for example, with moving people), not static ones. Dynamic objects violate geometric constraints used to estimate the 3D structure of the scene, thus leading to errors and noise in the predicted geometry. However, we found that we can gather image triplets of static scenes from an unexpected type of video: real-estate footage. Typical real-estate videos feature a series of shots of indoor and outdoor scenes (the interior of a room or stairway, exterior views of a house, footage of the surrounding area, and so on). Shots typically feature smooth camera movement and little or no scene movement. Hence, we built a dataset from thousands of real-estate videos shared on the Web as a large and diverse source of multi-view training imagery.

To build this dataset, which we call RealEstate10K,[51] we devised a pipeline for mining suitable clips from YouTube. This pipeline consists of four main steps:

1. Identifying a set of candidate videos to download.
2. Running a camera tracker on each video to both estimate an initial camera pose for each frame and to subdivide the video into distinct shots/clips.

Figure 1. Computing a depth map for a static scene from two images. Left: a stereo pair of a still life. Right, a depth map computed for this scene (warmer colors represent nearer points, and cooler colors further points).



3. Performing a full optimization, known as bundle adjustment, to derive high-quality poses for each clip.

4. Filtering to remove any remaining unsuitable clips.

The key component is the camera tracker—we use an algorithm called ORB-SLAM2, originally designed for robot localization from video, to estimate the pose of the moving camera for each video frame.[31] However, we had to modify this camera tracker to be able to handle effects such as cuts and cross-fades that occur in YouTube videos in the wild. The output of our data-mining pipeline—3D camera poses and a sparse point cloud of the scene—is illustrated in Figure 2.

Figure 2. Illustration of the output of our camera tracking pipeline for a single video clip.

The camera tracker takes as input a sequence of video frames (a) and outputs a sparse 3D point cloud (b) and a 3D camera trajectory, shown as the set of wireframe pyramids (c). We then sample triplets of frames from sequences (d) like this to form training data for our view-synthesis model. Given two selected source frames (e), we can choose a target third frame either in between the two frames (g), representing a view-interpolation problem instance, or we can choose a frame outside the two frames (f), representing a view-extrapolation problem instance. (Video stills in this figure and Figure 4 are used under Creative Commons license from YouTube user *SonaVisual*.)

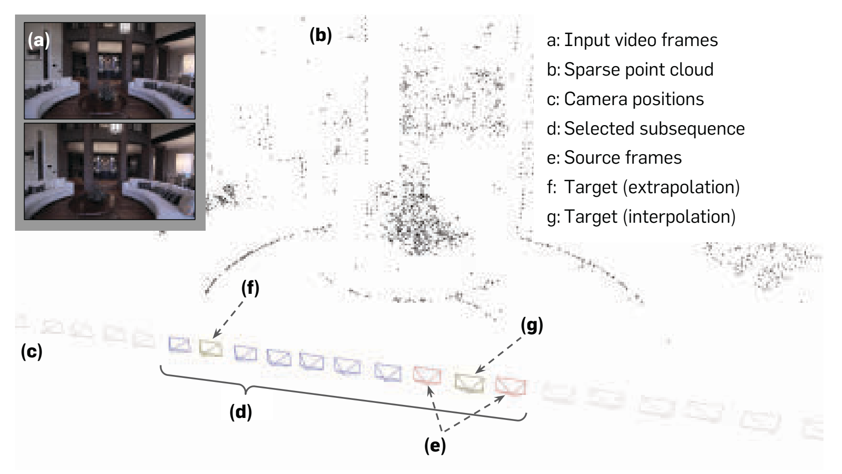
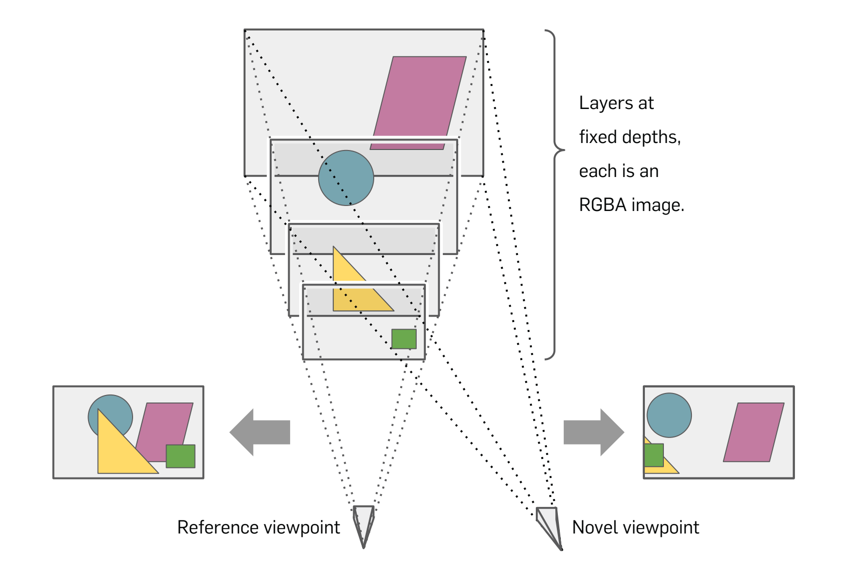


Figure 3. The multiplane image (MPI) scene format.

An MPI consists of a set of fronto-parallel planes at fixed depths from a reference camera coordinate frame, where each plane encodes an RGB image and an alpha (transparency) map that capture the scene appearance at the corresponding depth. The MPI representation can be used for efficient and realistic rendering of novel views of the scene.



While this figure shows a single tracked camera sequence, we collected thousands of such sequences from real-estate videos at scale. For each tracked sequence, we can sample triplets as frames to train a machine-learning model to perform view synthesis, as described earlier.

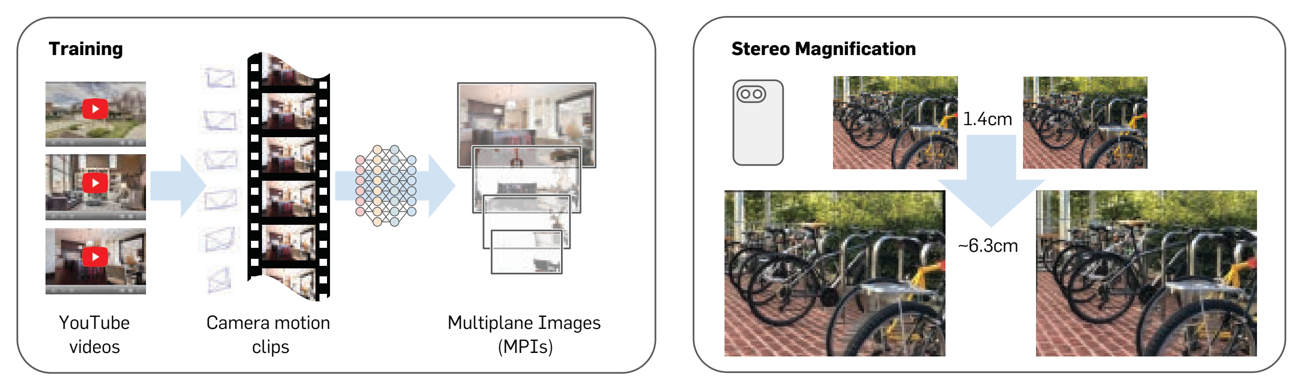
Beyond the idea of collecting training data in this way, a key design decision is how we represent the 3D scene for view synthesis. One common approach is to represent the 3D scene as a depth map, or an image representing the distance between the camera and each scene point, as illustrated in Figure 1. Given an image and a depth map, one can use the 3D information in the depth map to reproject the image to new viewpoints. However, a limitation of depth maps is that they only represent foreground scene content visible in the reference view of the depth map, not hidden surfaces that appear when the camera is moved to a new view, for example, the part of the countertop behind the fruit platter in Figure 1.

Instead, in our work, we use a layered representation called a multiplane image (MPI), so-called because it brings to mind the multiplane camera invented at Walt Disney Studios and used in traditional 2D animation.[48] The Disney version of a multiplane camera consists of a stack of planar transparencies arranged at different depths from a camera, each painted with content that should appear at a different depth (for example, a house at a nearby layer, and the moon in a further layer). By moving the transparencies at different speeds relative to the fixed camera, one can give the illusion of a 3D scene, similar to parallax scrolling in video games.

Our MPI scene format is a computational version of this idea, wherein we represent a scene as a set of RGB images with transparency arranged at fixed distances from a reference camera, as illustrated in Figure 3. To render the scene from a new viewpoint, we simply move each image in the MPI a corresponding amount and composite the transformed images in front-to-back order, also shown in Figure 3. MPIs are a very simple and convenient

Figure 4. Our stereo magnification framework.

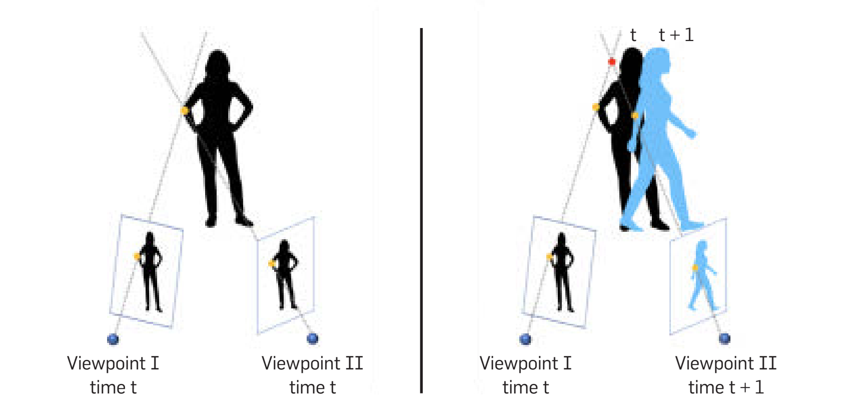
We extract camera motion clips from YouTube videos and use them to train a neural network to generate a MPI scene representation from narrow-baseline stereo image pairs. The inferred MPI representation can then be used to synthesize novel views of the scene, including ones that extrapolate significantly beyond the input baseline.



scene representation that have the advantage of being able to represent content hidden from the reference view, due to the use of multiple layers. MPIs can even handle reflective and transparent objects, and at least up to a certain amount of camera motion. MPIs are related to other layered representations used in vision and graphics, in particular the "stack of acetates" model introduced by Szeliski and Golland.[41]

Figure 4 illustrates our complete pipeline, wherein we train a deep-learning model to predict a MPI from a pair of input images using triplets of video frames as training data. We demonstrate this in an application that we call stereo magnification. The idea is that many modern cellphone cameras have two (or more) cameras that are very close together, for example, 1cm apart. From such closely spaced images, we might want to extrapolate views that are much further apart, for example, to enable a larger head motion in a virtual reality (VR) setting, or to create a stereo pair with the correct eye distance for viewing in 3D glasses. We successfully train a machine-learning model for this task and, even though we train from real-estate footage, we find that our model generalizes well to many other kinds of scenes. Please see our project Web page[a] for videos showing continuous view interpolation and extrapolation from two input frames.

Figure 5. Left: The traditional stereo setup assumes that at least two viewpoints capture the scene at the same time, and hence the 3D position of points can be computed using triangulation. Right: We consider the setup where both the camera and subject are moving, in which case triangulation is no longer possible since the so-called epipolar constraint does not apply.



While our model generalizes beyond real-estate scenes, one key assumption is that it assumes scenes are static, and a corresponding crucial challenge is *scenes with moving objects, and, in particular, people*. Estimating 3D information from multiple views of a dynamic scene poses additional challenges, which we address next using another surprising source of data.

Learning the Depth of Moving People by Watching Frozen People

We have shown how online videos of static scenes captured by a moving camera can be processed and leveraged to model the geometry of static scenes via a dedicated, learning-based, view-synthesis framework. We now show that by using a specific type of similar video (static scenes, moving cameras), we can tackle a particularly challenging task—estimating the geometry of dynamic scenes from ordinary videos, that is, when both the camera and the objects in the scene are freely moving. Most existing 3D-reconstruction algorithms assume the same object can be observed from at least two different viewpoints at the same time, which allows to compute the 2D position of points using triangulation (see Figure 5). This assumption is violated by dynamic objects when cap-

[a] https://people.eecs.berkeley.edu/˜tinghuiz/projects/mpi/

Figure 6. Learning the depth of moving people by watching frozen people.

Our model predicts dense depth from video in which both an ordinary camera and people in the scene are freely moving (right). We train our model on our new MannequinChallenge dataset, a collection of online videos of people imitating mannequins, that is, freezing in diverse, natural poses while a camera tours the scene (left). Because people are stationary, geometric constraints hold, allowing us to use techniques such as structure-from-motion (SfM) and multi-view stereo (MVS) to estimate depth, which serves as supervision during training.

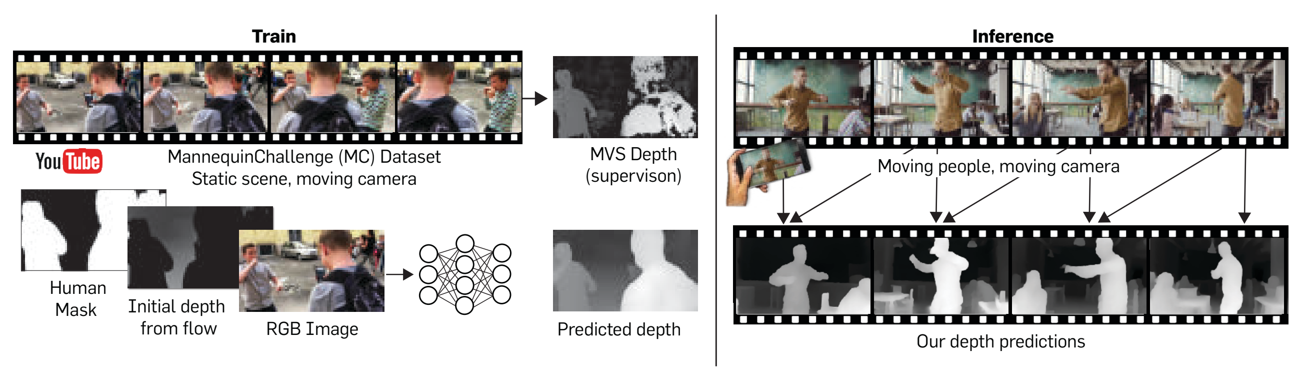
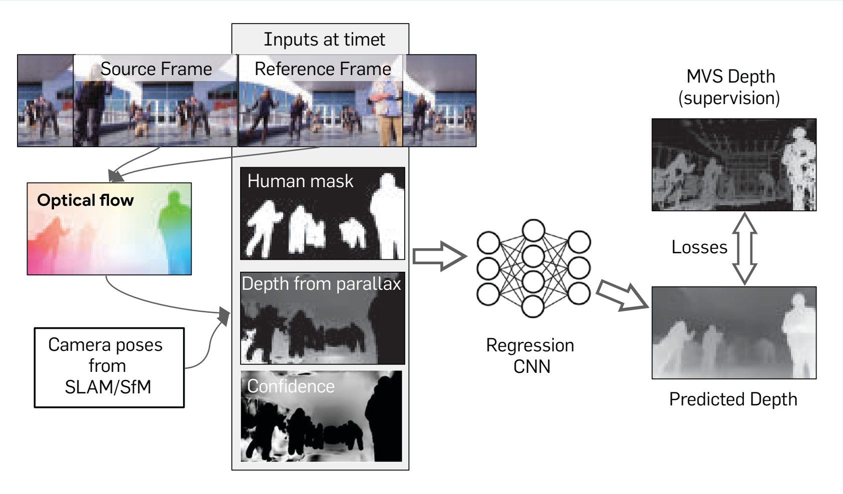


Figure 7. MannequinChallenge Dataset: (a) Each example is a frame from a MannequinChallenge video sequence in which the camera is moving but all humans are static. Because the entire scene is static, these videos span a variety of natural scenes, poses, and configurations of people.



tured by a moving camera. As a result, most existing methods either filter out moving objects (assigning them "zero" depth values) or ignore them (resulting in incorrect depth values). Our approach is to avoid imposing such geometric constraints by instead learning geometric priors about the shape and motion of dynamic objects from data.

While there has been a recent surge in the development of learning-based models for predicting geometry (for example, depth maps) from imagery, most existing methods consider only a single image as input (RGB-to-Depth) or are restricted to static scenes (as with the method we presented earlier). We extend this line of research to predicting geometry of dynamic objects from ordinary videos. More specifically, we consider the problem of predicting dense depth maps from ordinary videos when both the camera and the people in the scene are naturally moving.[28] (see Figure 6). We focus on humans because they are an interesting subject for augmented reality applications and 3D video effects. Furthermore, human motion is articulated and difficult to model, making them an important challenge to address.

MannequinChallenge Dataset. Where do we get the data needed to train a depth prediction model that can handle moving people in the scene captured by a single moving camera? Generating high-quality synthetic data in which both the camera and the people in the scene are naturally moving is very challenging. Depth sensors (for example, Kinect) can provide useful data but are typically limited to indoor environments and require significant manual work in capture.

Instead, we derive training data from a surprising source: a category of video in which people freeze in place—often in interesting poses—while the camera operator moves around the scene filming them—attempting the so-called "Mannequin Challenge."[49] Many such videos have been created and uploaded since late 2016, and these videos span a wide range of scenes with people of different ages, naturally posing in different group configurations. These videos comprise our new MannequinChallenge (MC) Dataset, which we recently released to the research community.[27]

To the extent that people succeed in staying still during the videos, these videos are no different from the real-estate videos discussed earlier—we can assume the scenes are static while the camera is moving, in which case multi-view geometric constraints and triangulation-based methods apply. We can then obtain accurate camera poses using the same camera-tracking pipeline we described previously. We can then obtain accurate depth information through further processing with vision methods known as Multi-View-Stereo (MVS). We illustrate such automatically derived depth data in Figure 7.

However, recovering accurate geometry from such raw video is challenging. First, there are videos that are not suitable for training. For example, people may "unfreeze" (start moving) at some point in the video, or the video may contain synthetic graphical elements in the background. Second, such in-the-wild videos often involve camera motion blur, shadows, or reflections. Thus, the raw depth maps estimated by MVS are often too noisy for use in training. To address these challenges, we developed an automatic framework for carefully filtering noisy video clips and individual depth values within frames in each clip (full details are described in Li.[28] This filtering is a crucial step in generating accurate, reliable supervision signals from raw video data.

Inferring the depth of moving people. Our data provides depth supervision for a moving camera and "frozen" people, but our goal is to handle videos with a moving camera and moving people. We need a machine-learning model that can bridge this gap.

One approach would be to infer depth separately for each frame of the video (such as RBG-to-Depth). We tried this, and while such a model already improves over state of the art single-image methods for depth prediction, this approach disregards depth information about the rigid (static) parts of scenes that can be inferred when considering more than a single frame. To benefit from such information, we design a two-frame model that uses depth information computed from motion parallax, that is, the relative apparent motion of static objects between two different viewpoints. In particular, we

Figure 8. Our model takes as input an RGB frame, a human segmentation mask, masked depth computed from motion parallax (via optical flow and SfM pose), and an associated confidence map. We ask the network to use these inputs to predict depths that match the ground -truth MVS depth.

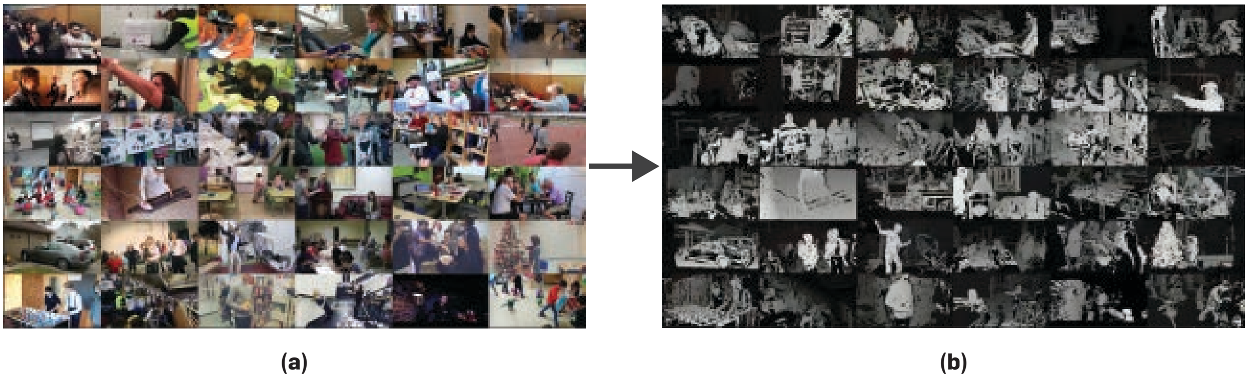


Figure 9. Depth-prediction results on video clips with moving cameras and people.

From left to right: (a) sample frames from the input video, (b-c) results of learning-based monocular depth-prediction methods;[8,19] (d) learning-based stereo depth-prediction method;[42] and (e) results of our method.

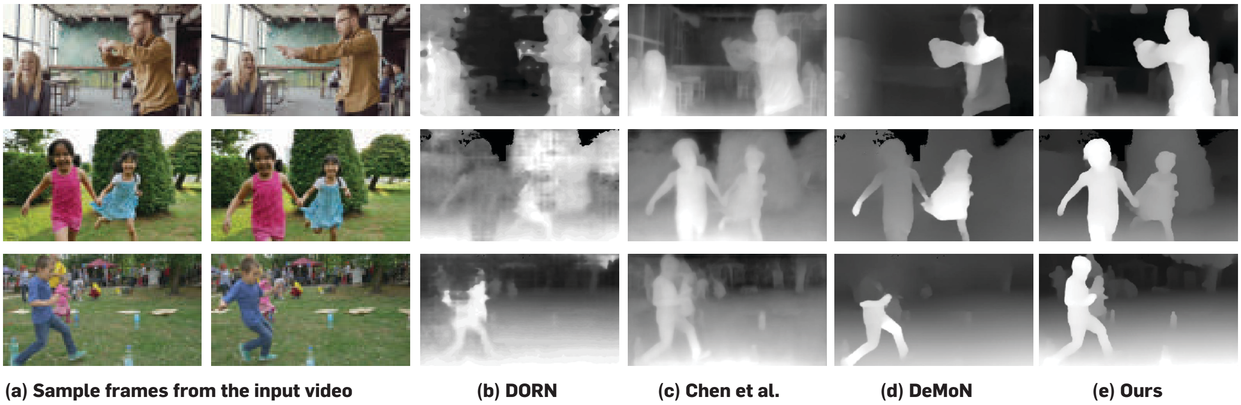
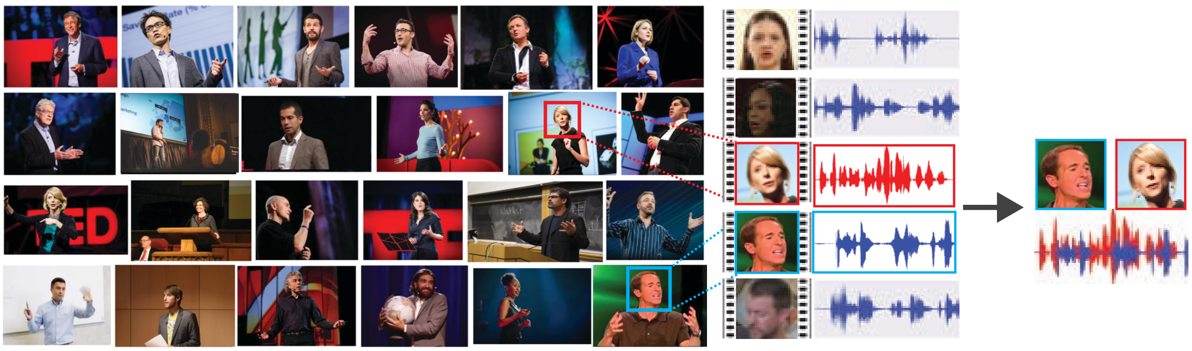


Figure 10. AVSpeech Dataset: We first gathered a large collection of 290,000 high-quality, online public videos of talks and lectures (a). Using audio and video processing, we extracted video segments with clean speech (no mixed music, audience sounds, or other speakers), and with the speaker visible in the frame (b). This resulted in 4,700 hours of video clips, spanning a wide variety of people, languages, and face poses. Each segment contains a single person talking with no background interference. From these clean segments, we then generate training examples of "synthetic cocktail parties" by mixing the audio tracks of different speakers (c).



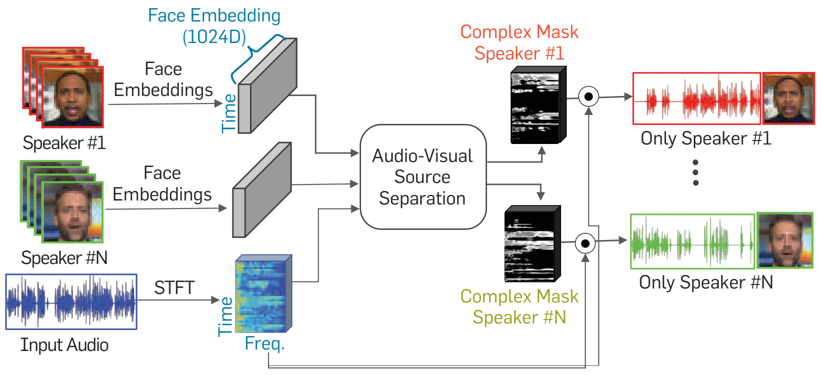
(a) Large collection of Internet videos of talks and lectures

(b) Video segments with localized speakers and clean speech

(c) Synthetic cocktail parties: mixed speech of different speakers

Figure 11. Audio-visual speech-separation model: We start by detecting and tracking talking people in an input video and compute a feature (face embedding) for each of the face thumbnails detected in each frame.

A complex spectrogram represents the input audio, which contains a mixture of speech and background noise. The network outputs a complex spectrogram mask for each speaker, which is multiplied by the noisy input and converted back to waveforms to obtain an isolated speech signal for that speaker.



first compute 2D-optical flow between each input frame and another frame in the video. This flow field depends on both the scene's depth and the relative position of the camera. However, because the camera positions are known, we can remove their dependency from the flow field, which results in an initial depth map.

At test time, since people are moving, the computed depth map would be incorrect in the human regions. We therefore segment and mask out those regions and only supply to the network depth information for the static environment, as illustrated in Figure 8. The network's job is to "inpaint" the depth values for the regions with people and refine the depth elsewhere. Our two-frame model leads to a significant improvement over the RGB-only model for both human and non-human regions.

Depth prediction results. In Figure 9, we show some examples of our depth-prediction model results on real videos, with comparison to recent state-of-the-art learning-based methods. Our depth maps are significantly more accurate and more consistent over time. Armed with the estimated depth maps, we can produce a range of 3D-aware video effects, including synthetic depth defocusing, generating a stereo video from a monocular one, and inserting synthetic computer-generated (CG) objects into the scene. Our depth maps also provide the ability to fill in holes and discolored regions with the content exposed in other frames of the video. Please see our webpage for a full set of results.[b]

Looking-to-Listen: Audio-Visual Source-Separation Model

In the previous two sections, we showed how raw online video can provide powerful visual signals that can be used as training data for complex visual tasks. Here, we go beyond visual signals by also leveraging auditory signals found in ordinary video. More specifically, our goal is to tackle the cocktail party problem—isolating and enhancing a single voice of a desired speaker from a mixture of sounds, such as background noise and other speakers.[9] Humans can do this very well—we have a remarkable ability to focus our auditory attention on a particular speaker while filtering out all other voices and sounds.[9] We want to teach machines this same abil-

[b] https://mannequin-depth.github.io/

ity by observing auditory and visual signals in online video.

The key idea of our work, called looking-to-listen, is to use visual signals to help process the audio signal. Intuitively, facial features—such as mouth movements or even facial expressions—should correlate with the sounds produced when that person speaks, which in turn can help to identify and isolate that person's speech signal from a mixture of sounds. To do so, we design and train a joint audio-visual model, where the input to the model is an ordinary video (frames + audio track), and the output is clean speech tracks, one for each person detected in the video. This is the first audio-visual, speaker-independent separation model; that is, the model is trained only once and then can be applied to any speaker at test time.

For many cross-modal tasks, the natural co-occurrence of audio and visual signals in video can readily provide supervision. Examples include learning audio-video representations,[3,5,7] cross-modal retrieval,[32,33,39] or sound source localization.[3,32,37,50] However, in our case, in order to train our model in a supervised manner, we need regular videos with mixed speech and background noise as input—and also ground-truth separated audio tracks for each of the speakers as supervision. Existing video does not provide such supervision and directly recording it at scale would have been difficult. However, we can think of ways to generate the exact training data we need from existing raw online video.

Specifically, we use online videos of talks, lectures, and how-to videos. Many of these videos contain a single, visible speaker with a clean recording of their speech and no interfering sounds. With these clean videos in hand, we can then generate training examples of "synthetic cocktail parties" by mixing clean audio tracks of different speakers and background noise, as illustrated in Figure 10. This allows us to train an audio-visual speech-separation model in a supervised manner by directly regressing to the clean audio tracks for each of the speakers.

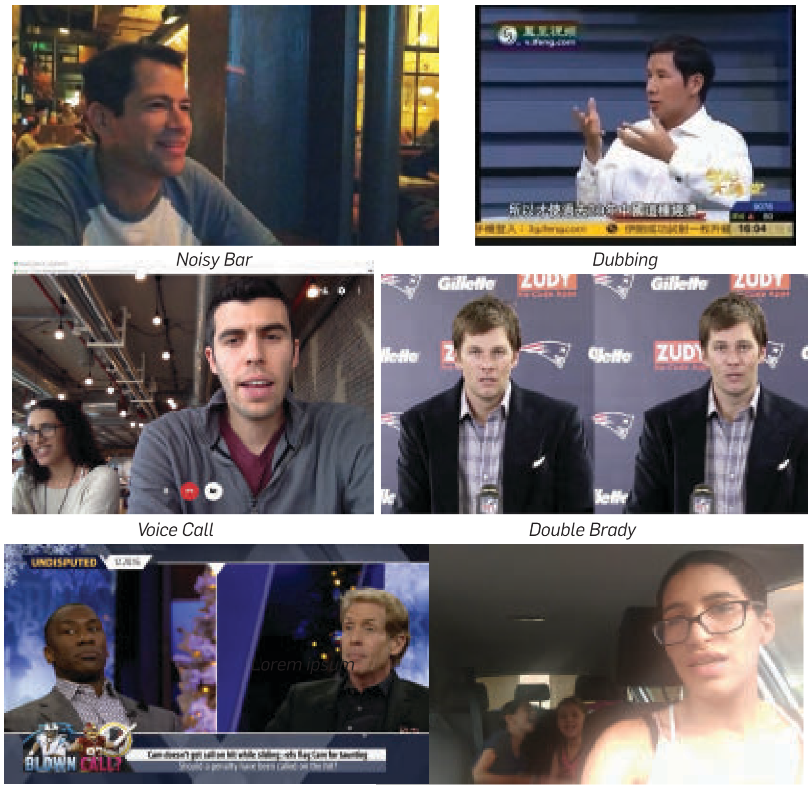
AVSpeech dataset. We collected a large amount of high-quality video clips, each containing a single, visible speaker with a clean recording of his or her speech and no other interfering background sounds. To do so, we started by crawling around 290,000 candidate videos from YouTube channels of lectures, TED Talks, and how-to videos. However, as discussed previously, raw video data never provides clean, perfectly accurate training data. In this case, significant parts of such videos may not be suitable for training, either because of the visual content (for example, shots of the audience, slides, or other visuals that do not include the speaker), or because of the audio content (for example, noisy speech). Avoiding such cases and assembling a large reliable corpus of data is a crucial step that calls for automatic processing that does not rely on human feedback. We achieve that by designing a dedicated filtering mechanism based on both video and audio processing, as described in detail in Ephrat.[15]

After filtering, we obtain roughly 4,700 hours of video segments with approximately 150,000 distinct speakers, spanning a wide variety of people, languages, and face poses. The dataset is available for academic use.[14] From these videos, training examples of synthetic cocktail parties are generated by mixing clean audio tracks of different speakers and background noise from the AudioSet dataset,[21] as illustrated in Figure 4 (b-c).

Audio-Visual Speech Separation model. With the AVSpeech dataset in hand, we design and train a model to decompose the synthetic cocktail mixture into clean audio streams for each speaker in the video, as illustrated in Figure 11.

The model takes both visual and auditory features as input. For the visual features, we only consider the face region by first detecting all the faces in each frame using an off-the-shelf face detector (for example, the Google Cloud Vision API). For each of the detected face thumbnails, we compute a visual

Figure 12. Speech separation in the wild: Representative frames from natural videos demonstrate our method in various real-world scenarios. All videos and results can be found in the project webpage. The "Undisputed Interview" video is courtesy of Fox Sports.



Noisy Bar

Dubbing

Voice Call

Double Brady

Undisputed Interview

Car

feature. This is done by feeding each face thumbnail to a pre-trained face recognition model and extracting the features from the lowest layer in the network, similar to the one used by Cole.[10] The rationale is that these features retain information necessary for recognizing millions of faces, while discarding irrelevant variation between images, such as illumination. For audio features, we use complex spectograms computed by the short-time Fourier transform (STFT) of three-second audio segments.

Our model first processes the visual and auditory signals separately and then fuses them together to form a joint audio-visual representation. With that joint representation, the network learns to output a time-frequency mask for each speaker. The output masks are multiplied by the noisy input spectrogram and converted back to a time-domain waveform to obtain an isolated, clean speech signal for each speaker. The model also outputs one mask for the background interference.

During training, the squared L_2 error between the clean spectrogram and the predicted spectrogram is used as a loss function to train the network. At inference time, our separation model can be applied to arbitrarily long segments of video and varying numbers of speakers. The latter is achieved by either directly training the model with multiple-input visual streams (one for speaker), or simply by feeding the visual features of the desired speaker to the visual stream. For full details about the architecture and training process, see our full paper.[15]

Speech separation results. Once trained, our model can be applied to real-world videos with arbitrary speakers. Figure 12 shows representative frames from an assortment of such videos containing heated debates and interviews, noisy bars, and screaming children. For all these challenging videos, our model successfully isolates and enhances the speech of the desired speaker, while suppressing all other sounds. See our project page for a full set of results.[c]

The "Double Brady" video is a synthetic example, in which we concatenated two different segments from the same video side by side. This is an extremely challenging case because the two speakers are identical (same voice, same appearance), only time delayed. Our audio-visual model successfully achieves a clean separation result in this case and significantly outperforms a state-of-the-art audio-only model. This highlights the utilization of visual information by our model. See our paper for a thorough numerical evaluation and comparison to a state-of-the-art audio-only model.[15]

This technology has recently launched in YouTube Stories, a sharing platform for short, mobile-only videos. Many of these videos are selfies taken in noisy locations (for example, parties or sporting events) with low-cost cellphone microphones. Our looking-to-listen technology now allows creators to isolate their speech from all other voices and sounds.

Internet video can be of unexpected use for predicting the lost 3D geometric information from 2D image data.

Conclusion and Future Work

The contents of online videos depict a wide array of phenomena that can be used to teach machines about our world. With such a rich resource available, part of the creative pursuit of research today involves identifying interesting types of information that can be automatically derived or synthesized from video and devising methods for identifying and processing such data.

While the visual content provided by raw video is enormous, it does not always perfectly represent our true visual world. For example, videos of some actions (for example, "brushing teeth") may be difficult to find, even though they are performed daily by billions of people.[15] Nevertheless, even if just a small fraction of videos is suitable for a given task, the sheer quantity of data allows for the creation of novel, real-world datasets—a key intellectual question then becomes finding and leveraging these needles in haystacks for specific tasks.

In this article, we reviewed several recent research efforts that use this insight to automatically source training data from noisy, raw online video for computer vision applications. However, this work is but an initial glimpse into a universe of possibility. We envision a wide spectrum of visual signals that can be automatically derived from In-

[c] https://looking-to-listen.github.io/

ternet video and can be used to teach machines about our world.

For example, we can envision training a model to estimate illumination by identifying videos of the same location taken at different times of day. On the theme of finding scenes under multiple illuminations, prior work in graphics and vision shows the power of reasoning about pairs of images of a scene with and without camera flash.[13,34] Can we mine such pairs of frames automatically from video of events like red carpet galas, and learn about shape and appearance?

Another example of finding needles in a haystack is identifying instances of known objects, such a specific kind of soda can where all instances have the same shape and material properties. Such objects can be thought of as accidental "light probes" that can reveal aspects of the incoming illumination, and thus provide training signals for scene structure, material properties, and illumination.

Ultimately, we hope that training data derived from raw videos will be expanded to robotics and autonomous navigation, where machines must operate in a rich variety of scenes and perform a wide variety of tasks but explicit training data is in limited supply.

Acknowledgments

The work highlighted in this article has been done in collaboration with Zhengqi Li, Forrester Cole, Richard Tucker, Ariel Ephrat, Inbar Mosseri, Oran Lang, Kevin Wilson, Avinatan Hassidim, Michael Rubinstein, Ce Liu, and William T. Freeman.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. YouTube-8m: A large-scale video classification benchmark (2016); arXiv:1609.08675.
2. Agrawal, P., Carreira, J., and Malik, J. Learning to see by moving. In *Proc. of the 2015 Int. Conf. on Computer Vision*, 37–45.
3. Arandjelovic, R. and Zisserman, A. Look, listen and learn. In *Proc. of the 2017 Int. Conf. on Computer Vision*, 609–617.
4. Arandjelovic, R. and Zisserman, A. Objects that sound. In *Proc. of the 2018 European Conf. on Computer Vision*, 435–451.
5. Aytar, Y., Vondrick, C., and Torralba, A. SoundNet: Learning sound representations from unlabeled video. *Neural Information Processing Systems* (2016), 892–900.
6. Caelles, S., Montes, A., Maninis, K-K., Chen, Y., Van Gool, L., Perazzi, F., and Pont-Tuset, J. The 2018 DAVIS Challenge on Video Object Segmentation; arXiv:1803.00557.
7. Castrejon, L., Aytar, Y., Vondrick, C., Pirsiavash, H., and Torralba, A. Learning aligned cross-modal representations from weakly aligned data. In *Proc. of the 2016 Conf. Computer Vision and Pattern Recognition*.
8. Chen, W., Fu, Z., Yang, D., and Deng, J. Single-image depth perception in the wild. *Neural Information Processing Systems* (2016), 730–738.
9. Cherry, E.C. Some experiments on the recognition of speech, with one and with two ears. *The J. Acoustical Society of America* (1953).
10. Cole, F., Belanger, D., Krishnan, D., Sarna, A., Mosseri, I., and Freeman, W.T. Synthesizing normalized faces from facial identity features. In *Proc. of the 2017 Conf. Computer Vision and Pattern Recognition*.
11. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. Temporal cycle-consistency learning. In *Proc. of the 2019 Conf. Computer Vision and Pattern Recognition*.
12. Eigen, D., Puhrsch, C., and Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *Neural Information Processing Systems* (2014), 2366–2374.
13. Eisemann, E. and Durand, F. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graphics* (2004).
14. Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W.T., and Rubinstein, M. AVSpeech Dataset (2018); https://looking-to-listen.github.io/avspeech/.
15. Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W.T., and Rubinstein, M. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *ACM Trans. Graphics 37*, 4 (2018), 112.
16. Fernando, B., Bilen, H., Gavves, E., and Gould, S. Self-supervised video representation learning with odd-one-out networks. In *Proc. of the 2017 Computer Vision and Pattern Recognition*, 3636–3645.
17. Flynn, J., Neulander, I., Philbin, J., and Snavely, N. DeepStereo: Learning to predict new views from the world's imagery. In *Proc. of the 2016 Conf. Computer Vision and Pattern Recognition*.
18. Fouhey, D.F., Kuo, W., Efros, A.A., and Malik, J. From lifestyle vlogs to everyday interactions. In *Proc. of the 2018 Conf. Computer Vision and Pattern Recognition*, 4991–5000.
19. Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D. Deep ordinal regression network for monocular depth estimation. In *Proc. of the 2018 Conf. Computer Vision and Pattern Recognition*.
20. Gemmeke, J.F., Ellis, D.P.W., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., and Ritter, M. AudioSet: An ontology and human-labeled dataset for audio events. In *Proc. of the 2017 ICASSP*, 776–780.
21. Gemmeke, J.F., Ellis, D.P.W., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., and Ritter, M. AudioSet: An ontology and human-labeled dataset for audio events. In *Proc. of the 2017 ICASSP*.
22. Gu, C. et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *Proc. of the 2018 Conf. Computer Vision and Pattern Recognition*, 6047–6056.
23. Kalantari, N.K., Wang, T-C., and Ramamoorthi, R. Learning-based view synthesis for light field cameras. *SIGGRAPH ASIA 2016*.
24. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale video classification with convolutional neural networks. In *Proc. of the 2014 Conf. Computer Vision and Pattern Recognition*.
25. Kay, W. et al. The kinetics human action video dataset (2017); arXiv:1705.06950.
26. Kwon, Y-H and Park, M-G. Predicting future frames using retrospective cycle GAN. In *Proc. 2019 IEEE Conf. Computer Vision and Pattern Recognition*, 1811–1820.
27. Li, Z., Dekel, T., Cole, F., Tucker, R., and Snavely, N. MannequinChallenge Dataset (2019); https://google.github.io/mannequinchallenge/.
28. Li, Z., Dekel, T., Cole, F., Tucker, R., Snavely, N., Liu, C., and Freeman, W.T. Learning the depths of moving people by watching frozen people. In *Proc. of the 2019 Conf. Computer Vision and Pattern Recognition*, 4521–4530.
29. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., and Agarwala, A. Video frame synthesis using deep voxel flow. In *Proc. of the 2017 IEEE Intern. Conf. on Computer Vision*, 4463–4471.
30. Misra, I., Zitnick, C.L., and Hebert, M. Shuffle and learn: Unsupervised learning using temporal order verification. In *Proc. European Conf. on Computer Vision*. Springer (2016), 527–544.
31. Mur-Artal, M.J.M.M., Tardós, R., and Tardós, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. on Robotics 31*, 5 (2015).
32. Owens, A. and Efros, A.A. Audio-visual scene analysis with self-supervised multisensory features. In *Proc. of the 2018 European Conf. on Computer Vision*.
33. Owens, A., Isola, P., McDermott, J., Torralba, A., Adelson, E.H., and Freeman, W.T. Visually indicated sounds. In *Proc. of the 2016 Conf. Computer Vision and Pattern Recognition*, 2405–2413.
34. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., and Toyama, K. Digital photography with flash and no-flash image pairs. *ACM Trans. Graphics*, 2004.
35. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., and Van Gool, L. The 2017 DAVIS challenge on video object segmentation; arXiv:1704.00675.
36. Real, E., Shlens, J., Mazzocchi, S., Pan, X., and Vanhoucke, V. YouTube-bounding boxes: A large high-precision human-annotated data set for object detection in video. In *Proc. of the 2017 Conf. Computer Vision and Pattern Recognition*, 5296–5305.
37. Senocak, A., Oh, T-H., Kim, J., Yang, M-H., and Kweon, I.S. Learning to localize sound source in visual scenes. In *Proc. of the 2018 Conf. Computer Vision and Pattern Recognition*.
38. Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., and Levine, S. Time-contrastive networks: Self-supervised learning from video. In *Proc. of the 2018 IEEE Intern. Conf. Robotics and Automation*.
39. Soler, M., Bazin, J-C., Wang, O., Krause, A., and Sorkine-Hornung, A. Suggesting sounds for images from video collections. In *Proc. of the 2016 European Conf. on Computer Vision*. Springer, 900–917.
40. Soomro, K., Zamir, A.R., and Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild (2012); arXiv:1212.0402.
41. Szeliski, R. and Golland, P. Stereo matching with transparency and matting. *Int. J. of Computer Vision* (1999).
42. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., and Brox, T. DeMoN: Depth and motion network for learning monocular stereo. In *Proc. of the 2017 Conf. Computer Vision and Pattern Recognition*.
43. Vondrick, C., Pirsiavash, H., and Torralba, A. Generating videos with scene dynamics. *Advances in Neural Information Processing Systems* (2016), 613–621.
44. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., and Murphy, K. Tracking emerges by colorizing videos. In *Proc. of the 2018 European Conf. on Computer Vision*, 391–408.
45. Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., and Li, H. Unsupervised deep tracking. In *Proc. of the 2019 Conf. Computer Vision and Pattern Recognition*, 1308–1317.
46. Wang, X., Jabri, A., and Efros, A.A. Learning correspondence from the cycle-consistency of time. In *Proc. of the 2019 Conf. Computer Vision and Pattern Recognition*.
47. Wei, D., Lim, J.J., Zisserman, A., and Freeman, W.T. Learning and using the arrow of time. In *Proc. of the 2018 Conf. Computer Vision and Pattern Recognition*, 8052–8060.
48. Wikipedia. Multiplane camera, 2017; https://en.wikipedia.org/wiki/Multiplane_camera.
49. Wikipedia. Mannequin Challenge, 2018; https://en.wikipedia.org/wiki/Mannequin_ Challenge.
50. Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J., and Torralba, A. The sound of pixels. In *Proc. of the 2018 European Conf. on Computer Vision*, 570–586.
51. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., and Snavely, N. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graphics 37*, 4 (2018), 65.

Tali Dekel is a research scientist at Google and an assistant professor on the faculty of Mathematics and Computer Science at the Weizmann Institute of Science, Rehovot, Israel.

Noah Snavely works at Google Research and is an associate professor in the Computer Science Department at Cornell Tech in the Cornell Graphics and Vision Group, New York, NY, USA.

DOI:10.1145/3474089

A method for reducing delivery delays for multimedia data produced by the Internet of Things.

BY XIAONAN WANG AND XINGWEI WANG

Multimedia Data Delivery Based on IoT Clouds

THE INTERNET OF Things (IoT) comprises embedded sensing and computing devices and aims to improve quality of life.[5,13] Mobile IoT can be applied in all walks of life, such as safe driving and smart healthcare.[2,16] For example, smartphones can produce road-safety multimedia data for safe and efficient driving, and sensor nodes attached to patients are capable of sensing vital signs for real-time monitoring.

With the IoT's dramatic development, the data it produces is experiencing a shift from simple text data to enormous multimedia data that is usually locally relevant and of a size that individuals cannot handle due to resource limitations.[1,10] Therefore, one of the most critical challenges facing IoT-related multimedia data is to improve its delivery efficiency by sharing resources with IoT devices to collaboratively produce and rapidly provide locally relevant multimedia data.[10]

Internet Protocol (IP) version 6 could be the IoT's future.[5,13] However, if IP-based data delivery methods[3,11,15,23,25] are directly deployed in the IoT, they might be confronted with the following challenges:[2,19]

- IP-based data delivery methods are centered on end-to-end communications;[25] that is, each data-delivery process is independently performed between a source and destination pair. A source node must retrieve data from a destination node even if an intermediate device can provide target data. Consequently, the data-delivery latency is considerable. Moreover, if the destination is unreachable, the data-delivery process might fail.[3,9]
- Because of the end-to-end feature, it is difficult for devices to collaborate, create, and share multimedia data because each data-delivery process is performed independently.[23] However, it is significant for IoT devices to collaborate because of their limited resources.
- If a device moves to a new IP domain, it must be configured with a Care-of Address (CoA).[12] Since CoA configuration is time-consuming, it introduces extra data-delivery latency and causes data-delivery failures.[18,22]

According to Kahn et al.,[8] Mobile Cloud (MC) is a novel technology that integrates cloud computing with mobile devices such as smartphones to overcome the resource limitations of an individual mobile device. In MC, cloud members can share resources and collaborate to create multimedia

» key insights

- **It is difficult to achieve collaboration and data sharing among IoT devices via IP-based data-delivery methods because each data-delivery process is performed independently.**
- **Mobile Cloud's features can help create a new method of achieving IoT because cloud members can share resources and collaborate to produce multimedia data.**
- **We integrate Mobile Cloud with IoT to construct IoT Cloud, allowing IoT Cloud members to collaboratively generate, locally maintain, and rapidly provide multimedia data.**

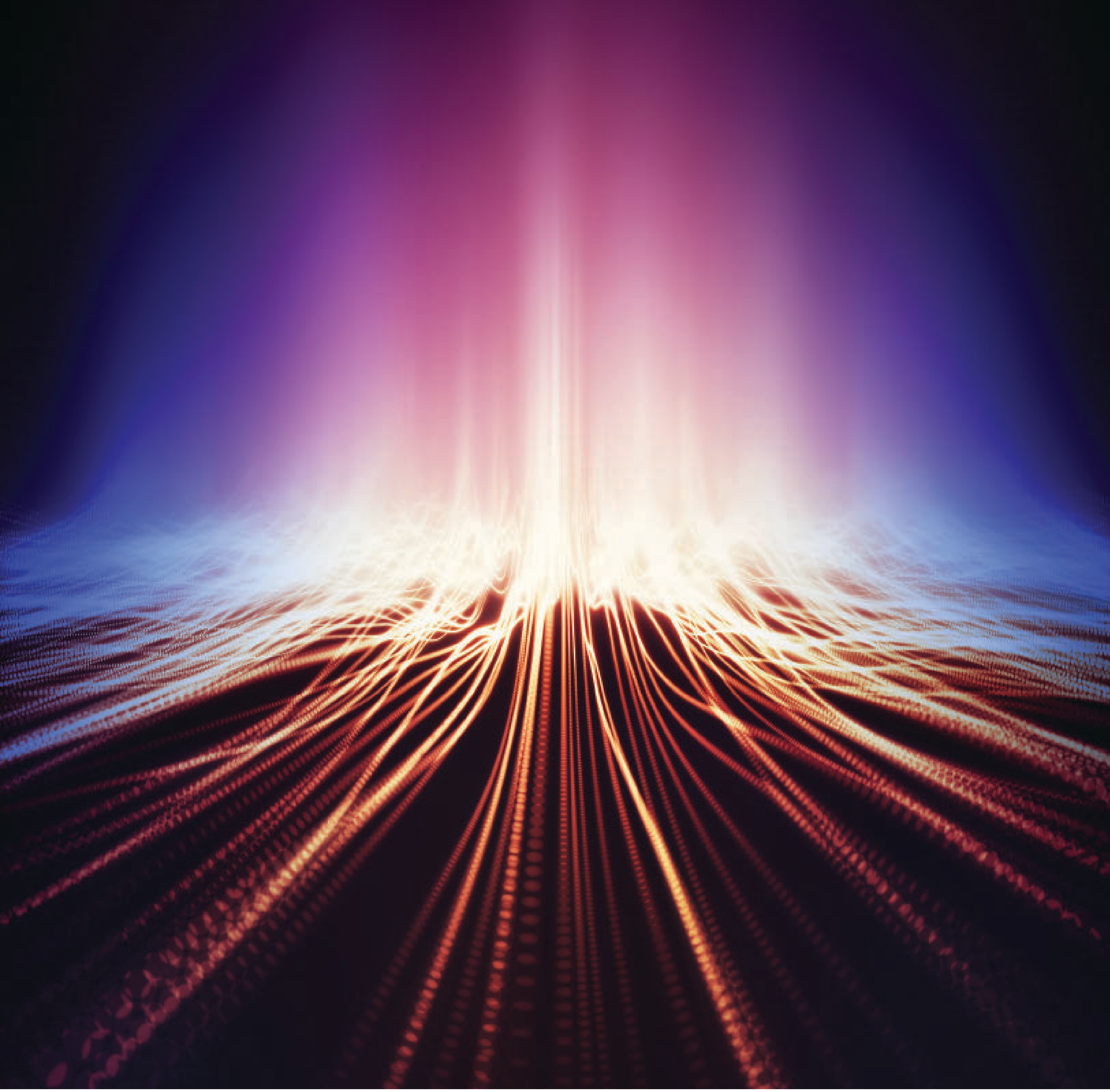


IMAGE BY KTSDESIGN

data, and a user can acquire data from any cloud member.[7,10,14] The features of MC might help overcome these challenges and create a new method of achieving IoT. Based on this observation, we are motivated to integrate MC with IoT to construct IoT Cloud (ITC) and achieve the following objectives:

- ITC members can collaboratively generate and locally maintain multimedia data so they can rapidly provide them.
- A user can retrieve multimedia data from the nearest cloud member so data-delivery latency can be reduced.
- A user can acquire multimedia data without CoA configuration to improve data-delivery success rates.

Based on the motivation and objectives, we propose a multimedia data delivery based on ITC (MDITC) and aim to reduce multimedia data-delivery latency and improve success rates via the following innovations:

- An ITC construction algorithm is proposed so IoT devices can collaborate to create locally relevant multimedia data. Based on the ITC construction algorithm, a multimedia data-delivery algorithm is proposed so a user can acquire data from the nearest member rather than a target device defined by a destination address. Moreover, a device is identified by an invariable device ID, which may be a media access control (MAC) address or may be configured by employing existing addressing methods and does not need to perform CoA configuration. Consequently, data-delivery failures caused by address changes are avoided.
- Multi-hop Ethernet architecture is

Figure 1. LRP configuration.

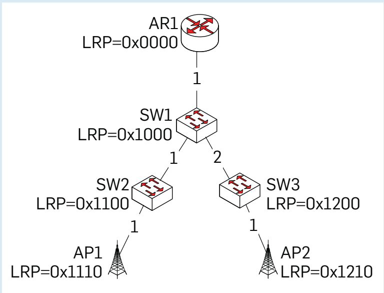


Table 1. Acronyms.

Acronym	Full name
IoT	Internet of Things
IP	Internet Protocol
CoA	Care of Address
MC	Mobile Cloud
ITC	IoT Cloud
MAC	Media Access Control
AR	Access Router
AP	Access Point
GRP	Global Routing Prefix
LRP	Link Routing Prefix

Table 2. Cloud address.

128-i-j-k	*i*	*j*	*k*
GRP	LRP	Cloud flag (1)	Content ID

Table 3. Unicast address.

128-i-j-k	*i*	*j*	*k*
GRP	LRP	Cloud flag (0)	Device ID

introduced to IoT. The proposed architecture consists of the multi-hop Ethernet backbone, made up of infrastructures, and the ITC, composed of mobile IoT devices. The proposed scheme is a combination of both wired and wireless systems because it consists of wired infrastructures, such as routers or switches, and wireless IoT devices, such as sensor nodes or smartphones. Based on the multi-hop Ethernet architecture, a multimedia data request aggregation mechanism is proposed to enable multiple users to retrieve multimedia data via one data-delivery process in parallel. Since the size of multimedia data is large, request aggregation substantially reduces data-delivery latency. Moreover, the multi-hop Ethernet architecture expands the area covered by a local network so, in most cases, a user can retrieve data locally. Consequently, data-delivery latency is further reduced.

The proposed scheme focuses on producing and sharing locally relevant multimedia data with a relatively short life span, such as road condition information, and aims to provide data rapidly. While IoT devices can upload and download data to and from the Internet cloud, the Internet cloud is costly and time-consuming, especially for locally relevant data with a relatively short life span.[10] Moreover, it is usually local or nearby users who are interested in locally relevant data.[24]

For example, drivers are interested in local and nearby road-safety information to improve driving safety and efficiency. Therefore, in the proposed scheme, IoT cloud members share and maintain data locally so that users can rapidly acquire data from the nearest member instead of the Internet cloud. The typical application of the proposed scheme is that users can quickly acquire multimedia data on road conditions associated with specific locations, such as intersections, to improve pedestrian safety and efficiency.[17,24] For instance, a pedestrian can effectively avoid road hazards or accidents by rapidly retrieving multimedia data on road conditions from the nearest IoT cloud member.

IoT Architecture

The IoT architecture comprises infrastructures and devices. Infrastructures, such as access routers (ARs), switches, and access points (APs), mainly perform routing functions, while devices, such as sensor nodes and smartphones, primarily collect location-related multimedia information. Devices are associated with an AP via wireless interfaces. A subnet is made up of one AR, switches, APs, and associated devices. The infrastructures in a subnet are organized into a tree topology, where the root is the AR, the intermediate nodes are the switches, and the leaf nodes are the APs, as shown in Figure 1. The devices associated with an AP construct an IoT Cloud and can collaborate to produce location-related multimedia data. The acronyms used by the proposed scheme are shown in Table 1.

To efficiently seek and retrieve location-related multimedia data, MDITC proposes a cloud address and a unicast address, as shown in Tables 2 and 3. A cloud address includes the global routing prefix (GRP), link routing prefix (LRP), cloud flag, and content ID. The GRP identifies a subnet, the LRP defines a specific location in the subnet, and the content ID identifies a kind of multimedia data, so a cloud address uniquely defines a specific type of location-related data. Since ITC is constructed to generate a type of location-related multimedia data, it is also identified by a cloud address. In a cloud address, the cloud flag is 1. To rapidly locate a device or an infrastructure, a unicast address is proposed and consists of the GRP, LRP, cloud flag, and device ID. In a unicast address, the cloud flag is 0. The device ID of an AR is 1 and the device ID of a switch or an AP is 0.

The *i*-bit LRP is divided into hierarchies, and *c* is an adjustment coefficient that represents one hierarchy and guarantees the uniqueness of LRP. The infrastructures at different depths of a tree are configured with a unique LRP with different hierarchies that are proportional to the depths. The LRP of an AR is 0, and it launches the following LRP configuration process:

1. The AR sends a *Conf-LRP* message from each interface *f* linking with a switch or an AP, and the payload in *Conf-LRP* includes *c*, *f*, and *h*. *h* is the hierarchy in a tree, and the initialization value is 0.
2. The infrastructure receiving *Conf-LRP* constructs its LRP, where the first $h{\cdot}c$ bits are equal to the ones of the source LRP in *Conf-LRP*, the following *c* bits are *f*, and the last $i\text{-}(h{+}1){\cdot}c$ bits are 0.
3. If the infrastructure is a switch, it increases *h* by 1, forwards *Conf-LRP* from each interface *f* linking with a switch or an AP, and goes to Step 2.
4. The LRP configuration is complete.

As shown in Figure 1, where *i* is 16 and *c* is 4, AR1 launches an LRP configuration process by sending *Conf-LRP* from interface 1. Switch SW1 receiving *Conf-LRP* is configured with LRP 0x1000 and forwards *Conf-LRP* from interfaces 1 and 2. Similarly, switch SW2/SW3 receiving *Conf-LRP* is configured with LRP 0x1100/0x1200 and forwards *Conf-LRP* from interface 1. Finally, AP1/AP2 receiving *Conf-LRP* is configured with LRP 0x1110/0x1210.

Multimedia Data Delivery

Each infrastructure maintains an index table to store the information on ITC members, and each index entry includes the interface and cloud address.

Multimedia data generation. Location-related multimedia data MD1 is identified by cloud address CA1, where the GRP is GRP1, the LRP is equal to the LRP of AP2, and the content ID is CID1. Messages *Gen-MD* and *Share-MD* are used to generate location-related multimedia data. AP2 generates MD1 by creating ITC1:

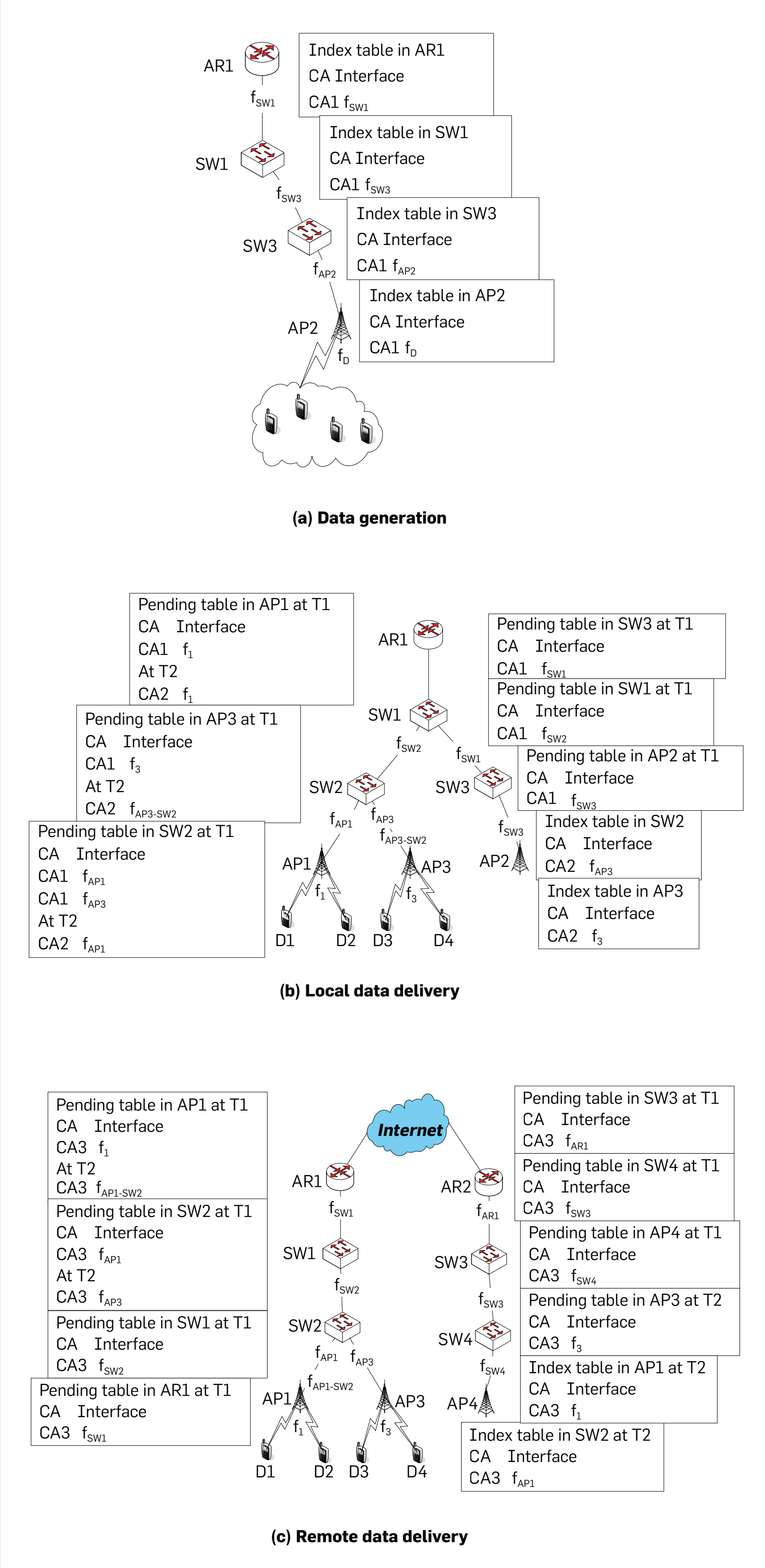
1. AP2 constructs a unicast address, where the GRP and device ID are 0 and the LRP is AP2's 1, and then sends a *Gen-MD* message to associated devices. The source address in *Gen-MD* is the unicast address, and the destination address is CA1.
2. If the device receiving *Gen-MD* can share the resources to generate part(s) of MD1, it returns a *Gen-Ack* message with the generated data.
3. AP2 processes the data in the received *Gen-Ack* to construct MD1 and sends a *Share-MD* message with MD1.
4. The device receiving *Share-MD* selectively stores MD1 and becomes an ITC1 member.

As shown in Figure 2a, AP2 creates ITC1 to generate multimedia data MD1, defined by cloud address CA1. Then, AP2 starts an index entry-generation process so that its upstream infrastructures—including switch SW3, switch SW1, and AR1—establish an index entry for CA1.

After a device becomes an ITC1 member, the payload of a beacon sent by the device includes CA1. If an AP receiving a beacon with CA1 from interface *f* does not have an index entry for CA1, it launches the following index entry-generation process:

1. The AP creates an index entry—where the cloud address is CA1 and the interface is *f*—and sends a *Gen-Index* message with CA1 to its parent.
2. If the parent receiving *Gen-Index* from interface *f*′ does not have an index entry for CA1, it goes to Step 3. The parent abandons *Gen-Index* and goes to Step 4.

Figure 2. Multimedia data delivery.



(a) Data generation

(b) Local data delivery

(c) Remote data delivery

3. The parent creates an index entry for CA1. If the parent is a switch, it forwards *Gen-Index* to its parent and goes to Step 2.

4. The process is complete, as shown in Figure 3a.

Local multimedia data delivery. To achieve request aggregation, each infrastructure stores a pending table, and each pending entry includes the interface and the cloud address. Multimedia data MD1 is identified by cloud address CA1 and is generated by creating ITC1. User D1 is located in subnet S1 with GRP1 and is associated with AP1. If either *Condition* 1 or *Condition* 2 is satisfied, D1 acquires MD1 based on the local multimedia data-retrieval process:

Condition 1. The GRP of CA1 is GRP1 and the LRP is equal to the 1 of AP2 in S1.

Condition 2. The GRP of CA1 is not GRP1, but there is at least one ITC1 member in S1 that can provide MD1.

1. D1 constructs its unicast address, where the GRP is 0 and the LRP is AP1's 1, and then sends a Req-MD message, where the source address is the unicast address and the destination address is CA1.

2. After Req-MD is received from interface *f*, it is dealt with based on the following cases:

▸ **Case 1: An infrastructure receives *Req-MD***

If the infrastructure has the pending entry, where the cloud address is CA1 and interface is *f*, it goes to Step 4. The infrastructure creates a pending entry for CA1. If the infrastructure has only one pending entry for CA1, it goes to Step 3. Otherwise, it goes to Step 4.

▸ **Case 2: A device receives *Req-MD***

If the device can provide MD1, it returns a *Rep-MD* with MD1 and goes to Step 4.

3. The infrastructure deals with *Req-MD* based on the following cases:

▸ **Case 3: There is an index entry for CA1**

The infrastructure forwards *Req-MD* from the interface in the index entry and goes to Step 2.

▸ **Case 4: There is no index entry for CA1**

If the infrastructure is AP2, it generates MD1, and returns *Rep-MD* with MD1 and goes to Step 4. The infrastructure forwards *Req-MD* from the interface maximally matching the LRP of CA1 and goes to Step 2.

4. After the *Rep-MD* is received, it is dealt with based on the following cases:

▸ **Case 5: An infrastructure receives *Rep-MD*.**

For each pending entry for CA1, the infrastructure forwards *Rep-MD* from the interface in the entry, removes the pending entry, and goes to Step 4.

▸ **Case 6: A device receives the *Rep-MD***

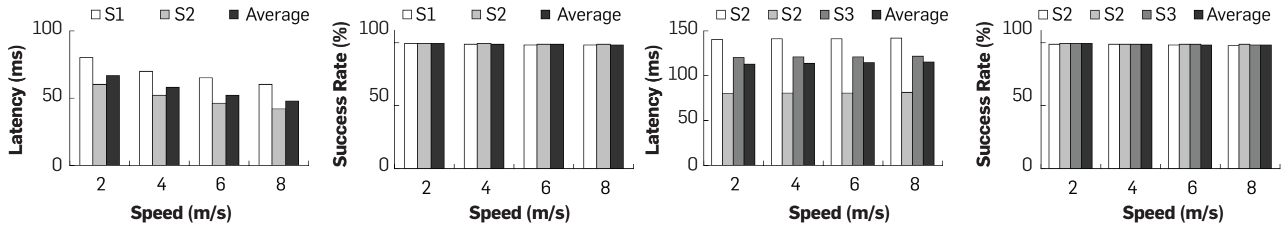
The device selectively stores MD1 and becomes a member.

5. The process is complete, as shown in Figure 4.

Remote multimedia data delivery. User D1 is associated with AP1 and is located in subnet S1, where the AR is AR1 and the GRP is GRP1. In subnet S2, the AR is AR2 and the GRP is GRP2. Multimedia data MD3 is identified by CA3, where the GRP is GRP2 and the LRP is equal to the 1 of AP4 in S2. MD3 is generated by creating ITC3, and D1 acquires MD3 via the following process:

1. D1 constructs its unicast address, where the GRP is 0 and the LRP is AP1's 1, and sends Req-M, where the source address is the unicast address and the destination address is CA3.

Figure 3. Multimedia data-delivery evaluation.



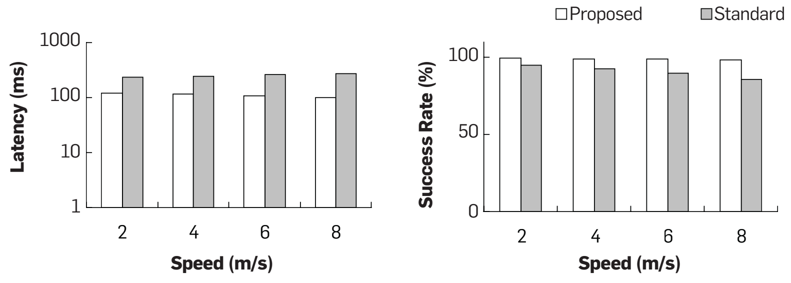
(a) Local data-delivery latency

(b) Local data-delivery success rate

(c) Remote data-delivery latency

(d) Remote data-delivery success rate

Figure 4. Multimedia data-delivery comparison.



(a)

(b)

Table 4. Simulation parameters.

Parameters	Values
Subnets	5
Communication range	100m
Network hierarchy	5
Devices	200
Speed	2-8m/s
MAC	IEEE 802.11, IEEE 802.3
Mobility model	Random walk
Simulation time	1,000s
Runs	20

2. After *Req-MD* is received from interface *f*, it is dealt with based on the following cases:

▸ **Case 1: An infrastructure receives *Req-MD***

If *Condition* 3 is satisfied, it goes to Step 4. If both *Condition* 4 and *Condition* 5 are satisfied, it goes to Step 4. If both *Condition* 4 and *Condition* 6 are satisfied, it goes to Step 3. The infrastructure creates a pending entry for CA3. If the infrastructure has only one request entry for CA3, it goes to Step 3. Otherwise, it goes to Step 4.

Condition 3: The infrastructure has a pending entry, where the cloud address is CA3 and interface is *f*.

Condition 4: The infrastructure is AR2 and *f* is the interface linking with the Internet backbone.

Condition 5: The infrastructure has a pending entry for CA3.

Condition 6: The infrastructure has no pending entry for CA3.

▸ **Case 2: A device receives *Req-MD***

If the device can provide MD3, it returns *Rep-MD* with MD3, and goes to Step 4.

3. The infrastructure deals with the *Req-MD* based on the following cases:

▸ **Case 3: The infrastructure in S1 receives *Req-MD***

If the infrastructure is AR1, it updates the GRP of the source address in *Req-MD* with GRP1, forwards *Req-MD* to the Internet backbone where *Req-MD* is routed to AR2, and goes to Step 2. The infrastructure forwards *Req-MD* to its parent and goes to Step 2.

▸ **Case 4: The infrastructure in S2 receives *Req-MD***

If the infrastructure has an index entry for CA3, it forwards *Req-MD* from the interface in the entry and goes to Step 2. If the infrastructure is not AP4, it forwards *Req-MD* from the interface maximally matching the LRP of CA3 and goes to Step 2. AP4 generates MD3, returns *Rep-MD*, and goes to Step 4.

4. After *MD-Rep* is received, it is dealt with based on the following cases:

▸ **Case 5: An infrastructure receives *Rep-MD***

If AR2 receives *Rep-MD*, it forwards *Rep-MD* to the Internet, where *Rep-MD* is routed to AR1 and goes to Step 4. For each pending entry for CA3, the infrastructure forwards *Rep-MD* from the interface in the entry, removes the pending entry, and goes to Step 4.

One of the most critical challenges facing IoT-related multimedia data is to improve its delivery efficiency by sharing resources with IoT devices.

▸ **Case 6: One device receives the *Rep-MD***

The device selectively caches MD3 and becomes a member.

5. The process is complete, as shown in Figure 4.

Evaluation

The proposed scheme is evaluated in *ns*-2, as shown in Figures 3 and 4. The simulation parameters are shown in Table 4.

As shown in Figure 3a, data-delivery latency decreases with speed. The main reason is the increase in speed augments the area where the ITC members spread, reducing the distance between a user and the nearest member or the intermediate infrastructure performing request aggregation.

In ITC, users acquire data from the nearest members or share data from the intermediate infrastructure, so data-delivery latency in S1 and S2 decreases. In S2, the data is delivered between a user and the intermediate infrastructure performing request aggregation, so data-delivery latency is lower than the one in S1. The average data-delivery delay is the one of multiple users retrieving data in either S1 or S2; it also reduces with the growth in speed. Consequently, the success rates in S1 and S2 slightly grow, as shown in Figure 3b.

In Figure 3c, with the increase in speed, the remote multimedia data latency slightly grows. The main reason is that the retransmission of the lost packets brings the extra delivery latency. In S1, a user retrieves data from a remote ITC member. Since the distances between a user and the local AR and between an ITC member and the remote AR are hardly affected by speed, data-delivery latency tends to be steady. In S2 or S3, a user obtains data from the intermediate infrastructure that performs request aggregation and is located in the local or remote subnet. Similarly, the distances between a user and the local AR and between the local or remote AR and the intermediate infrastructure are hardly impacted by speed, so data-delivery latency in S2 or S3 also tends to be stable. The delay in S2 is minimal and the one in S1 is maximal.

To sum up, the mobility of ITC members has little impact on data-delivery latency due to the following reasons: Firstly, most users acquire data from intermediate infrastructures performing request aggregation in either S2 or S3. Since mobility has no impact on intermediate infrastructures, it has little influence on data delivery latency. Secondly, during mobility, CoA reconfiguration and address binding account for a large proportion of mobility handover latency.[18,22] Since ITC members perform neither CoA reconfiguration nor address binding, delays caused by mobility are avoided. Lastly, users acquire data from the nearest ITC members that have usually completed reconfiguration to system changes. Since the increase in speed weakens the link performance, the remote data-delivery success rate slightly decreases, as shown in Figure 3d.

This proposal is compared with the standard,[11] as shown in Figure 4.

Figure 4a shows the effect of speed on the average delay of users retrieving data locally or remotely. As depicted in Figure 3a, local data-delivery latency decreases with the growth in speed because the distance between a user and the nearest ITC member or the intermediate infrastructure performing request aggregation is shortened. As illustrated in in Figure 3c, remote multimedia data latency slightly grows with the increase in speed. Since multi-hop Ethernet architecture expands the area covered by a local network, in most cases users can retrieve data locally.

Consequently, the average delay of users retrieving data slightly decreases with the growth in speed. Since the growth in speed increases the handover frequency, data-delivery latency and cost in the standard grow. As shown in Figure 4b, network performance degrades as speed increases, so data-delivery success rates in the proposal and the standard decrease. This proposal reduces data-delivery delay by nearly 43.68% and improves data-delivery success rates by nearly 15.12%.

Conclusion

In this article, we propose MDITC which, according to experimental results, reduces data-delivery delays and improves success rates. The main reasons are twofold:

- Users can acquire multimedia data from the nearest members via one data-delivery process.
- ITC members do not need to perform lengthy CoA configuration.

The proposed scheme integrates MC with IoT to construct ITC, and ITC has the following differences from clustering:

- The main objective of clustering is to enhance routing reliability and scalability by reducing node population involved in forwarding and enhancing network stability.[4,20] The main goal of ITC is to enable resource sharing and collaboration among cloud members to create, share, and provide multimedia data.
- The architecture of clustering is hierarchical because each cluster member independently generates data, and a cluster head is responsible for aggregating data generated by cluster members to enhance data-delivery efficiency.[4,20] The architecture of ITC is flat, since each member is equal and has a right to cache and provide multimedia data.
- In clustering, each data-delivery process is independently performed between a source and destination pair. That is, a source node has to retrieve data from a specific node identified by a destination address.[4,20] In ITC, by contrast, multiple users can acquire data from the nearest ITC member via one data-delivery process.
- In clustering, cluster heads need to perform CoA configuration and suffer from data-delivery failures caused by address changes.[4,20] In ITC, each member does not need to be configured with CoA, so data-delivery failures caused by address changes are avoided.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant No. 61872073 and the LiaoNing Revitalization Talents Program under Grant No. XLYC1902010.

References

1. Baccarelli, E., Chiti, F., Cordeschi, N., Fantacci, R., Marabissi, D., Parisi, R., and Uncini, A. Green multimedia wireless sensor networks: Distributed intelligent data fusion, in-network processing, and optimized resource management. *IEEE Wireless Commun. 21*, 4 (2014), 20–26.
2. Balico, L.N., Loureiro, A.A.F., Nakamura, E.F., Barreto, R.S., Pazzi, R.W., and Oliveira, H.A.B.F. Localization prediction in vehicular ad hoc networks. *IEEE Commun. Surveys & Tutorials 20*, 4 (2018), 2784–2803.
3. Bello, O., Zeadally, S., and Badra, M. Network layer inter-operation of device-to-device communication technologies in Internet of Things (IoT). *Ad Hoc Networks 57* (2017), 52–62.
4. Cooper, C., Franklin, D., Ros, M., Safaei, F., and Abolhasan, M.A. Comparative survey of VANET clustering techniques. *IEEE Commun. Surveys and Tutorials 19*, 1 (2017), 657–681.
5. Hennebert, C. and Dos Santos, J. Security protocols and privacy issues into 6LoWPAN stack: A synthesis. *IEEE Internet of Things J. 1*, 5 (2014), 384–398.
6. Hinden, R.M. and Deering, S.E. IP version 6 addressing architecture. *RFC 4291*, 2006.
7. Jiau, M.K., Huang, S.C., Hwang, J.N., and Vasilakos, A.V. Multimedia services in cloud-vehicular networks. *IEEE Intelligent Transportation Systems Mag. 7*, 3 (2015), 62–79.
8. Khan, A.U.R., Othman, M., Madani, S.A., and Khan, S.U. A survey of mobile cloud computing application models. *IEEE Commun. Surveys & Tutorials 16*, 1 (2014), 393–413.
9. Khelifi, H., Luo, S., Nour, B., Moungla, H., Faheem, Y., and Hussain, R. Named data networking in vehicular ad hoc networks: State-of-the-art and challenges. *IEEE Commun. Surveys & Tutorials 22*, 1 (2020), 320–351.
10. Lee, E., Lee, E.K., Gerla M., and Oh, S.Y. Vehicular cloud networking: Architecture and design principles. *IEEE Commun. Mag. 52*, 2 (2014), 148–155.
11. McPherson, D., Oran, D., Thaler, D., and Osterweil, E. Architectural considerations of IP anycast. *RFC 7094*, 2014.
12. Perkins C., Johnson D., and Arkko, J. Mobility support in IPv6. *RFC 6275*, 2011.
13. Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., McCann, J., and Leung, K. A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities. *IEEE Wireless Commun. 20*, 6 (2013), 91–98.
14. Vegni, A.M. and Loscri, V. A survey on vehicular social networks. *IEEE Commun. Surveys & Tutorials 17*, 4 (2015), 2397–2419.
15. Wang, X. Data acquisition in vehicular ad hoc networks. *Commun. ACM 61*, 5 (May 2018), 83–88.
16. Wang, X. and Cai, S. Secure healthcare monitoring framework integrating NDN-based IoT with edge cloud. *Future Generation Computer Systems 112* (2020), 320–329.
17. Wang, X., Cheng, H., and Yao, Y. Addressing-based routing optimization for 6LoWPAN WSN in vehicular scenario. *IEEE Sensors J. 16*, 10 (2016), 3939–3947.
18. Wang, X., Dou, Z., Wang, D., and Sun, Q. Mobility management for 6LoWPAN WSN. *Computer Networks 131* (2018), 110–128.
19. Wang, X. and Li, Y. Content delivery based on vehicular cloud. *IEEE Trans. Vehicular Technology 69*, 2 (2020), 2105–2113.
20. Wang, X. and Qian, H. Constructing a 6LoWPAN wireless sensor network based on a cluster tree. *IEEE Trans. Vehicular Technology 61*, 3 (2012), 1398–1405.
21. Whaiduzzaman, M., Sookhak, M., Gani, A., and Buyya, R. A survey on vehicular cloud computing. *J. Network and Computer Applications 40* (2014), 325–344.
22. Zhao, W. and Xie, J. IMeX: Intergateway cross-layer handoffs in Internet-based infrastructure wireless mesh networks. *IEEE Trans. Mobile Computing 11*, 10 (2012), 1585–1600.
23. Zhu, Y.H., Chi, K., Tian, X., and Leung, V.C. Network coding-based reliable IPv6 packet delivery over IEEE 802.15.4 wireless personal area networks. *IEEE Trans. Vehicular Technology 65*, 4 (2016), 2219–2230.
24. Zhou, J., Leppanen, T., Harjula, E., Ylianttila, M., Ojala, T., Yu, C., and Yang, L.T. Cloudthings: A common architecture for integrating the Internet of Things with cloud computing. In *Proc. of the IEEE 17th Intern. Conf. Computer Supported Cooperative Work in Design* (2013), 651–657.
25. Zhu, Y.H., Qiu, S., Chi, K., and Fang, Y. Latency aware IPv6 packet delivery scheme over IEEE 802.15.4 based battery-free wireless sensor networks. *IEEE Trans. Mobile Computing 16*, 6 (2017), 1691–1704.

Xiaonan Wang is a professor at the Changshu Institute of Technology in Changshu, China.

Xingwei Wang is a professor at Northeastern University in Shenyang, China.

Association for Computing Machinery
STEM
Author's Rights
SIGs
Science
Open
Copyright
Machine Learning
Emerging Technologies
CACM
Digital Library
Articles
Research
Transactions
OpenSurround Service
Gold Open Access
TechNews
Publishing
Articles
eLearn
IT
Author-Izer
CHORUS
3D Printing
Magazines
Cybersecurity
Proceedings
Community
Big Data
Robotics
Networks
Cloud
HCI
AI
OpenTOC
Ubiquity
Journals
IoT
Technology
Access
HPC
XRDS
ACM
Software
Computer Science
Scholarly
Conferences
Computing
Green Open Access
Hybrid Open Access
Communications
Inroads
Computers in Entertainment
Queue

DOI:10.1145/3446912

A future-state architectural strategy designed to support chatbot integration with service delivery systems.

BY ALISTAIR BARROS, RENUKA SINDHGATTA, AND ALIREZA NILI

Scaling Up Chatbots for Corporate Service Delivery Systems

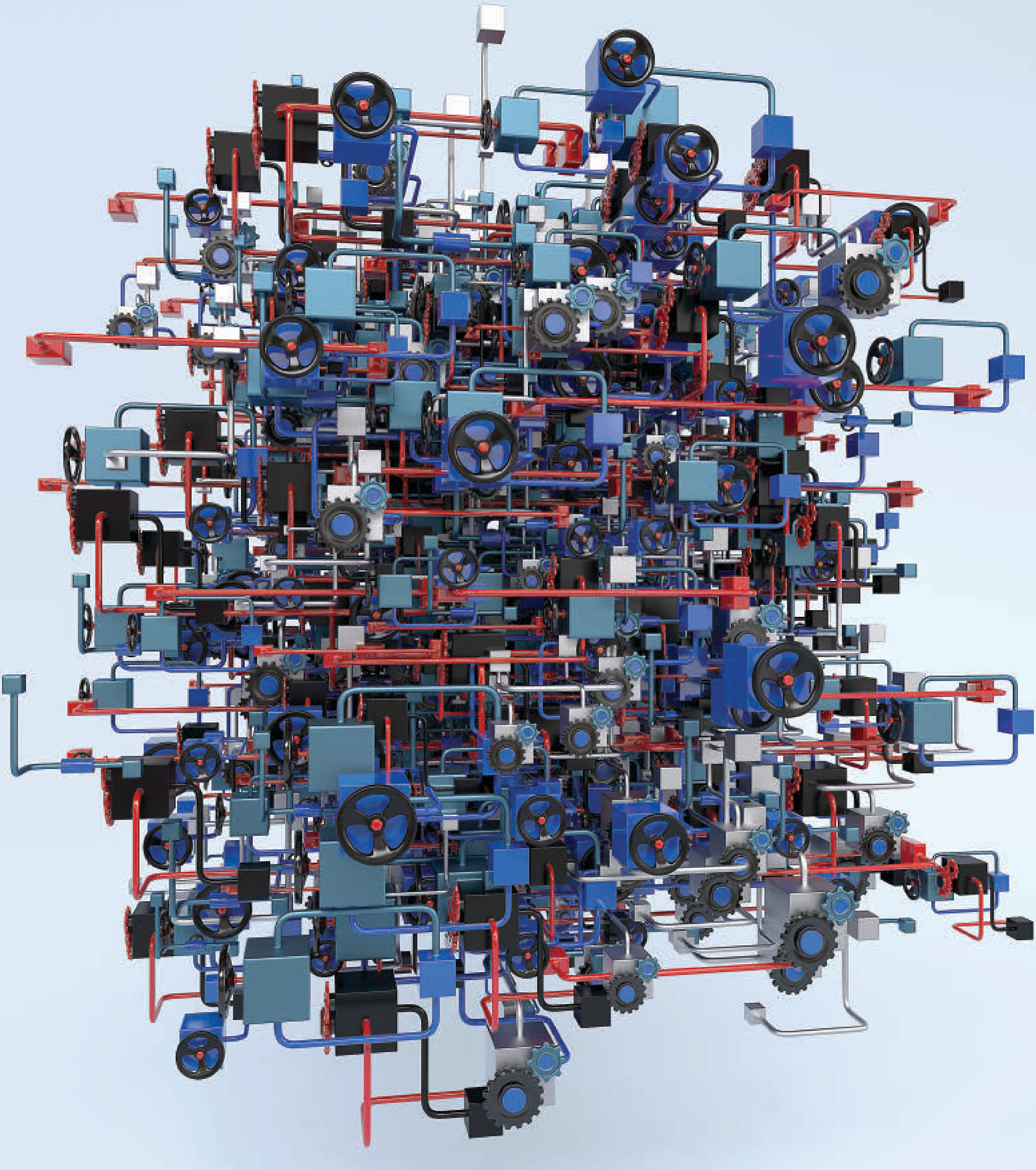
CONVERSATIONAL AGENTS, OR chatbots, providing question-answer assistance on smart devices, have proliferated in recent years and are poised to transform online customer services of corporate sectors.[1,6] Implemented through dialogue management systems, chatbots converse through voice-based and textual dialogue, and harness natural language processing and artificial intelligence to recognize requests, provide responses, and predict user behavior.[5,28] Market analysts concur on current adoption trends and the magnitude of growth and impact of chatbots anticipated in the next five years. According to a report by Grand View Research, for instance, already 45% of users prefer chatbots as the primary point of communications for customer service enquiries, translating into a global 'chatbot' market of $1.23 billion by 2025, at a compounded annual growth rate (CAGR) of 24.3%.[9]

The strategy for conducting conversations using chatbots requires an efficient resolution of two key aspects. First, user queries or automatically perceived needs through user interactions have to be interpreted and mapped into categories, or user intents. This is based on historical processing of queries and needs, and the use of intent classification techniques.[12] Second, conversations must be constructed for specific intents using frame-based dialogue management[2] and neural response generation techniques.[15] In frame-based dialogue management, the chatbot needs to converse with the user to have a fully filled frame (for example, flight information) in which all slot values are provided by the user (for example, airline carrier, departure time, departure location, and arrival location). Inputs on one or more frames results in meeting the user's goal. The

» key insights

- **The key question is how can chatbot capabilities and architecture scale up for corporate services of increasing complexity, sensitivity, delivery processes, and duration?**
- **There are two distinct types of service interactions for delivering complex services via chatbots, including: search-oriented interactions (that is, service discovery: search for available services, rules, locations, and forms) and action-oriented interactions (for example, filling application forms, checking for eligibility and entitlements, negotiations for service offers, and service on-boarding tasks). Chatbots must scale up to cover both types of services.**
- **Chatbots should be retrofitted into a multifaceted contextual-aware service delivery, where dialogue can be effectively anchored, directed and sustained through distinct contexts to meet user goals.**

IMAGE BY MOVIE ELEMENTS



dialogue flow is constructed through an ordered sequence of frames.

As seen through widely adopted examples such as Google Assistant, Apple Siri, and Amazon Alexa, chatbots have been effective for search, recommender, and singleton task capabilities involving bounded contexts, where service knowledge is limited and the conversations are short (typically, not more than a few question-answers). Building upon these capabilities, examples have emerged from retail, financial, government, and other corporate sectors, where chatbots are advancing and moving toward having better capabilities of assisting with in financial, trust and risk sensitive services, traditionally delivered by staff and enterprise systems.[22,23]

These next-generation chatbots are expected to operate in broader user contexts and operating environments of systems, given the extent of data, business rules, and processes involved.[23] Examples of trends underway are as follows. Transport for London's Travel demonstrated how chatbots can be integrated with enterprise systems to find information about services in user contexts, including geospatial and historical user preferences and patterns.[27] Online travel systems such as Expedia, Booking.com, Skyscanner, Cheapflights, and airline companies such as KLM, British Airways, and Icelandir, assist customers in making and changing flights, thus extending the reach of actions associated with dialogue to tasks undertaken by enterprise systems and transactional tasks within them. AI Jim from Lemonade Insurance[26] integrates chatbot dialogue with inputs and outputs of multistep processes by supporting customers in capturing claim details, scheduling repair quotes, and providing instant payments, where possible. Despite these advances, the task-based repertoire of current chatbots are currently limited to single tasks or a small set of tasks where user conversations involve filling in a small number of frames, typically one or two.

This article provides a domain-specific exposition, drawn from social welfare, on how chatbots can scale customer service delivery processes in corporate sectors. Services in social welfare have long-running processes, with multiple steps, stages and user goals, and carry increased data exchange through service interactions compared to the examples above. They involve a mixture of customer, self-assistance interactions and staff-assistance through call centers and service centers and their processes are managed through customer relationship management and other enterprise systems. The nature of the service processes and service dialogue is studied to understand how user conversations associated with different users goals align with, and are interweaved through, long-running delivery processes. As a result of empirical insights, an architectural strategy which supports enhanced tight-coupling of chatbot systems and service delivery systems, is proposed. Through this, we identify the key challenges for developing and integrating dialogue models and service delivery processes, through extensions to machine learning techniques and provide recommendations.

Empirical Method

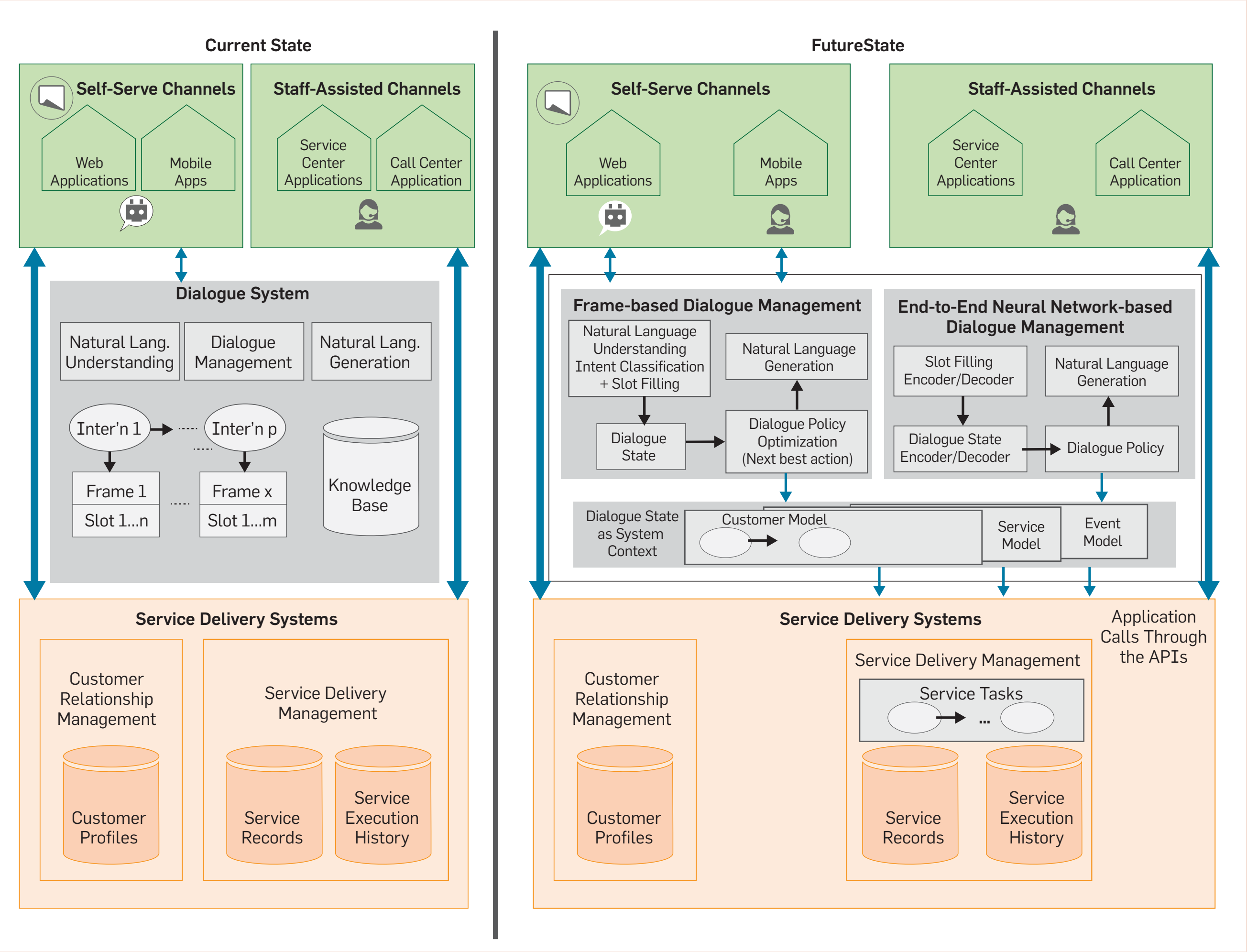
Our study of a complex service domain relates to a large provider of social welfare services in a Commonwealth country. Its services, involving unemployment benefits, family support, medical insurance payments, senior pensions, and many other payments, are among the highest in volume and complexity for customer services. The agency provides call centers, service centers, a website and mobile applications among its channels to support customers across many demographics, needs, and vulnerabilities, to access payment-based and other assistance services. It utilizes whole-government, one-stop shop service centers and websites and third-party partner systems as further channels.

First, we conducted an archival analysis of service delivery process steps and customer journey maps at the same time with an extensive observation of service delivery interactions to understand the nature of service interactions. With regards to the archival analysis, we reviewed the organization's existing service delivery process steps and customer journey maps for the top one hundred (the most requested services) which were designed by the service management, analytics, channel management, IT, and enterprise architecture groups at the organization. These process steps were in the form of operational guidelines and BPMN process models. With regards to the observation of service delivery interactions, we focused on observing the face-to-face interactions between staff and customers at six major offices in three large cities to further our understanding of the service interactions at a large call center of the organization and observed customer interactions with the self-service technologies (for example, website and chatbot) at the areas of the service offices which were providing computer and internet facilities for customers. Overall, we observed over 250 face-to-face service interactions and over 400 service interactions between customers and the self-service technologies.

Next, given that design of digital services often builds upon contributions from multiple areas of expertise,[10,14] we conducted two workshops where experts from several different areas of organization with different areas of expertise participated. In each workshop we discussed the process steps, the challenges and opportunities of digitizing service delivery using intelligent agents, with a focus on complex social welfare services using animated process steps and scenario walkthroughs. Each workshop took three hours with 10 participants including the directors, associate directors, and service support personnel at the service desk management, applications, infrastructure, and business analytics departments within the social welfare agency.

The overall result of these observations and workshops was a deep understanding of the nature of services and service process steps at the organization and different patterns of service interactions from the customer's perspective, which also led to modifying some of the service delivery process steps at the organization. Our insights from these methods are presented later. These insights helped us to move to the next step which is designing protocols for focus groups and individual interviews to understand how the patterns of service interactions need to correspond to the back-end processing of the interactions, enabling us to pro-

Current and future state strategy for integrated chatbots and service delivery systems.


Current State
Self-Serve Channels
Web Applications
Mobile Apps
Staff-Assisted Channels
Service Center Applications
Call Center Application
Dialogue System
Natural Lang. Understanding
Dialogue Management
Natural Lang. Generation
Inter'n 1
Inter'n p
Frame 1
Slot 1...n
Frame x
Slot 1...m
Knowledge Base
Service Delivery Systems
Customer Relationship Management
Customer Profiles
Service Delivery Management
Service Records
Service Execution History
FutureState
Self-Serve Channels
Web Applications
Mobile Apps
Staff-Assisted Channels
Service Center Applications
Call Center Application
Frame-based Dialogue Management
Natural Language Understanding Intent Classification + Slot Filling
Natural Language Generation
Dialogue State
Dialogue Policy Optimization (Next best action)
End-to-End Neural Network-based Dialogue Management
Slot Filling Encoder/Decoder
Natural Language Generation
Dialogue State Encoder/Decoder
Dialogue Policy
Dialogue State as System Context
Customer Model
Service Model
Event Model
Service Delivery Systems
Application Calls Through the APIs
Customer Relationship Management
Customer Profiles
Service Delivery Management
Service Tasks
...
Service Records
Service Execution History

pose recommendations for developing an architectural strategy for integrating chatbots and service delivery systems.

We then conducted six focus groups (on average, 12 people in each focus group, each of which took sixty to ninety minutes) and twelve individual interviews (on average, 60 minutes each interview) were conducted. Participants were different individuals within the same groups of participants who had attended the two workshops (that is, directors, associate directors, and service support personnel at the service desk management, applications, infrastructure, and business analytics departments within the agency). We followed Miles et al.[21] for our thematic analysis of the individual interviews and followed Nili et al.[24] for thematic analysis of focus groups. The researchers in the team discussed the thematic analysis process and the findings iteratively at five meetings, resulting in compete consensus among the researchers on the insights we gained from the data.

The overall insight from individual interviews and focus groups was that the heart of the challenge of chatbot scalability in the corporate sector is integration of customer dialogue processes with task-based service delivery processes (in an interweaved way), which existing chatbots do not provide. In this regard, we also identified major challenges including: the challenge of chatbot systems and service delivery systems be properly integrated, and the need for assisted learning of dialogue models be done coherently with service delivery processes. Drawing upon the insights we gained from these methods, we explain the current state architecture and its technical capabilities and propose recommendations for a future state architectural strategy for integrating chatbots and service delivery systems in the corporate sector, contextualized through the social welfare domain. Moreover, the details of the systems involved (for example, channel applications and service delivery systems) are also shown in the current and future state architecture covered and depicted on the left-hand side of the accompanying figure.

Background on Customer Service Delivery in Social Welfare

Organizations in the finances, energy, government, and other corporate sectors provide a wide range of product-oriented services for diverse customer segments, needs, and circumstances.[4,8] Unlike the delivery of 'everyday' consumables through e-commerce applications, examples such as home loans and insurance claims carry higher risk, longer cycles (days, months, or years) and mutual exchange of personal, agency and third-party data, making them difficult to be delivered seamlessly through a self-managed set of steps online.[18] This is especially the case for social welfare services, sought by individuals, families and other social groups, across a wide spectrum of demographics and temporary or permanent vulnerabilities (for example, unemployed, aged, single parents and child custodians, disabled, and students).[7,23]

Many social welfare service types across different domains are offered through government social welfare providers through different types of payments (for example, aged pension, medical insurance, unemployment benefits, and paid parental leave), concessions (for example, public transport, and health services) and tenancies (for example, public housing).[23] Not only is service matching complex given the range of needs and circumstances of customers that need to be properly elicited and evidenced, but access to these is subject to equity and transparency regulated by legislated policy and business rules.

Services are delivered through a combination of front-office interactions that take place via self-serve and staff-assisted channels and‘back-office processes managed by various service delivery management systems. Many tasks of the service delivery (for example, service matching, filling in claims application forms and tracking claims assessments service access) are available through self-serve channels, reflecting the digital by default strategy advanced by governments in recent years.[19] In addition, service and contact centers are available for customers to engage in *staff-assisted interactions* that are required when they face uncertainties in finding, applying, and accessing, or when they need to be physically present, for example, for meetings or identification processes.[13,23]

The channels are supported by channel applications, for example, service center applications, contact center systems, self-serve Web and mobile applications, and business center applications. The purpose of these applications is to provide context-specific interfaces for customer and service processes presented to customers or staff, for example, presenting coherent service information for different customer cohorts, supporting service discovery or allowing service application forms to be filled in. The applications are tailored to the processes for specific channel contexts and have dedicated capabilities, for example, customer dialogue scripts used for the call center applications. The channel applications interact with enterprise systems, providing the core processes. These consist of: a customer relationship management system (CRM); a service management delivery system (managing service applications, services for customers, service contracts and obligations and interactions with third-party systems); and a payment engine. We observed the both channel applications, CRM and service delivery management systems were adapted from either bespoke or off-the-shelf enterprise systems (for example, SAP ERP), providing an integration framework for these including shared databases.

A key challenge is to effectively understand specific contexts in the requests, from which the relevant direction of dialogue can proceed, from a choice of dialogue directions.

Self-serve channels can support chatbots for specific and well-understood tasks such as service finding and providing indicators for service eligibility as part of online customer inquiries. Conventionally, chatbots have operated through dialogue management systems, which use natural language processing and semantic knowledge of the domains to parse user requests, infer intents and formulate responses.[5] A basic strategy for operating chatbots is to store unstructured question-and-answers through social media, automatically classify these (intent classification), match user queries using these to retrieve recommended responses, and to allow user rating of the recommendations. In this way, internal and external social channels can be harnessed through chatbots. However, more targeted forms of chatbot strategy involved dialogue management, through conversational interactions. This involved identifying key data entities relevant to customer intents, for example, customer identification as frames and customer name, customer identifier, date of birth, and residential address as slots.[11] For multi-interaction conversations,[3] multiple frames/slots and corresponding dialogue request/response interactions can be coordinated in defined sequences, allowing capture, confirmation, action, and interactions to be coordinated.

Multiple chatbot applications apply, for example, for anonymous customers, identified customers and (internal) staff. These applications operate through specific channels, for example, the chatbots for anonymous and identified customers are available through self-serve and mobile applications while chatbots for staff operate on the business center channel application. We observed that these applications are not currently connected to service delivery systems. They only support channel applications, which control interactions with "backend" service delivery systems.

Search-oriented interactions. When customers pose requests at channels as part of the first stage, or potentially other stages, of service delivery, they typically undergo a general service triage step so that their requests are immediately answered or routed to different tasks, systems or channels for further processing. In the simplest case, the needs are obvious in the request. A service or a related aspect is explicitly provided in a request posed by a customer, for example, get study allowance, aged carer's allowance or make an appointment for a social worker assessment. The interactions that follow serve to confirm the service need and communicate details about where to access it and the conditions of eligibility and use.

However, in other cases, the service need is either ambiguous or unknown in the request. This can arise when providers offer a variety of services for similar customer segments and circumstances, for example, the set of welfare benefits for single parents, which vary depending on whether parents are working full/part-time or not, have direct custody of dependents or not, or are biological parents or custodians of dependents. As such, the indications of needs in requests must be 'unpacked' through question-answer dialogue. The key challenge as part of this is to effectively understand specific contexts in the requests, from which the relevant direction of dialogue can proceed, from a choice of dialogue directions—for example, as seen through dialogue scripts used by staff in call and service centers.

There are at least three types of contexts at play when customers present requests in which they are uncertain or unaware of the services relevant for their needs. The first is a customer's personal context (or circumstances). The nature of the need may be couched in terms of customer circumstances or more directly in terms of service needs for circumstances. The second type of context for open-ended requests concerns services, whereby a service or service area is indicated through, or can be inferred from, a request. For example, when the need for unemployment benefits is requested by a customer, the dialogue is directed to confirm the service area or specific service stated by the customer and the circumstances of customers relevant for the service, for example, the customer's age, any existing services being provided and any current, part-time employment, and income level. It may turn out that instead of unemployment benefits, a customer needs either youth allowance, parenting payments, unemployment benefits, or the aged pension, depending on their age and dependent responsibilities. The third context is a *situational context*, such as an event or occurrence, which in some way implicates a customer's circumstances or needs and may trigger the discovery of required services or action in relation to existing services being accessed. As examples, emergencies such as flooding, industrial action or changes to social welfare policy may be reported, requiring triage-style dialogue involving details of the situation, in order to determine specific concerns, incidences, or needs of customers. This can then be used to determine further course of interactions in relation to customer and service contexts.

Action-oriented interactions. Most stages of service delivery, notably those that occur after service needs have been identified, involve actions on the part of both customers and providers. Examples of actions include customer registrations, filling and lodging application forms, assessment checking for eligibility and entitlements, negotiations for service offers, and service on-boarding tasks. While the nature of actions varies widely, we observed common characteristics in terms of their service interactions. Action-oriented interactions can be more cognitively targeted than search-oriented interactions because the services typically known when action are requested, specific information is involved, and the tasks are deterministic, compared to more open-ended nature of service discovery.

Consider a prominent action task, involving the filling in application forms, which follows service needs being identified and customers being registered. The details relate to their circumstances with over 150 data fields for information, such as: customer identity, relationships, dependencies, employment, and details about existing services or claims, and information from third-parties, for example, taxation reference, study course transcripts, medical certificates, and income pay details. When filling in application forms, people may have questions concerning uncertainties in the form, commitments being made, corrective actions and tracking of progress. Examples of questions which trigger interactions include: the meaning of data fields such as dependents, where the applicant does not have custody of children and does not live at the same address as them; the interpretation of defacto partner for a recently formed relationship where both partners live at a parent's residence; and the privacy implications for providing consent to the agency to obtain information about the customer from a third-party organization like the tax agency.

In more complex cases of questions, customers may query the interpretation of service eligibility or entitlement rules related to their circumstances, as required to be captured in the form. Examples of commitments include provision of data that is missing and agreeing to provide consent to the provider to get information from a third-party agency. Although updates and corrections are often made during the claims application process, we observed corrective actions being ones that are formal (for example, making appointments for social worker assessments or contesting decisions for assessment outcomes). Tracking involves targeted and straight-forward questions about the claims processing and assessment progress.

Architectural Strategy for Integrating Chatbots and Service Delivery Systems

Chatbots have been designed for efficient natural language interactions with users and are managed through dialogue management systems and knowledge (frame-based dialogue systems), which are decoupled from service delivery systems—with no data sharing and limited process integration with service delivery systems. This raises uncertainties about how chatbots could be more effectively integrated and exploited through service delivery systems, and how service interactions handled by both chatbots and channel applications could be better integrated, leading to an enriched and a coherent user experience. To open up insights in this regard, we elaborate on key limitations of how chatbots are architected to operate with service delivery systems and, accordingly propose recommendations for a future state architectural strategy. The figure depicts the current and future state architectural strategy, referred to in the discussion of the recommendations.

The architectural strategy is extended from the current state of service delivery operations and systems, as discussed earlier. As such, it addresses the following aspects: channel user interfaces which govern the types of interactions and modalities that can be supported; channel applications which coordinate the resources that can be engaged in service interactions (self-serve or staff-assisted); dialogue management system managing the dialogue-related knowledge using frames and dialogue interactions sequences which refer to the frames; the service delivery systems through which service interactions and tasks are coordinated, notably customer relationship management and service delivery management systems. The aspects of the current state architecture were drawn from our empirical analysis. Using this structure, the future state analysis for scaling up chatbots in an integrated service delivery systems architecture is discussed in the following (noting that the blue arrows between the components are the architectures signify key differences).

Context-Awareness of Dialogue Intents and Conversations in Customer Interactions

The wide-ranging set of requirements that customers seek at different stages of service delivery involve distinct contexts, specific to customers, services, or events. As analyzed through the public social welfare agency, customers typically nominate circumstances when they are uncertain about which services are suitable for them. In this case, service delivery interactions focus on determining service needs based on customer contexts. In cases where customers request services, the focus is on service contexts, while specific aspects of services such as eligibility checks result in interactions that relate to customer contexts.

However, current approaches for dialogue management make no distinction between such distinct contexts. Rather, a singular dialogue context is captured, through relevant frame and slot-filling method, and chatbots determine dialogue intents from customer requests against this. Existing chatbot architectures do not address scenarios that involve multiple goals to be addressed in a single conversation. For example, a customer could start a conversation with a search-oriented interaction such as, “Am I eligible for any payment if I am studying and working for a few hours per week?” the chatbot needs to capture minimal details of the customer to determine one of more payment services the customer may be eligible for and further capture specific details that are required to identify the customer’s eligibility for the payment. These complex scenarios cannot be captured by having singular dialogue graph that is often the design approach of enterprise chatbots using frame-based dialogue management.

For the future state, we recommend that distinct contexts be managed through the dialogue management system and used to direct conversations. As illustrated in the figure’s architectural strategy, a general dialogue context (or dialogue state) can be used to orchestrate overall dialogue flow. It can be used to generally triage intents and coordinate dialogue related to customer, service or event contexts. As an example, when a customer describes the need for financial support related to food, rent, and transport, the customer context can be used to direct the conversation to identify the customer circumstances such as income, dependencies and other circumstances. Once this is determined, the conversation can shift to service eligibility rules, among others, which is supported by specific dialogue knowledge for the service context.

Accordingly, we propose: *chatbots should be retrofitted into a multi-faceted contextual-aware service delivery, where dialogue can be effectively anchored, directed, and sustained through distinct contexts to meet user goals.* The distinct contexts (general, customer, service and event) are implemented through corresponding respective service, customer and event models of the dialogue system.

To support and distinguish search-oriented and action-oriented interactions with the customer, it is useful to maintain different contexts. The customer model would consider frames and slots related to the customer and customer circumstances. By ‘model,’ we refer to the behavioral specification of the dialogue (or dialogue policy). During a search-oriented interaction, the goal of the dialogue would be to capture relevant and optimal information about the customer. For action-oriented interactions the focus would shift to capturing or retrieving information about the service. To support relevant interactions, the dialogue policies or rules of customer, service, or event model are handcrafted as a set of production rules. A production rule is a condition-action pair. The rules consider the conditions pertaining to customer, service or event context. The action could involve seeking more details from the customer, retrieving information of a service, or handing off to a staff agent when the conversation cannot be handled further.

The production rules when chained together implicitly result in dialogue graphs. Each context specific dialogue graph is aligned to the business process tasks and business rules of the service delivery system. The alignment of contextual dialogue graphs to the underlying business process of service delivery system ensures that actions of the chatbot are coherent with the actions directed by the system when the customer uses other channels such as a web-based interface or a staff-assisted enquiry. Designing dialogue rules and constructing the dialogue graphs as a sequence of possible interactions can be time-consuming, expensive and intractable. The interleaving of different contextual models further makes it difficult to choose an optimal rule for dialogue designer. Additionally, the coverage and completeness of production rules and dialogue graphs is difficult to assess given distinct ways of customer interactions. Hence, learning from real-world conversations is imperative.

There are multiple techniques to generating optimal dialogue graphs dynamically: reinforcement learning, automated planning. Optimality could be the cardinality of customer interactions that leads to customer meeting their set goal. In reinforcement learning, the dialogue manager learns from dialogue conversations based on the feedback from the customer or a staff agent supervising the chatbot responses.[15] The feedback collected at the end of the conversation between the chatbot and the customer is used to refine dialogue trees. Based on the feedback rewards or penalties are applied to the refine the conversation. Another approach of generating dialogue tree is to use automated planning. A non-deterministic planner is used to orchestrate the production rules. Given the complexity of service system rules and artifacts, use of learning approaches or automated plan generation approaches to generate dialogue graphs of a large service delivery systems is still an open research challenge and their suitability needs to be ascertained. Hence, we propose they are used with traditional proven and reliable hand-crafted dialogue trees.

Context-awareness of service delivery execution in customer interactions. Service delivery systems manage information related to customers and services required for managing service delivery tasks while dialogue systems provide knowledge contexts related to customers and services required for managing dialogue interactions. However, in the current state, no alignment exists across service delivery and dialogue systems in relation to customer, service and event information. This can lead to redundancies and inconsistencies concerning this information across both systems, for example, the management of customer circumstance attributes related to service eligibility is captured in both frames of dialogue management systems and business rules of service delivery systems. More problematically, dialogue systems do not fully leverage detailed information available in service delivery systems. For example, the conditions for service eligibility, circumstance reporting obligations and penalties of violating obligations, could be provided to users and further clarified through dialogue interactions and information displayed in the user interface information.

If service models and service data in service delivery systems were aligned with dialogue management, the data returned from systems could be updated into concretely specified slots of frames and be used as part of user interactions. The frames and the slots could be filled with the data retrieved from the service systems data for dialogue construction. In addition, dialogue frames and slots could be specified abstractly, with data retrieved from service systems data for dialogue to then be constructed flexibly. For example, a customer identification frame could have slots determined from systems data, such as customer reference number, customer name, residential address, and marital status. This way, frames related to customer identity could be concretely refined (expanded) through identity information in customer profiles managed by service delivery systems. Likewise, details about what services offer (for example, payments and concessions), their claim forms, conditions of access, among others can be linked directly through an abstract-concrete data alignment relationship between dialogue and service delivery systems. Such relationships would also allow for dialogue models to automatically support changes to data at the service delivery systems level.

If service models and service data in service delivery systems were aligned with dialogue management, the data returned from systems could be updated into concretely specified slots of frames and be used as part of user interactions.

We therefore propose: *chatbots should be integrated with service delivery processes, such that dialogue interactions and systems interactions are effectively integrated.*

As indicated in the figure, to support transparency of data at the service delivery systems level for alignment with dialogue models of dialogue management systems, we recommend support for dedicated masters data repositories for customers profiles and service catalogues.[23] For example, service catalogues store information such as service delivery processes, service tasks and their inputs/outputs, and service level agreements, can be fully referred to and aligned with service dialogue models. Access to execution state of services can further enrich service dialogue contexts.

Supporting a tight integration of the chatbot interactions with the service delivery system requires the frames and slots need to be aligned to the artifacts of the service delivery system. The dialogue actions further need to invoke relevant service operations with the inputs as slot values and interpret the results of the service delivery system into responses. The dialogue production rules that lead to calling the service operations with relevant input values of slots need to be hand-crafted by the dialogue designer. This can be time-consuming and expensive to specify and maintain for all possible customer interactions, especially in scenarios where there are hundreds of services operations with different input parameters. Often, it is feasible for the dialogue designers to account for the most common dialogue interactions rules where the integration with one or more service operations is specified.

To alleviate the cost and effort of building the dialogue actions for all scenarios, deep neural networks have been explored to train dialogue manager. Recent studies have explored the use of end-to-end training of task-oriented dialogues. Here, a chatbot is trained with the dialogue conversations that allows it to learn all tasks in the dialogue pipeline: slot filling, dialogue state management, and the dialogue actions including making calls to specific service operations.[15] Although these approaches have gained research attention, their applicability for goal-oriented chatbots comprising of multiples frames and several slots is still a challenge. End-to-end neural network-based models need for large amounts of hand-curated data before they start generating a coherent and fluent response. They further provide no control on the actions chosen by the chatbot.[20] Their advantage, however, is the ability to memorize and learn from past conversations and can be suitable for learning the actions when new slot values are encountered, the sequence or the sets of services operations that can be called, or when certain exceptions occur in the dialogue interactions not recorded dialogue designer.

To alleviate the cost and effort of building the dialogue actions for all scenarios, deep neural networks have been explored to train the dialogue manager.

Accordingly, we propose: *chatbots should use the hybrid approach of using traditional frame-based method in conjunction with end-to-end neural network-based learning to support reliable common dialogue interactions and distinct staff-assisted customer interactions.*

In the hybrid approach, scenarios, where the frame-based dialogue management fails to provide suitable response indicated by long conversations or user requesting staff-assistance, the dialogue is first directed to a staff agent to continue, and close the conversation. The staff agent responds to the customer and invokes relevant calls to the service delivery system. The conversational data containing the responses and the invocations to the service operations is used to train the end-to-end neural network-based dialogue manager. The model is trained to identify slots, provide the relevant response, invoke the relevant service operations of the service delivery system. The chatbot learns from the newly accumulated data.

The approach benefits from the use of an optimal subset of real-world conversations to teach the chatbot.[29] Hence, the training data only contains conversations that the chatbot needs to learn for future interactions.[16] Updates to production rules, resulting from changes in government policies or service delivery rules, are updated by dialogue designers. The end-to-end data-driven model would re-learn from the staff assisted conversations when such updates are made. For example, if policies related to an age pension changes and the dialogue rules are updated, the neural network model is re-trained with the new dialogue conversations related to age pension. The hybrid approach allows the chatbot to use the reliable and traditional frame-based approach for a standard set of scenarios, and end-to-end dialogue learning with machine teaching by staff assisted conversations for other alternate dialogue interactions.

To support the hybrid approach to learning from human-assisted teaching and feedback, the conversation from a chatbot can be routed to a contact center. Similarly, when customers are assisted through calls, they should be automatically routed to other staff, systems or even chatbots, depending on how manageable the required action is. A future-state service delivery system should provide pro-active processes where the right forms of channels and interactions are available and pre-empted where possible, depending on in-situ customer needs.

Thus, we propose: *chatbots should operate as part of a coherent, multi-channel service delivery process aimed at a consistent, predictable and proactive interactions.* This means the delivery processes need to be carefully orchestrated to use delivery resources including staff in different delivery roles and chabots, in a way that is consistent and avoids delivery latencies for customers.

Our proposed architectural strategy supports service delivery via a chatbot that requires dialogue graphs or the models to consider different contextual elements of the social service domain and support distinct customer interactions. The interactions are aligned with the underlying service delivery process. As interleaving these interactions into a tractable dialogue graph is difficult, data assisted learning methods can be used to generate the optimal dialogue graphs. Further, a tight integration of the chatbot with the service delivery system is required to provide coherent interactions to the customer, consistent with interactions via other channels. To provide such a tight integration for all dialogue interactions, we suggest a hybrid approach that combines frame-based dialogue management with an end-to-end neural network-based machine taught dialogue manager.

Conclusion

In this article, we presented important insights of corporate service delivery, drawn from the public sector, in order to elicit key ways in which chatbot systems can be applied and extended in this setting. Our contributions were twofold. Firstly, we provided an exposition of multi-channel service delivery systems involving both self-serve and staff assisted interactions and the different types of interactions that customers encounter across different stages of service delivery. In particular, this shed light on the different systems contexts that chatbot systems need to couple with (for example, front-end channel applications and service delivery systems) and the specific dialogue management foci that are in play for different service delivery interactions.

Building upon these insights, our second area of contribution related to four key recommendations for chatbot integration with service delivery systems are: dialogue management for customers, services and events, for effective targeting of dialogues for specific service delivery tasks; alignment and integration of dialogue management with service delivery processes and data; hybrid approach that uses traditional frame-based dialogue management in conjunction with human-teaching and feedback based end-to-end neural network-based dialogue management; and multi-channels allowing chatbot and staff to co-opted for assistance tasks. Future work can focus on detailed design, development and experimental analysis of these recommendations with corporate systems not only in the public sector but also in any sector featuring complex customer services.

References

1. Androutsopoulou, A., Karacapilidis, N., Loukis, E. and Charalabidis, Y. Transforming the communication between citizens and government through AI-guided chatbots. *Gov. Info. Q. 36*, 2 (2019), 358–367.
2. Bobrow, D.G., Kaplan, R.M., Kay, M., Norman, D.A., Thompson, H. and Winograd, T. GUS, a frame-driven dialog system. *Artificial Intelligence 8*, 2 (1977), 155–173.
3. Bohus, D. and Rudnicky, A.I. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language 23*, 3 (2009), 332–361.
4. Candello, H., Pinhanez, C., Millen, D. and Andrade, B.D. Shaping the experience of a cognitive investment adviser. In *Proceedings of the Intern. Conf. Design, User Experience, and Usability* (2017), 594–613. Springer, Cham.
5. Demirkan, H., Bess, C., Spohrer, J., Rayes, A., Allen, D. and Moghaddam, Y. Innovations with smart service systems: Analytics, big data, cognitive assistance, and the internet of everything. *Commun. Assoc. Information Systems, 37*, (2015), 733–752.
6. Engin, Z. and Treleaven, P. Algorithmic government: Automating public services and supporting civil servants in using data science technologies. *The Computer J. 62*, 3 (2019), 448–460.
7. Falco, E. and Kleinhans, R. Beyond technology: Identifying local government challenges for using digital platforms for citizen engagement. *Intern. J. Information Management 40* (2018), 17–20.
8. Fargnoli, M., Costantino, F., Di Gravio, G. and Tronci, M. Product service-systems implementation: A customized framework to enhance sustainability and customer satisfaction. *J. Cleaner Production 188* (2018), 387–401.
9. Grand View Research. Chatbot market size to reach $1.25 billion by 2025; https://www.grandviewresearch.com/press-release/global-chatbot-market
10. Grenha Teixeira, J., Patrício, L., Huang, K.H., Fisk, R.P., Nóbrega, L. and Constantine, L. The MINDS method: integrating management and interaction design perspectives for service design. *J. Service Research 20*, 3 (2017), 240–258.
11. Hakkani-Tür, D., Tür, G., Celikyilmaz, A., Chen, Y.N., Gao, J., Deng, L. and Wang, Y.Y. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. *Interspeech* (2016) 715–719.
12. Jung, S., Lee, C., Kim, K. and Lee, G. G. Hybrid approach to user intention modeling for dialog simulation. In *Proceedings of the ACL-IJCNLP 2009 Conf. Short Papers*, 17–20. Assoc. Computational Linguistics.
13. Kirkpatrick, K. AI in contact centers. *Commun. ACM 60*, 8 (Aug. 2017), 18–19.
14. Kristensson, P., Parasuraman, A., McColl-Kennedy, J. R., Edvardsson, B. and Colurcio, M. Linking service design to value creation and service research. *J. Service Management 27*, 1 (2016), 21–29.
15. Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J. and Jurafsky, D. Deep reinforcement learning for dialogue generation. EMNLP 2016, 1192-1202
16. Lindvall, M., Jesper M. and Löwgren, J. The importance of UX for machine teaching. 2018 AAAI Spring Symposium Series.
17. Liu, B., Tur, G., Hakkani-Tur, D., Shah, P. and Heck, L. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. Naacl-hlt, 2018, 2060–2069.
18. Makasi, T., Nili, A., Desouza, K. and Tate, M. Chatbot-mediated public service delivery: A public service value-based framework. *First Monday 25*, 12 (2020).
19. Mergel, I. Digital service teams in government. *Gov. Inform. Q*, (2019), 101389.
20. Metz, R. Microsoft's neo-Nazi sexbot was a great lesson for makers of AI assistants. *MIT Technology Review* (2018); https://goo.gl/TF8DQx.
21. Miles M.B., Huberman, A.M. and Saldana, J. Qualitative Data Analysis, A Methods Sourcebook. SAGE Publications, Thousands Oaks, CA, 2014.
22. News Microsoft.com. Artificial intelligence transforms even the most human services; https://news.microsoft.com/en-au/features/artificial-intelligence-transforms-even-human-services/
23. Nili, A., Barros, A. and Tate, M. The public sector can teach us a lot about digitizing customer service. *MIT Sloan Management Review 60*, 2 (2019), 84–87.
24. Nili, A., Tate, M. and Johnstone, D. A framework and approach for analysis of focus group data in information systems research. *Commun. Asso. for Information Systems 40*, (2017),.
25. Nott, G. Cognitive virtual agent Amelia debuts on NSW Govdc. Computerworld; https://www.computerworld.com.au/article/629831/cognitive-virtual-agent-amelia-debuts-nsw-govdc/
26. Schreiber, D. Lemonade sets a new world record. Lemonade Renters & Home Insurance | protect the stuff you love; www.lemonade.com/blog/lemonade-sets-new-world-record/
27. Transport for London. TfL launches new social media 'TravelBot, (2017); https://tfl.gov.uk/info-for/media/press-releases/2017/june/tfl-launches-new-social-media-travelb
28. Van Doorn, J., Mende, M., Noble, S.M., Hulland, J., Ostrom, A. L., Grewal, D., and Petersen, J.A. Domo arigato Mr. Roboto: Emergence of automated social presence in organizational frontlines and customers' service experiences. *J. Service Research 20*, 1 (2017), 43–58.
29. Zhu, X. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *Proceedings of the 29th AAAI Conf.* 2015.

Alistair Barros is a professor of service sciences in the School of Information Systems, Faculty of Science, at Queensland University of Technology, Brisbane, Australia.

Renuka Sindhgatta is a lecturer in service sciences in the School of Information Systems, Faculty of Science, at Queensland University of Technology, Brisbane, Australia.

Alireza Nili is a lecturer in service sciences in the School of Information Systems, Faculty of Science, at Queensland University of Technology, Brisbane, Australia.

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

DOI:10.1145/3469279

Collaborations between two communities have unearthed a sweet spot for future programming efforts.

BY SARAH E. CHASINS, ELENA L. GLASSMAN, AND JOSHUA SUNSHINE

PL and HCI: Better Together

IN THE LAST 10 years, the computer science (CS) community has developed novel programming systems that are transforming our world. Data journalists are wielding new programming tools to enrich many major media outlets with interactive visualizations. Microsoft Excel, the primary data programming environment for hundreds of millions of people, now comes with a program synthesis tool that helps users clean and transform their data, sparing them from writing painful spreadsheet formulas. These projects share an important common factor: they succeed because they make programming easier. They demonstrate the power of combining human-computer interaction (HCI) and programming languages (PL). We organized the PLATEAU workshop, part of a growing community that tackles work at this intersection. Here are the research problems that led us to this hybrid field:

PL → HCI Josh Sunshine began his career as a PL researcher working on the design of the Plaid language. He was drawn to HCI techniques when he tried to run a user study of Plaid. He found that users failed to complete even simple tasks—the language was just too difficult. His language design work since then has relied heavily on formative HCI methods like contextual inquiry and natural programming elicitation.[24] The end result is usable languages and successful users.

HCI → PL Elena Glassman was working toward her Ph.D. in HCI when she developed a tool for visualizing student code to help teachers see where student solutions overlap and where they differ. For each new programming assignment, she had to build a new analyzer, which was tedious, time-consuming, and required her expertise as the tool's designer. Later, a colleague introduced her to program synthesis (PL). She realized that she could equip her tool with an example-based synthesizer so that teachers could author custom analyzers for their own assignments.

HCI ↔ PL In talking to social scientists about their technical challenges, Sarah Chasins learned that Web scrap ing was a big obstacle to obtaining data for their research. She began iteratively developing a Programming-By-Demonstration (PBD) Web automation tool with its own custom language to meet the social scientists' needs. Over the course of the work, each individual subproblem demanded both PL and HCI. For example, combined PL and HCI approaches put parallel scraping in reach. To make her new parallelization construct usable, Sarah phrased the problem in terms of a familiar task (HCI); to implement it, she compiled to parallel programming primitives (PL).

The rising tide of PL+HCI research arrives as we observe a few key trends. First, advances in language engineering support make it easier for anyone to develop new languages. Second, methodological and theoretical innovations in HCI make it easier than ever to

IMAGE BY ANDREY POPOV

study humans doing rich and complex computing tasks like programming, which lets us apply HCI techniques to language development. Third, broad and diverse new audiences are seeking automation.

- On the basis of these trends and our own knowledge of the field, we have identified a few key directions, summarized in the accompanying figure, that HCI and PL experts should explore to take full advantage of the combined power of HCI and PL:
- HCI practitioners can benefit from new tools that make it easy to build domain-specific and general-purpose programming languages. However, users need help writing safe and correct programs, and PL techniques can help. Finally, users may not always want to write code directly; to balance ease-of-use with the power and flexibility of programming, our interfaces should give users multiple ways to express their intent.
- PL practitioners can use need-finding techniques to identify high-impact problem domains or programmers' current and future pain points. They can make better design decisions via cognitive and behavioral theory where those theories are available. Where theory is not available, they can make better design decisions by leveraging iterative design cycles that incorporate user feedback.

The remainder of this article describes misconceptions that have inhibited work at the intersection of these two subfields, how each subfield can benefit from the other, and the kinds of dramatic research successes that result from successful PL+HCI unions. Finally, we discuss the directions for future work and how they will deliver important new languages and interfaces. Our key takeaways are summarized in Table 1.

Table 1. Key directions for HCI experts looking to integrate PL practices and PL experts looking to integrate HCI practices.

To interface designers:	To language designers:
Give users PLs	Pick good problems,
But help them use PLs responsibly,	Develop theories of human capabilities and behavior,
And don't expect code alone.	And get frequent user feedback when you lack theory.

What Are We Talking About Here?

HCI is concerned with creating new ways of interacting with computers, using computers to enhance human-to-human interaction, and studying how existing systems affect individuals and society. PL is concerned with the theory, design, and implementation of programming languages, program analyses, and program transformations. This article is devoted to work that combines PL and HCI techniques to advance the goals of either field. However, many other fields and subfields consider how we can use programming languages to serve humans. We provide pointers to a few of the most relevant fields here:

Software engineering. PL, HCI, and software engineering (SE) have a key overlapping interest: getting computers to do what we want. Modern PL-HCI research often ends up at SE venues as the closest fits in today's conference landscape, but much of the work at the PL-HCI border does not fit naturally within SE's scope of interest. In particular, SE primarily focuses on professional software engineers, and many PL-HCI works are aimed at other audiences.

Psychology of programming. The psychology of programming (PoP) community has a long history of studying everything from the cognitive work of individual programmers to how they deal with large codebases to how they work in engineering teams. (See Blackwell et al.[4] for an overview of how the field evolved from the late 1960s into the present.) This critically important work has unfortunately had limited impact on mainstream PL.[14,29] This article advocates for more work that crosses the boundaries between PL and HCI, but we hope readers will recognize that many of the same arguments apply for crossing the disciplinary boundaries between PL and PoP.

Computer science education. CS education (CSEd) research, because it focuses on interventions that make it easier for novices to learn CS, often involves forays into programming languages and tools. For example, consider the Alice[9] and Scratch[32] projects, which set off the modern interest in block-based editors and structure editors. This style of CSEd work often advances HCI goals, but like SE it emphasizes a particular audience—novice programmers who want to learn CS—and a particular set of goals.

End-user programming. This subfield has a long, rich history and, like SE and CSEd, an emphasis on a particular subset of users. In this case, the target audience excludes professional software developers and includes users in other domains who need computational support for their goals. The body of work in end-user programming (EUP) extends back to *"A Small Matter of Programming*,"[26] and it remains an active domain.[18,21]

Some of the work in the intersection of PL and HCI fits neatly into these related communities, and some of it does not. Certainly, the new collaborations between PL and HCI researchers are not the first efforts to tackle the goals laid out in this article—see for example Kay,[17] Myers et al.,[24] and Pane et al.,[28] in addition to the works cited earlier. However, this article highlights the work we can do when we bring HCI and PL techniques together at the same table that we cannot do in isolation. Substantive collaboration across these fields—and with SE, PoP, CSEd, and EUP!—offers a promising route toward usable languages and powerful interfaces. We are excited to see what these subfields can do together as they begin a fresh wave of cross-disciplinary collaborations.

Common PL+HCI Misconceptions

We begin by addressing a few of the misconceptions that sometimes stand in the way of PL-curious HCI researchers and HCI-curious PL researchers.

Misconception: PL doesn't care about people. This common misconception reflects the idea that PL researchers only care about logic and proofs, or only about compiler performance—not about people. In fact, much of the field's work on language features and developer tools has been driven by an interest in the user experience. Although work that brings HCI techniques to bear on PL problems is still fairly young, the interest in making programming languages and tools more usable is longstanding. For instance, the entire program synthesis community sprang up around the idea that some programming tasks are easier for machines than for humans and should be offloaded to specialized program generation tools.

Misconception: PL just makes new general-purpose languages. Another common misconception we hear about PL research is that it is all about creating new general-purpose programming languages. Since the most popular languages are at least 20 years old, they ask, has the programming languages community had any impact? Some even argue that *PL research is stagnant*. In fact, only a tiny fraction of papers at programming languages conferences (<1%) discuss new general-purpose language designs. Most research investigates novel implementation techniques, program analyses, verification and synthesis techniques, tools to support language engineers, and new language features. Popular languages are taking advantage of that work as they evolve. For example, the tremendous performance improvements in JavaScript engines were built on just-in-time compilation techniques developed by PL researchers.

Misconception: PL can't benefit from human factors research. Some researchers contend that HCI methods are not applicable to programming languages because they are complex learned artifacts. The benefit of new language constructs may only come after substantial education and experience, and they believe HCI methods are limited to tools for end users and novices. In fact, HCI methods have been used to study everything from nuclear power plant control systems to augmented reality and flight control systems. Another misconception we hear from PL practitioners is that HCI methods are only useful for surface concerns like fonts, colors, and layout. HCI is not, and has never been, restricted to purely surface-level or visual features. It can encompass everything from the user's mental models as they learn a new tool to the class of information passed between user and tool to the set of abstractions that lets them express their needs.

Misconception: HCI is all about evaluation. Another common one: HCI is just about evaluating interfaces via users studies. HCI has never had a narrow focus on purely evaluative work. Over the course of its 40-year history, the HCI community has developed methods for engaging users in the entire iterative design cycle. At the beginning of the design process, need-finding and formative studies offer low-cost ways to identify existing pain points and anticipate usability problems early. Throughout the design process, a vast space of HCI methods—for example, heuristic evaluation, cognitive walkthroughs, "Wizard of Oz" studies, rapid prototyping, think-aloud studies, natural program elicitation—can give developers more information to make better-informed design decisions.

Misconception: HCI is just implementing what users *say* they want. Another misconception, about formative studies in particular, is that using HCI during the design process means simply implementing what users say they want. This is the Steve Jobs, Henry Ford "If I had asked people what they wanted, they would have said faster horses" concern. Conducting formative studies does not have to mean asking users what they want and then delivering what they request. Some need-finding research involves listening to user requests; but a great deal is focused on observing users' behavior in a given context, even testing hypotheses about their behavior. Via iterative design of prototypes, researchers can expose potential users to multiple hypothetical futures they would never have requested and solicit feedback. These strategies empower potential users to shape technologies that have never existed before, putting those technologies on track to be useful and usable.

Misconception: Doing HCI is too hard. This misconception usually revolves around either the idea that user studies need to include dozens of people to be valid or the idea that the IRB approval process is grueling. In fact, even studies with small numbers of participants can contribute important insights and evaluations. The key is to include enough participants to provide evidence of the claims we want to make. If we build a tool for rare domain experts, we may run a study with five people that focuses on qualitative insights. Or, if we expect our tool to have a large effect on outcomes, enrolling 10 participants in a within-subjects study may be enough to show meaningful differences between their experiences using the experimental and control interfaces. If the class of potential participants is large, we may run a medium-sized lab study of 20–25 people or a large online study that focuses on quantitative insights. Finally, if we want a lightweight way to check our ideas during a design process, it can be enough to watch over a friend's shoulder and hear them talk through using our prototype; this informal, n=1 'study' can be enough to reveal critical design flaws or spark new ideas!

IRB processes vary by institution, but most have official low-risk ('exempt') submission categories for which the approval process is lightweight and fast—and a vast majority of PL+HCI studies fall into these categories. Colleagues who do exempt human subjects research are a great resource for institution-specific advice about IRB processes.

HCI and PL: A Two-Way Street

While PL and HCI have had relatively little cross-over in terms of collaborations and shared literature, each community has developed techniques that can help researchers in the other field. Here we describe a few concrete ways that HCI concepts and techniques can improve PL outcomes; PL concepts and techniques can improve HCI outcomes; and PL and HCI researchers can integrate their complementary expertise to advance goals that matter in both communities.

PL → HCI: The power of PL-backed interfaces. Languages are powerful interfaces for communicating with computers. Unlike typical menu- and button-based interfaces, languages are compositional: they provide a set of primitives and a means of combination, empowering users to create new primitives out of existing ones. If they are Turing-complete, they can describe any computable function. Even a non-Turing-complete language can express an infinite space of functions. While both GUIs and languages are often designed around making it easy for users to say common things, a language empowers users to say uncommon things too. Users can even interact with standard interface elements instead of code and still wield the power of a programming language, if the interface automatically generates programs for the user (for example, via program synthesis). These PL-backed interfaces can help us realize the vision for powerful interfaces advanced by Shneiderman in his seminal "Direct Manipulation."[34] In particular, the expressive power of programming languages can elicit the "desire to explore more powerful aspects of the system" that is often lacking from GUIs.

Building PLs can be easy. Language engineering has become easier with the development of new, easier language implementation support tools like language workbenches and parser generators. Designing task-relevant abstractions and instantiating them in a domain-specific language now takes only minimal training. For HCI work that benefits from the power and flexibility of a language, these new PL tools can support interface and system designers in making new languages, abstractions, and domain-specific languages.

Using PLs can be easy. PL advances like synthesis and modern retargeting let us ask users for a little work and get a lot in return. With techniques like programming by demonstration and programming by example, users can provide non-code specifications (for example, input-output pairs) and get a program in return. With retargeting approaches, we can take programs originally intended for one purpose and reuse them to create new artifacts.

Using PLs correctly. PL, like all other areas of computer science, brings technical capabilities to the table that can help address HCI concerns. For example, the PL community has developed verification techniques to the point that they can check more than simple, low-level properties; they can verify functional correctness, safety, security, accessibility, even adherence to social norms,[30] and other properties that matter to the HCI community. Many PL techniques, such as program analysis, bug fixing, and verification, can offload tasks to the machine, reducing the cognitive load of human operators and designers. These sophisticated techniques are already being applied in professional programming environments. As more end users begin automating tasks, we see opportunities to apply these same techniques to their computer and robot interactions.

HCI → PL: Iterative, user-centered design. User-centered design focuses us on assessing the usefulness and usability of our languages and tools throughout the design process—not just in a final evaluative step. Need-finding studies let designers identify key needs, stumbling blocks, and challenges before the design process even begins. When we tackle needs that have already been validated via need-finding studies, we have good reason to believe our languages or tools can solve real users' problems. Formative studies throughout the design process let us progress steadily toward usability during the language or tool building process. Soliciting feedback from users at multiple points in the design process means we are less likely to end up with user-antagonistic tools at the end, when we have already sunk years of time, energy, and engineering into them.

Theories of human cognition and behavior, design heuristics. Making every design decision based on direct user observation would be expensive, time-consuming, and impractical. Design heuristics, for example, Green et al.[12] describe elements of interactive systems that designers have found over and over are critical to usability, like the visibility of the system's status or the ability to 'undo' an action. Theories of human cognition and behavior make predictions about what users will, will not, and cannot do in any system we construct for them. Like design heuristics, theory predictions are guidelines to narrow our design space and form expectations that may or may not be violated when the user and the system ultimately interact. Some programming tools are already designed on the basis of programming-specific behavioral theory, for example: understanding how programmers backtrack enabled researchers to develop selective undo in Integrated Developer Environments (IDEs).[37] Developing more and deeper theory can pay dividends for the entire programming languages community.

Can We Simply Stage HCI and PL Expertise?

Design choices interact. We cannot ask the PL expert to design the abstractions, deliver them to the HCI expert for a second pass, and expect the optimal design as a result. These choices interact. Design decisions that we make to improve learnability have implications for how we achieve modularity, and vice versa.

To achieve tools and languages that meet the goals of both subfields, we need HCI and PL expertise at the same table. It is not enough to know what users want unless we can make a language, synthesizer, or programming environment that delivers it. Likewise, making a new language, synthesizer, or environment will not advance our goals unless the new artifact meets real user needs.

Evaluation: Beyond user studies. Many programming systems developers are interested in making claims about their advantages for users. HCI has developed many methods for evaluating these claims. These include traditional user studies in the lab but also low-cost heuristic methods, deeper long-term case studies,[35] and rigorous analysis of field data like user logs. To back up the strongest and most exciting claims, we may need multiple evaluation methods—for example, user logs to acquire large-scale data and a lab study to understand the otherwise contextless log statistics. Readers interested in learning more about the diverse set of human-factors evaluations we can apply to programming interactions are encouraged to read Myers et al.'s excellent essay, "Programmers Are Users Too: Human-Centered Methods for Improving Programming Tools."[25]

HCI ↔ PL: In a few domains, both HCI and PL currently advance the state of the art, although these advances are not always shared across the disciplinary divide. In these domains, we hope to engender a richer culture of cross pollination, in the belief that both communities can benefit from the findings of the other.

Abstraction design. Each subfield has its own culture and design goals. They both contribute to features that matter to users, but often to different sets of features. The PL community has deep expertise in developing modular, reusable abstractions. The HCI community has deep expertise in developing abstractions that are easy to learn or match the existing mental models of their target users. With rich histories of abstraction design across both fields, a union of these forms of expertise holds the promise of delivering useful, usable, and powerful abstractions.

Interactive and non-interactive environments. Programming environments that demand a mix of interactive and non-interactive modes are common in the real world. For example, programmers draft code in a relatively non- interactive text window, then refactor the same code via an interaction with their editor of choice. HCI has developed rich theories of interactive computing environments, while PL has long studied how to shape languages to produce good experiences for non-interactive programming settings. For modern programming systems that demand both modes, it is the combination of both PL and HCI expertise that offers the guidance we need (also, see the sidebar "Can We Simply Stage HCI and PL Expertise?").

What It Looks Like When It Goes Well

Bringing HCI and PL expertise together at the same table lets us meet challenges that neither field can accomplish alone. This section highlights how the union of these fields equips us to bring programming to new audiences, improve the programming experience for novices and experts alike, and fine-tune the division of labor between human and machine.

PL+HCI brings the power of programming to new audiences. Non-coders want programs. They want programs that collect, analyze, and visualize data; programs to control their own phones, computers, and other devices; programs to eliminate boring, repetitive tasks. But for now, there is still a gap between the programming skills of the average adult and the skills required to write the programs they want or need.

The union of PL and HCI techniques can close that gap by dramatically reducing the programming skills required to automate important tasks. Modern domain-specific languages put simple but useful programs in reach in domains like building websites[36] or automating smart home actions (IFTTT). With HCI, we can learn the kinds of inputs users are willing and able to provide; with PL, we can invent techniques that turn those inputs into the programs users need. Already, modern program synthesis empowers non-coders to build new voice assistant skills via a conversation with their phone;[20] write feedback about one student program to propagate feedback to many students' programs;[15] scrape large datasets from the Web by demonstrating how to scrape one row;[7] transform and clean data by giving examples of a few transformed items or cells;[13,19] and visualize or model a dataset by providing just the dataset.[6,23]

PL+HCI lets us use formal reasoning to create richer programming experiences. Some work that starts as advances to programming language theory or implementation ultimately invents novel programming interaction techniques. The simplicity of Smalltalk's object model enabled the language designers to develop many novel programming conveniences that we now take for granted—for example, an integrated development environment, reflection, and unit testing frameworks.[17] Work that starts as an effort to make incomplete programs well-typed can ultimately let us build programming environments that work just as well for partial programs as complete programs.[27] Work that starts as an effort to create bidirectional mappings between program inputs and outputs can let us build programming environments in which users can program by editing code or by tweaking a diagram.[16] By deeply considering formal models of programming, we can ultimately produce richer interactive programming systems.

PL+HCI produces better decisions about the division of labor between the human and the machine. Computers are better at some tasks than humans, and vice versa, and this landscape shifts as computing advances and education evolves. In the classical model of programming, the programmer instructs the machine, and the machine follows the instructions. Modern programming tools can divide programming tasks between human and machine in new and creative ways. For example, reasoning about whether a robot upholds human social norms (for example, maintaining eye contact) is usually left to the human programmer, but new human-robot interaction work offloads this task to a verifier,[30] effectively erecting guardrails that keep programmers from violating their own design goals. Synthesis tools let users offer input-output examples and other non-code specifications when those specifications are easier to provide than the code itself. Rather than requiring humans to hand-write low-level image processing pipelines, the Halide project[31] allows programmers to write in a high-level language, delegating the low-level scheduling details to the computer. This new generation of tools leverages a diverse array of techniques, everything from program synthesis and machine learning to domain-specific languages and program verification.

The union of PL and HCI techniques can dramatically reduce the programming skills required to automate important tasks.

Obstacles

There are two key obstacles to accomplishing our vision of united PL and HCI. First, most PL and HCI researchers lack knowledge of each other's tools and methods. The prerequisites for work in these fields are disjoint. Many, even most, researchers in PL or HCI enter one of these subfields without learning even the basics of the other.

Second, the demands for rigor in the PL and HCI communities do not always compose. We need knowledge of both communities to selectively apply the standards of each community as appropriate. Not all tasks should be neatly evaluated in a one-hour controlled user study. Not all claims require mathematical proofs. We are in danger of smothering exciting new research if we ask authors to check boxes that make sense for the single-subfield contributions we have seen before but not for the new contributions they are offering.

We believe that learning about both fields and their intersection is the best remedy to both these obstacles. We hope this article's glimpse into PL+HCI research inspires readers to learn more. Readers who want a preview of the kinds of contributions that will push this field forward—the kinds

Table 2. What can PL practitioners borrow from HCI? This table summarizes three classes of PL contribution that we can produce by drawing on HCI techniques.

Contribution	Key Message	Elaboration
Need-Finding	Pick good problems	If we identify real needs before we begin designing, we have a better chance of contributing useful, high-impact programming languages and tools.
Behavioral Theory	Develop theories of hu man capabilities and behavior	Given that user evaluation is time-consuming and expensive, we can make better design decisions more quickly if our field builds up theories that predict user behavior.
Iterative Refinement	And get frequent user feedback when you lack theory	PL innovation constantly uncovers new design questions. We can apply user-centered design—a feedback loop between builders and users, a cycle of evaluations and redesigns—to inform our decisions in addition to any applicable theory.

of papers we should be accepting into our favorite venues—should read on to the next section for a taste of how we advance to the vision of productively integrated HCI and PL.

Where Do We Go Next?

In this section, we present a vision of where this hybrid field should go now. We highlight a few types of contributions that harness PL and HCI's combined strengths. Readers who want to participate in the PL+HCI field can read on for a guide on how to contribute. We give specific recommendations for those with HCI backgrounds and those with PL backgrounds.

PL practitioners: Consider the following contribution types. We believe a few key contribution types have the potential to dramatically improve PL practice. By borrowing techniques from HCI, PL practitioners can produce higher-impact languages and tools, make their languages more usable by novices and experts, and make it easier for future language designers to produce usable languages.

Need-finding studies. Need-finding studies help us produce a rich understanding of the needs of a target population and ultimately identify the problems that real users need to solve. Contextual inquiry, interviews, surveys, analyses of log data, analyses of forums and StackOverflow, exploratory user studies—all of these can reveal important user needs. Need-finding has played an important role in shaping successful PL projects ranging from D3[5] and Vega[33] to FlashFill.[13] At their best, need-finding studies produce needs analyses that are useful not just for motivating a single project but for the research community as a whole.

The HCI and SE communities already publish standalone need-finding papers for a variety of user populations. The PL community serves different populations, with different problems, using different techniques. We have started to see excellent need-finding papers that address these populations, problems, and techniques, but we have just scratched the surface.[22]

Contribution: Standalone need-finding studies for populations, settings, and tasks that could be particularly well-served by novel programming language research.

Cognitive and behavioral theory transfer and development. Psychology, cognitive science, linguistics, and many other fields study the characteristics of human cognition. Their work offers theories about the classes of reasoning that humans find easy, hard, and impossible, with and without training. In the domain of PL design, a scientific understanding of how programmers write programs could guide us to better language and tool designs. In the 1970s, the PoP field started building the foundation for this direction, and their work points us to methods we can reuse for learning about modern high-level languages. Critical work in this field continues, drawing on work in software engineering, psychology, CS education, and HCI. Although language design rarely motivates current work in this area, we see a huge opportunity to design experiments to generate language-relevant theory.

One way to bootstrap theory development is to borrow or adapt theories from other disciplines. Social sciences ranging from psychology and economics to cognitive science, organizational behavior, and learning science offer behavioral theory that may apply to programming. It is common in other disciplines to write papers that adapt or transfer theory from one domain to another. We know of no examples in programming languages literature, but there are many such papers in HCI[1] and software engineering.[2] As we establish or adapt behavioral theories of programming, language designers can base design decisions on predicted user behavior, rather than direct experimentation with target users, to quickly make languages more useful and usable.

Contribution: Theory development and theory transfer, for predicting human cognition and behavior during interaction with programming systems.

Iterative refinement. We can learn from users throughout our PL design processes. Formative studies enable designers to learn from users before implementing a complete system. For instance, we can conduct formative user studies with incomplete prototypes, learning where users stumble and what features help them. Such studies can help us ensure our language designs are usable, learnable, and not error-prone before substantial effort is put into developing formalisms, proofs, compilers, and other high-effort artifacts. Methods for soliciting user feedback during the design process include surveys, interviews, focus groups, natural programming elicitation, think-aloud studies, "Wizard of Oz" studies in which a human plays the role of the compiler, and studies with other low-cost prototypes. We can evaluate some of the same questions without even recruiting users, for example, via cognitive walkthroughs or heuristic evaluation. HCI venues often publish papers describing such formative studies and the resulting designs. Programming language designs that have been iteratively refined via formative methods should similarly find a place in the literature.[8] As designers, we should get input from users early and often.

Contribution: Language and tool designs guided by user-centered, iterative design processes.

HCI → PL summary. With a new, broader, and more diverse audience interested in computing, we face an exciting time in our field's history. As we develop more of the contribution types described in this work, we are poised to offer useful, usable programming languages and tools that tackle high-impact problems. Taken together, these three contribution types, summarized in Table 2, tell us: Pick good problems, develop theories of human capabilities and behavior, and get frequent user feedback when you lack theory.

HCI practitioners: Consider the following contribution types. Going forward, we hope to see a few contributions become more common in the HCI community, as HCI increasingly draws on advances in PL. With new PL techniques, HCI practitioners can deliver even more powerful and flexible interfaces, help users avoid important classes of failures, and offer accessible new pathways into the world of computing.

PL-backed interfaces. Where you might typically design a GUI, consider giving your users a language—either a domain-specific programming language or a graphical UI that offers the key components of a language: primitives and means of composition. From here on, we will refer to the class of interfaces with primitives and means of composition, whether they are textual or graphical, as PL-backed interfaces.

PL-backed interfaces offer power and flexibility, enabling new interactions that other UI types cannot support. Con- sider the success stories of languages like D3[5] and Vega,[33] which could have been encapsulated within authoring tools, but not without sacrificing some expressiveness and user control. With advances in tools for language design and implementation—including support for domain-specific languages, language extensions, embedded languages—it is now much easier to offer PL-backed interfaces.[10]

Contribution: Developing or studying languages as UIs.

Guardrails to make PL-backed UIs safer. Giving users powerful, flexible PL-backed interfaces can empower them, but it can also empower them to make new mistakes. We can address this by building guardrails into our UIs, tools that prevent or catch errors, bugs, and bad outcomes. For this goal too, PL offers a wealth of techniques for aiding programmers, everything from verification (for example, verifying that a robot control program makes the robot compliant with human social norms[30]) to program analysis (for example, a spreadsheet extension that identifies likely spreadsheet errors based on discrepancies with other nearby formulas[3] or generates spreadsheet tests automatically[11]).

Contribution: Developing or studying techniques for enforcing or encouraging correct use of PL-backed interfaces.

Non-code inputs to PL-backed UIs. Although providing a PL-backed interface can put powerful new computing experiences in reach, it often takes more than a well-designed language to help users unlock a PL's full potential. We can help users author complex programs via PL tools that write code based on non-code specifications. Program synthesis paradigms like programming by demonstration, programming by example, and programming by manipulation offer users alternative ways to express their intent. For example, a Helena[7] user demonstrates how to collect the first row of their target dataset in a standard Web browser, and Helena synthesizes a program that traverses thousands or millions of webpages to collect the full dataset. Programming by demonstration thus enables social scientists and other domain experts to collect the data they need from the Web. Leveraging new PL techniques lets us design new interfaces for programming and ultimately brings the power of programming to new audiences.

Contribution: Developing or studying techniques for creating code from non-code specifications.

PL → HCI summary. We are excited for the potential of PL-backed interfaces in the future of HCI. As our users face increasingly complex new computing tasks, now is the time to put human-centered languages in the hands of more users. Together these three contribution types, summarized in Table 3, offer a simple takeaway message: Give users PLs, but help them use PLs responsibly, and don't expect code alone.

Fostering HCI+PL research. We believe the six contribution types discussed previously—need-finding, behavioral theory, iterative refinement, PL-backed interfaces, guardrails, and creating code from non-code—can advance both human-computer interaction and programming languages, that they represent important new frontiers for both subfields. We want to invest in these contribution types. What actions should we take, as individuals and as a community, to produce more work like this?

▸ **For new or aspiring PL+HCI researchers:** For new researchers, this article describes classes of work that represent important but under-explored contributions. Are you bringing expertise that would help you write a theory transfer paper? A "guardrails" paper? As our community is opening to these topics, now is a great time to consider these directions. If you're looking to test-drive this path, start attending talks. Take a PL class if you are more familiar with HCI, take an HCI class if you are more familiar with PL, or take one of the new crop of courses at the HCI-PL intersection. Start collaborations across the boundary. Find ways to publish the work in multiple but substantial coherent pieces, if necessary, to reach both fields.

▸ **For reviewers:** This article provides an overview of why these contribution

Table 3. What can HCI practitioners borrow from PL? This table summarizes three classes of HCI contribution that we can produce by drawing on PL techniques.

Contribution	Key Message	Elaboration
PL as UI	Give users PLs,	Giving your users a PL gives them a powerful tool that offers flexibility, expressiveness, and control.
Guardrails	But help them use PLs responsibly,	Both fields offer methods for building in guardrails—checks in the languages, tools, and environments that make errors less likely when we give users languages as interfaces. For example, static analysis, verification, and type systems can all offer important guardrails.
Beyond Code	And don't expect code alone.	Don't expect code alone to be enough to put the target programs in reach, especially for complex domains. Sometimes users need further aids, some of which can come from PL—for example, program synthesis, programming by demonstration, anything that makes code out of non-code, new editors, new programming experiences.

types are necessary, why they hold the promise of enriching both fields. We encourage you to read more on these topics, but we hope this article is reason enough to think twice before dismissing these contributions, even if the papers strike you as unusual or unprecedented at first.

▸ **For advisors and mentors:** Increasingly, we find researchers are succeeding not despite but because of their cross-disciplinary research. Students considering work at this intersection are not sacrificing job prospects. And as reviewers in both communities are becoming more open to work that combines contributions in both PL and HCI, there is less and less reason to limit your students to a single domain.

▸ **For the research community as a whole:** Venues like VL/HCC, PLATEAU, PPIG, and LIVE have long track records of recognizing and evaluating work at the intersection of PL and HCI. However, the work needs to appear at flagship conferences to thrive. These flagship conferences should invite reviewers with PL+HCI expertise and evaluate the work rigorously based on appropriate evidence standards.

▸ **For the industrial and practitioner community as a whole:** We want to see powerful PL-backed interfaces and usable programming languages reaching real users. We should be pouring resources and engineering effort into making it easier for humans to control computers. Few companies have engineering teams working on language design and language usability questions jointly. If you sell a product—cloud computing resources, data analysis suites—that people use via programming, or that people may want to automate, spin up an engineering team that joins PL and HCI expertise.

Conclusion

Computers have given us services, scientific results, and communication modes that we would not have achieved without them—but many modern interactions with computers feel constrained. Many users feel as if they work in service of the machine rather than the other way around. Even expert programmers still spend a surprising amount of time wrestling with the command line or tackling painful sysadmin tasks. If we are successful in this PL+HCI effort, it will be easier for us—programming experts, novices, and previously unreached users alike—to communicate our intent quickly and accurately to computers. It will be easier for us to rally computers to our billions of exciting and diverse human goals.

References

1. Alkhatib, A. and Bernstein, M. Street-level algorithms: A theory at the gaps between policy and decisions. In *Proceedings of the 2019 Conf. Human Factors in Computing Systems*.
2. Barik, T., Ford, D., Murphy-Hill, E., and Parnin, C. How should compilers explain problems to developers? In *Proc. Joint Meeting of the Euro. Software Engineering Conf. and Symp. Foundations of Software Engineering*, 2018.
3. Barowy, D., Berger, E., and Zorn, B. Excelint: Automatically finding spreadsheet formula errors. In *Proceedings of the ACM on Programming Languages 2* (2018), 1–26.
4. Blackwell, A., Petre, M., and Church, L. Fifty years of the psychology of programming. *Intern. J. Human-Computer Studies 131* (Nov. 2019), 52–63; http://oro.open.ac.uk/62027/.
5. Bostock, M., Ogievetsky, V., and Heer, J. D3 data-driven documents. *IEEE Trans. Visualization and Computer Graphics 17*, 12 (2011), 2301–2309.
6. Chasins, S. and Phothilimthana, P. Data-driven synthesis of full probabilistic programs. In *Proceedings of the 2017 Computer-Aided Verification*. Springer International Publishing.
7. Chasins, S., Mueller, M., and Bodik, R. Rousillon: Scraping distributed hierarchical web data. In *Proceedings of the 2018 Symp. User Interface Software and Technology*.
8. Coblenz, M., Aldrich, J., Myers, B., and Sunshine, J. Interdisciplinary programming language design. In *Proceedings of the 2018 Intern. Symp. New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!)*, 133–146.
9. Cooper, S., Dann, W., and Pausch, R. Alice: A 3-d tool for introductory programming concepts. *J. Comput. Sci. Coll. 15*, 5 (Apr. 2000), 107–116.
10. Erdweg, S. et al. The state of the art in language workbenches. In *Proceedings of the 2013 Intern. Conf. Software Language Engineering*.
11. Fisher, M., Rothermel, G., Brown, D., Cao, M., Cook, C., and Burnett, M. Integrating automated test generation into the WYSIWYT spreadsheet testing methodology. *ACM Trans. Softw. Eng. Methodol. 15*, 2 (Apr. 2006), 150–194; https://doi.org/10.1145/1131421.1131423.
12. Green, T. and Petre, M. Usability analysis of visual programming environments: A 'cognitive dimensions' framework. *J. Visual Languages & Computing 7*, 2 (1996), 131–174.
13. Gulwani. S. Automating string processing in spreadsheets using input-output examples. In *Proceedings of the Symp. Principles of Programming Languages*, 2011.
14. Hansen, M., Lumsdaine, A., and Goldstone, R. Cognitive architectures: A way forward for the psychology of programming. In *Proceedings of the ACM Intern. Symp. New Ideas, New Paradigms, and Reflections on Programming and Software, Onward!* 2012, 27–38. ACM, New York, NY, USA; https://doi.org/10.1145/2384592.2384596.
15. Head, A., Glassman, E., Soares, G., Suzuki, R., Figueredo, L., D'Antoni, L., and Hartmann, B. Writing reusable code feedback at scale with mixed-initiative program synthesis. In *Proceedings of the 4th ACM Conf. Learning @ Scale*, 2017.
16. Hempel, B., Lubin, J., Lu, G., and Chugh, R. Deuce: A lightweight user interface for structured editing. In *Proceedings of the 2018 Intern. Conf. Software Engineering*, 2018.
17. Kay, A. The early history of Smalltalk. *History of Programming Languages*—II, 1996, 511–598.
18. Ko, A., et al. The state of the art in end-user software engineering. *ACM Comput. Surv. 43*, 3 (Apr. 2011); https://doi.org/10.1145/1922649.1922658.
19. Le, V. and Gulwani, S. FlashExtract: A framework for data extraction by examples. In *Proceedings of the 2014 Conf. Programming Language Design and Implementation*.
20. Li, T., Radensky, M., Jia, J., Singarajah, K., Mitchell, T., and Myers, B. Pumice: A multi-modal agent that learns concepts and conditionals from natural language and demonstrations. In *Proceedings of the 2019 Symp. User Interface Software and Technology*.
21. Lieberman, H., Paternò, F., and Wulf, V. *End User Development (Human-Computer Interaction Series)*. Springer Verlag, Berlin, Heidelberg, 2006.
22. Ma'ayan, D., Ni, W., Ye, K., Kulkarni, C., and Sunshine, J. How domain experts create conceptual diagrams and implications for tool design. In *Proceedings of the 2020 Conf. Human Factors in Computing Systems*.
23. Moritz, D., Wang, C., Nelson, G., Lin, H., Smith, A., Howe, B., and Heer, J. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Trans. Visualization and Computer Graphics 25*, 1 (2018), 438–448.
24. Myers, B., Pane, J., and Ko, A. Natural programming languages and environments. *Commun. ACM 47*, 9 (Sept. 2004), 47–52; https://doi.org/10.1145/1015864.1015888.
25. Myers, B., Ko, A., LaToza, T., and Yoon, Y. Programmers are users too: Human-centered methods for improving programming tools. *Computer 49*, 7 (2016).
26. Nardi, B. A *Small Matter of Programming: Perspectives on End User Computing.* MIT Press, Cambridge, MA, USA, 1993.
27. Omar, C., Voysey, I., Hilton, M., Aldrich, J., and Hammer, M. Hazelnut: A bidirectionally typed structure editor calculus. In *Proceedings of the 2017 Symp. Principles of Programming Languages*.
28. Pane, J. and Myers, B. Usability issues in the design of novice programming systems. Project status report, 08, 1996.
29. Pane, J. and Myers, B. The influence of the psychology of programming on a language design: Project status report, 06, 2000.
30. Porfirio, D., Sauppé, A., Albarghouthi, A., and Mutlu, B. Authoring and verifying human-robot interactions. In *Proceedings of the 2018 Symp. User Interface Software and Technology*.
31. Ragan-Kelley, J., Barnes, C., Adams, A., Paris, S., Durand, F., and Amarasinghe, S. Halide: A language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. In *Proceedings of Programming Language Design and Implementation*, 2013.
32. Resnick, M., et al. Scratch: Programming for all. *Commun. ACM 52*, 11 (Nov. 2009), 60–67; https://doi.org/10.1145/1592761.1592779.
33. Satyanarayan, A., Moritz, D., Wongsuphasawat, K., and Heer, J. Vega-lite: A grammar of interactive graphics. *IEEE Trans. Visualization and Computer Graphics 23*, 1 (2016), 341–350.
34. Shneiderman, B. Direct manipulation: A step beyond programming languages. *Computer 16*, 8 (1983), 57–69.
35. Shneiderman, B. and Plaisant, C. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel evaluation methods for information visualization*, 1–7.
36. Verou, L., Zhang, A., and Karger, D. Mavo: Creating interactive data-driven web applications by authoring html. In *Proceedings of the 2016 Symp. User Interface Software and Technology*.
37. Yoon, Y. and Myers, B. Supporting selective undo in a code editor. In *Proceedings of the 2015 Intern. Conf. Software Engineering*.

Sarah E. Chasins (schasins@cs.berkeley.edu) is an assistant professor of Electrical Engineering and Computer Science at University of California, Berkeley, CA, USA.

Elena L. Glassman (glassman@seas.harvard.edu) is an assistant professor of computer science and the Stanley A. Marks and William H. Marks Assistant Professor at the Radcliffe Institute for Advanced Study at Harvard University, Cambridge, MA, USA.

Joshua Sunshine (sunshine@cs.cmu.edu) is a senior Research Fellow in the School of Computer Science at Carnegie Mellon University, Pittsburgh, PA, USA.

research highlights

P. 108
Technical Perspective
The Quest for Optimal Multi-Item Auctions
By Constantinos Daskalakis

P. 109
Optimal Auctions through Deep Learning
By Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai S. Ravindranath

P. 117
Technical Perspective
eBP Rides the Third Wave of Mobile Health
By Josiah D. Hester

P. 118
eBP: An Ear-Worn Device For Frequent and Comfortable Blood Pressure Monitoring
By Nam Bui, Nhat Pham, Jessica Jacqueline Barnitz, Zhanan Zou, Phuc Nguyen, Hoang Truong, Taeho Kim, Nicholas Farrow, Anh Nguyen, Jianliang Xiao, Robin Deterding, Thang Dinh, and Tam Vu

DOI:10.1145/3470440

To view the accompanying paper, visit doi.acm.org/10.1145/3470442 rh

Technical Perspective
The Quest for Optimal Multi-Item Auctions

By Constantinos Daskalakis

WHAT IS THE best way to sell one or multiple indivisible items to maximize revenue? Should they be priced separately, priced in bundles, or sold using a more complex sales mechanism? Despite their practical importance, ancient history, and centuries of scientific study, these questions have remained unanswered. The challenge facing sellers and scientists alike is that there are myriad sales mechanisms. While we are used to price tags on items in stores, many other sales mechanisms exist. Writing a model encompassing all possible mechanisms is already onerous, let alone optimizing over them.

It thus came as a surprise when, in the 1980s, economists developed mathematics that could pinpoint the best way of selling a *single* item to one or multiple strategic buyers, given distributional information about their values for the item. When these values are independent and identically distributed, the optimum can be cast as the familiar ascending price auction with a reserve price, such as what eBay uses, and becomes a somewhat more exotic auction when values are not identically distributed. Myerson's extremely influential, Nobel-prize winning work prescribes exactly how to set it up.

Alas, pushing this program to the *multi-item* setting has bedeviled economists and computer scientists. It should be appreciated that organizing the simultaneous sale of multiple items is neither an innocuous nor an esoteric problem. When governments sell radio spectrum, they simultaneously sell multiple licenses in one big auction. When ad exchanges sell banner ads on, say, the *New York Times* frontpage, they simultaneously sell all banners for each impression. When online retailers sell various items to competing buyers, they should organize their sales cleverly. Despite the importance of multi-item settings, optimal solutions remain elusive.

Indeed, extensive study has revealed that optimal multi-item mechanisms, even for selling two items to one buyer, are tremendously more complex than single-item ones. Bundling and randomized allocations are necessary, and the optimal mechanism may exhibit various counterintuitive properties, as I describe in a recent survey.[2] At the time of writing, there only exist results characterizing optimal multi-item *single-buyer* mechanisms.[3] Those results, using optimal transport theory to characterize the structure of the optimal mechanism in terms of stochastic dominance conditions satisfied by the buyer's value distribution, suggest that there is no one-size-fits-all optimal multi-item mechanism.

Absent a simple, universal structure in the optimal mechanism there are two natural approaches, which both bode well with the CS aesthetic and have been extensively pursued. One is trading optimality for simplicity, as pursued in a growing body of work that has been identifying progressively simpler, approximately optimal mechanisms for progressively more complex multi-item multi-buyer settings.

Another approach is to develop frameworks for *computing* optimal mechanisms on an instance-by-instance basis, given the buyers' value distributions. Such frameworks are useful both for their end products, that is, the mechanisms they compute, but also as exploratory tools, for investigating the existence of structure in the optimal mechanism over a range of settings of interest. There is a large volume of work on this topic, computing optimal mechanisms in quite complex settings via convex programming.[2]

The following paper by Dütting et al. contributes a very interesting and forward-looking new take on this computational challenge, initiating the use of deep learning for mechanism design.

Rather than taking the value distributions as input, their framework receives samples from that distribution and trains a neural network representing the mechanism to attain good empirical revenue. The sample complexity of optimal multi-item mechanisms has been somewhat studied, so we have a reasonable understanding of what settings allow near-optimal mechanisms to be identified from polynomially many samples; in particular, it is known that some conditional independence across item values is needed for statistical efficiency.[1]

Rather than pursuing optimality guarantees for their mechanisms, they again deviate from prior work by pursuing best-effort guarantees, that is, the best revenue that gradient descent may deliver, even without endowing their neural networks with much inductive bias about the structure of the optimal mechanism. Moreover, rather than exact truthfulness they seek approximate truthfulness, enforced via min-max training formulations for their neural networks.

With those design choices, the authors provide a flexible computational framework, which readily accommodates continuous distributions, and is more scalable in some dimensions of the problem. Despite the lack of strong inductive bias and gradient-based optimization, empirical testing shows that the mechanisms identified by their framework attain close to optimal revenue in simple settings where optimal mechanisms have been otherwise identified, while in other settings where the optimum had not been identified it computes mechanisms that are near-optimal. How robust are these findings in broader settings? And, if they are, what would be that elusive property of optimal multi-item mechanisms that enables relatively agnostic architectures to represent them and gradient-descent to identify them? C

References

1. Brustle, J., Cai, Y., Daskalakis, C. Multi-item mechanisms without item-independence: Learnability via robustness. In *Proceedings of the 21st ACM Conf. Economics and Computation*, 2020, 715–761.
2. Daskalakis, C. Multi-item auctions defying intuition? *SIGecom Exchanges 1*, 14 (2015), 41–75.
3. Daskalakis, C., Deckelbaum, A., Tzamos, C. Strong duality for a multiple-good monopolist. *Econometrica 3*, 85 (2017), 735–767.

Constantinos Daskalakis is a professor at MIT's EECS department and a member of CSAIL, Cambridge, MA, USA.

DOI:10.1145/3470442

Optimal Auctions Through Deep Learning

By Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai S. Ravindranath

Abstract

Designing an incentive compatible auction that maximizes expected revenue is an intricate task. The single-item case was resolved in a seminal piece of work by Myerson in 1981. Even after 30–40 years of intense research, the problem remains unsolved for settings with two or more items. We overview recent research results that show how tools from deep learning are shaping up to become a powerful tool for the automated design of near-optimal auctions auctions. In this approach, an auction is modeled as a multilayer neural network, with optimal auction design framed as a constrained learning problem that can be addressed with standard machine learning pipelines. Through this approach, it is possible to recover to a high degree of accuracy essentially all known analytically derived solutions for multi-item settings and obtain novel mechanisms for settings in which the optimal mechanism is unknown.

1. INTRODUCTION

Optimal auction design is one of the cornerstones of economic theory. It is of great practical importance, as auctions are used across industries and by the public sector to organize the sale of their products and services. Concrete examples are the US FCC Incentive Auction, the sponsored search auctions conducted by web search engines such as Google, and the auctions run on platforms such as eBay. In the standard *independent private valuations* model, each bidder has a valuation function over subsets of items, drawn independently from not necessarily identical distributions. The auctioneer knows the value distribution, but not the actual valuations (willingness to pay) of bidders. The bidders may act strategically and report untruthfully if this is to their benefit. One way to circumvent this is to require that it is in each agent's best interest to report its value truthfully. The goal then is to learn an incentive compatible auction that maximizes revenue.

In a seminal piece of work, Myerson resolved the optimal auction design problem when there is a single item for sale.[17] Quite astonishingly, even after 30–40 years of intense research, the problem is not completely resolved even for a simple setting with two bidders and two items. Our focus is on designing auctions that satisfy *dominant-strategy incentive compatibility (DSIC)*, which is a robust and desirable notion of incentive alignment. Although there have been some elegant partial characterization results,[6, 10, 15, 20] and an impressive sequence of algorithmic results, for example, Babaioff et al.[1] and Cai et al.,[2] these apply to the weaker notion of *Bayesian incentive compatibility (BIC)* except for the setting with one bidder, when DSIC and BIC coincide.

In Dütting et al.,[7] we have introduced a new, deep-learning-based approach to address the problem of optimal, multi-item auction design. In particular, we use multilayer neural networks to encode auction mechanisms, with bidder valuations forming the input and allocation and payment decisions forming the output. The networks are trained using samples from the value distributions, so as to maximize expected revenue subject to constraints for incentive compatibility. Earlier work has suggested to use algorithms to automate the design of mechanisms,[3] but where scalable, this earlier work had to restrict the search space to auction designs that are known to be incentive compatible.[13, 23] The deep learning approach, in contrast, enables searching over broad classes of not necessarily truthful mechanisms. Another related line of work has leveraged machine learning to optimize different aspects of mechanisms,[8, 18] but none of these offers the generality and flexibility of our approach.

Our framework provides two different approaches to handling DSIC constraints. In the first, we leverage results from economic theory that characterize DSIC mechanisms and model the network architecture appropriately. This approach, which we refer to as RochetNet, is applicable in single-bidder multi-item settings and provides exactly DSIC mechanisms.[22] In the second, we lift the DSIC constraints into the objective via the *augmented Lagrangian* method, which has the effect of introducing a penalty term for DSIC violations. This approach, which we refer to as *RegretNet*, is also applicable in multibidder multi-item settings for which we do not have tractable characterizations of DSIC mechanisms but will generally only find mechanisms that are approximately incentive compatible.

In this Research Highlight, we describe the general approach and present a selection of experimental results in support of our general finding that these approaches are capable of recovering, to a high degree of accuracy, the optimal auctions from essentially all analytical results obtained over the past 30–40 years and that deep learning is also a powerful tool for confirming or refuting hypotheses concerning the form of optimal auctions and can be used to find new designs. In the full version of the paper, we also prove *generalization bounds* that provide confidence intervals on the expected revenue and expected violation of DSIC based on empirical properties obtained

An extended abstract was published in *Proceedings of the 36th International Conference on Machine Learning, 2019*. A full version of this paper is available at https://arxiv.org/abs/1706.03459. All code is available through the GitHub repository at https://github.com/saisrivatsan/deep-opt-auctions.

during training, the complexity of the neural network used to encode the allocation and payment rules, and the number of samples used to train the network. Others have provided generalization bounds for training revenue-maximizing auctions in simpler settings; see, for example, Morgenstern and Roughgarden.[16]

Follow-up work has extended our approach to handle *budget constraints*,[9] as well as to a problem in social choice, the so-called *facility location problem*,[12] studied specialized architectures for single-bidder settings,[24] introduced networks that encode symmetry,[21] and provided methods to certify the strategy-proofness of learned mechanisms.[4]

2. OPTIMAL AUCTION DESIGN

We start by stating the optimal auction design problem and providing a few illustrative examples.

In the general version of the problem, we are given n bidders $N = \{1, ..., n\}$ and m items $M = \{1, ..., m\}$. Each bidder i has a valuation function $v_i: 2^M \rightarrow \mathbb{R}_{\geq 0}$, where $v_i(S)$ denotes how much the bidder values the subset of items $S \subseteq M$. In the simplest case, a bidder may have *additive* valuations. In this case, she has a value $v_i(\{j\})$ for each individual item $j \in M$, and her value for a subset of items $S \subseteq M$ is $v_i(S) = \sum_{j \in S} v_i(\{j\})$. If a bidder's value for a subset of items $S \subseteq M$ is $v_i(S) = \max_{j \in S} v_i(\{j\})$, we say this bidder has a *unit-demand* valuation. We also consider bidders with specific combinatorial valuations but defer the details to our full version.

Bidder i's valuation function is drawn independently from a distribution F_i over possible valuation functions V_i. We write $v = (v_1, ..., v_n)$ for a profile of valuations and denote $V = \prod_{i=1}^{n} V_i$. The auctioneer knows the distributions $F = (F_1, ..., F_n)$ but does not know the bidders' realized valuation v. The bidders report their valuations (perhaps untruthfully), and an auction decides on an allocation of items to the bidders and charges a payment to them. We denote an auction (g, p) as a pair of allocation rules $g_i : V \rightarrow 2^M$ and payment rules $p_i : V \rightarrow \mathbb{R}_{\geq 0}$ (these rules can be randomized). Given bids $b = (b_1, ..., b_n) \in V$, the auction computes an allocation $g(b)$ and payments $p(b)$.

A bidder with valuation v_i receives a utility $u_i(v_i; b) = v_i(g_i(b)) - p_i(b)$ for a report of bid profile b. Let v_{-i} denote the valuation profile $v = (v_1, ..., v_n)$ without element v_i, similarly for b_{-i}, and let $V_{-i} = \prod_{j \neq i} V_j$ denote the possible valuation profiles of bidders other than bidder i. An auction is *dominant strategy incentive compatible* (DSIC) if each bidder's utility is maximized by reporting truthfully no matter what the other bidders report. In other words, $u_i(v_i;(v_i, b_{-i})) \geq u_i(v_i; (b_i, b_{-i}))$ for every bidder i, every valuation $v_i \in V_i$, every bid $b_i \in V_i$, and all bids $b_{-i} \in V_{-i}$ from others. An auction is ex post *individually rational* (IR) if each bidder receives a nonzero utility, that is, $u_i(v_i; (v_i, b_{-i})) \geq 0\ \forall i \in N, v_i \in V_i$, and $b_{-i} \in V_{-i}$.

In a DSIC auction, it is in the best interest of each bidder to report truthfully, and so the revenue on valuation profile v is $\sum_i p_i(v)$. Optimal auction design seeks to identify a DSIC auction that maximizes expected revenue.

EXAMPLE 1 (VICKREY AUCTION[26]). *A classic result in auction theory concerns the sale of a single item to n bidders. It states that the following auction—the so-called Vickrey or second-price auction—is DSIC and maximizes social welfare: Collect a bid b_i from each bidder, assign the item to the bidder with the highest bid (breaking ties in an arbitrary but fixed manner), and make the bidder pay the second-highest bid.*

EXAMPLE 2 (MYERSON AUCTION[17]). *A simple example shows that the Vickrey auction does not maximize revenue: Suppose there are two bidders with $v_i \in U[0, 1]$, then its expected revenue is 1/3. Higher revenue can be achieved with a second-price auction with reserve r: As before, collect bids b_i, allocate to the highest bid but only if this bid is at least r, and make the winning bidder (if any) pay the maximum of the runner-up bid and r. It is straightforward to verify that this auction is DSIC and that choosing r = 1/2 leads to an expected revenue of* 5/12 > 1/3.

In the simple example with a single item and uniform valuations, a second-price auction with reserve 1/2 is in fact the optimal auction. This auction illustrates a special case of Myerson's theory for the design of revenue-optimal, single-item auctions.[17] Comparable results are not available for selling multiple items, even when we are trying to sell them to a single bidder!

3. THE LEARNING PROBLEM

At the core of our approach is the following reinterpretation of the optimal auction design problem as a learning problem, where in the place of a loss function that measures error against a target label, we adopt the negated, expected revenue on valuations drawn from F.

More concretely, the problem we seek to solve is the following: We are given a parametric class of auctions, $(g^w, p^w) \in \mathcal{M}$, for parameters $w \in \mathcal{R}^d$ for some $d \in \mathcal{N}$, and a sample of bidder valuation profiles $\mathcal{S} = \{v^{(1)}, ..., v^{(L)}\}$ drawn i.i.d. from F. Our goal is to find an auction that minimizes the negated, expected revenue $-\mathrm{E}[\sum_{i \in N} p_i^w(v)]$ among all auctions in $\mathcal{M}$ that satisfy incentive compatibility.

We consider two distinct approaches for achieving DSIC. In the first approach, we make use of characterization results. When it is possible to encode them within a neural network architecture, these characterizations from economic theory usefully constrain the search space and provide exact DSIC. At the same time, the particular characterization that we use is limited in that it applies only to single-bidder settings. The second approach that we take is more general, applying to multi-bidder settings, and does not rely on the availability of suitable characterization results. On the other hand, this approach entails search through a larger parametric space and only achieves approximate DSIC.

We describe the first approach in Section 4 and return to the second approach in Section 5.

4. THE ROCHETNET FRAMEWORK

We have developed two different frameworks that achieve exact DSIC by applying appropriate structure to the neural network architecture. One framework, referred to as *MyersonNet*, is inspired by Myerson's lemma[17] and can be used for the study of multi-bidder, single-item auctions (see the full version of this paper). A second framework, referred to as *RochetNet*, is inspired by Rochet's characterization theorem for DSIC auctions in single-bidder settings.[22] We give the construction of RochetNet for additive preferences, but this can be easily extended to unit-demand valuations.

4.1. The RochetNet architecture

For this single-bidder, multi-item setting, let $v \in \mathbb{R}^m_{\geq 0}$ denote the bidder's additive valuation, so that v_j is its value for item j. Let $b \in \mathbb{R}^m_{\geq 0}$ denote the bid, which need not be truthful. The allocation rule $g^w : \mathbb{R}^m_{\geq 0} \to [0,1]^m$, for parameters w, defines for each item $j \in [J]$ the probability $g^w_j(b) \in [0,1]$ with which the item is allocated to the bidder. The payment rule $p^w(b) : \mathbb{R}^m_{\geq 0} \to \mathbb{R}$ defines the payment $p^w(b)$ made by the bidder.

The mechanism (g^w, p^w) induces a *utility function* u^w: $\mathbb{R}^m_{\geq 0} \to \mathbb{R}$. For truthful bids, v, the utility function induced by the mechanism is

$$u^w(v) = g^w(v) \cdot v - p^w(v). \tag{1}$$

The RochetNet architecture represents the rules of a mechanism through a *menu*. The menu encodes a set of K choices, where each choice consists of a randomized allocation together with a price. The network selects the choice for the bidder that maximizes the bidder's reported utility given its bid, or chooses the *null outcome* (no allocation, no payment) when this is preferred. This yields the following utility function:

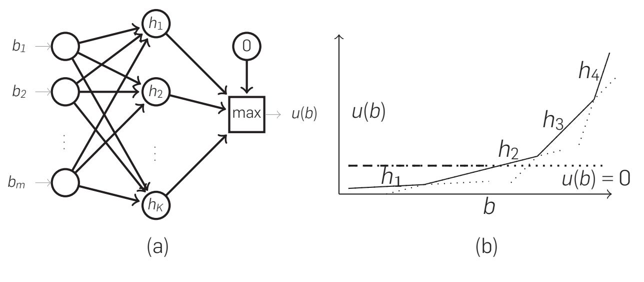
$$u^w(v) = \max\left\{\max_{k\in[K]}\{\alpha_k \cdot v + \beta_k\},\ 0\right\}, \tag{2}$$

with parameters $w = (\alpha, \beta)$, where $\alpha \in [0, 1]^{mK}$ and $\beta \in \mathbb{R}^K$. For choice $k \in [K]$, parameters $\alpha_k \in [0, 1]^m$ specify the randomized allocation and parameter $\beta_k \in \mathbb{R}$ specifies the negated price (β_ks will be negative, and the smaller the value of β_k, the larger the payment).

For input b, let $k^*(b) \in \text{argmax}_{k\in[K]\cup\{0\}}\{\alpha_k \cdot b + \beta_k\}$ denote the best choice for the bidder, where choice 0 corresponds to $\alpha_0 = 0$ and $\beta_0 = 0$ and the null outcome. This best choice defines the allocation and payment rule—for bid b, the allocation is $g^w(b) = \alpha_{k^*(b)}$ and the payment is $p^w(b) = -\beta_{k^*(b)}$.

RochetNet represents this induced utility function as a single layer neural network as illustrated in Figure 1(a). The input layer takes a bid $b \in \mathbb{R}^m_{\geq 0}$ and the output of the network is the induced utility. Figure 1(b) shows an example of an induced utility function for a single item ($m = 1$) and a network with a menu consisting of four choices ($K = 4$).

Figure 1. RochetNet: (a) Neural network representation of a menu, shown here with *K* choices as well as the null outcome (0); here, $h_k(b) = \alpha_k \cdot b + \beta_k$ for $b \in \mathbb{R}^m$, $\alpha_k \in [0, 1]^m$, and $\beta_k \in \mathbb{R}$. (b) An induced utility function represented by RochetNet for the case of a single item ($m = 1$) and a network with a menu with four choices ($K = 4$).



The network architecture ensures that the utility function is monotonically non decreasing, convex, and 1-Lipschitz, conforming to Rochet's characterization.[22] It also easily provides the following theoretical property.

THEOREM 4.1. *For any parameterization w, the mechanism* (g^w, p^w) *corresponding to RochetNet is DSIC and IR.*

PROOF. For DSIC, note that (1) the available choices are fixed, and independent of the report; and (2) for a truthful report, the "max" structure of RochetNet ensures that the bidder receives the choice that maximizes its true expected utility, and thus, the bidder can do no better than this. For IR, note that the expected utility for a true report is at least zero because of the availability of the null outcome. □

4.2. Training

During training, we seek to minimize the negated, expected revenue. Let F denote the distribution on valuation v. To ensure that the objective is a continuous function of α and β (so that parameters can be optimized through gradient descent), the best choice $k^*(v)$ at input v is approximated during training via a *softmax* operation in place of the argmax. With this, we seek to minimize the following loss function, which corresponds to the approximate, negated revenue:

$$\mathcal{L}(\alpha, \beta) = \mathrm{E}_{v\sim F}\left[\sum_{k\in[K]\cup\{0\}} \beta_k x_k(v)\right], \tag{3}$$

where

$$x_{k(v)} = \text{softmax}_k\big(0,\ c(\alpha_1 \cdot v + \beta_1),\ \ldots,\ c(\alpha_K \cdot v + \beta_K)\big) \tag{4}$$

and $c > 0$ is a constant that controls the quality of the approximation. The softmax function is $\text{softmax}_k(cz_0, cz_1, \ldots, cz_K) = e^{cz_k} / \sum_{k'} e^{cz_{k'}}$ and takes as input $K + 1$ real numbers and returns a probability distribution with each entry proportional to the exponential of the corresponding input. Once trained, RochetNet is used at test time with a hard max in place of the softmax to ensure exact DSIC and IR.

We train RochetNet using samples drawn from the bidder's value distribution. Given a sample $\mathcal{S} = \{v^{(1)}, \ldots, v^{(L)}\}$, we minimize the empirical loss, which is

$$\min_{\alpha,\beta} \frac{1}{L} \sum_{\ell\in[L]} \left(\sum_{k\in[K]\cup\{0\}} \beta_k x_k\left(v^{(\ell)}\right) \right). \tag{5}$$

We use projected stochastic gradient descent (SGD) to minimize (5). We estimate gradients for the loss using mini-batches of size 2^{15} valuation samples in every iteration. In the projection step, we project each parameter α_{jk} (for item j, choice k) onto [0, 1] to provide a well-defined probability.

5. THE REGRETNET FRAMEWORK

We next describe our second approach to handling DSIC constraints and the corresponding framework, which we refer to as *RegretNet*. Unlike the first approach, this second approach does not rely on characterizations of DSIC mechanisms. Instead, we replace the DSIC constraints with a differentiable approximation and lift the DSIC constraints into the objective by augmenting the objective with a term that accounts for the extent to which the DSIC constraints are violated. Here, we provide an overview of the special case in which bidders have additive values for items, but the framework also handles more general settings.

5.1. Expected ex post regret

We can measure the extent to which an auction violates incentive compatibility through a particular variation on *ex post* regret introduced in Dütting et al.[8] Fixing the bids of others, the ex post regret for a bidder is the maximum increase in her utility, considering all possible nontruthful bids.

For mechanisms (g^w, p^w), we will be interested in the *expected ex post regret* for bidder i:

$$rgt_i(w) = \mathrm{E}\left[\max_{v_i' \in V_i} u_i^w(v_i;(v_i',v_{-i})) - u_i^w(v_i;(v_i,v_{-i}))\right],$$

where the expectation is over $v \sim F$ and $u_i^w(v_i;b) = v_i(g_i^w(b)) - p_i^w(b)$ for model parameters w. We assume that F has full support on the space of valuation profiles V, and recognizing that the regret is nonnegative, an auction satisfies DSIC if and only if $rgt_i(w) = 0, \forall i \in N$, except for measure zero events.

Given this, we reformulate the learning problem as minimizing expected negated revenue subject to the expected ex post regret being zero for each bidder:

$$\begin{aligned}&\min_{w \in \mathbb{R}^d} \quad \mathrm{E}_{v\sim F}\left[-\textstyle\sum_{i\in N} p_i^w(v)\right]\\ &\text{s.t.} \quad rgt_i(w) = 0, \forall i \in N.\end{aligned}$$

Given a sample $\mathcal{S}$ of L valuation profiles from F, we estimate the empirical ex post regret for bidder i as:

$$\widehat{rgt}_i(w) = \frac{1}{L}\sum_{\ell=1}^{L}\left[\max_{v_i' \in V_i} u_i^w(v_i^{(\ell)};(v_i',v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)};v^{(\ell)})\right], \tag{6}$$

and seek to minimize the empirical loss (negated revenue) subject to the empirical regret being zero for all bidders:

$$\begin{aligned}&\min_{w \in \mathbb{R}^d} \quad -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w(v^{(\ell)})\\ &\text{s.t.} \quad \widehat{rgt}_i(w) = 0,\ \forall i \in N.\end{aligned} \tag{7}$$

We additionally require the designed auction to satisfy IR, which can be ensured by restricting the search space to a class of parameterized auctions that charge no bidder more than her valuation for an allocation.

5.2. The RegretNet architecture

In this case, the goal is to train neural networks that explicitly encode the allocation and payment rule of the mechanism. The architectures generally consist of two logically distinct components: the allocation and payment networks. These components are trained together and the outputs of these networks are used to compute the regret and revenue of the auction.

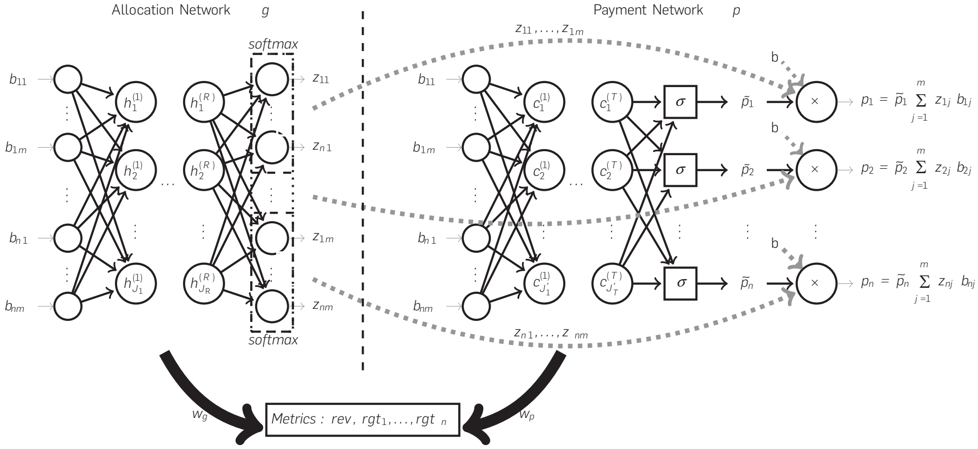
An overview of the RegretNet architecture for additive valuations is given in Figure 2.

The allocation network encodes a randomized allocation rule $g^w : \mathbb{R}^{nm} \to [0, 1]^{nm}$ and the payment network encodes a payment rule $p^w : \mathbb{R}^{nm} \to \mathbb{R}^n_{\geq 0}$, both of which are modeled as feedforward, fully-connected networks with a tanh activation function in each of the hidden nodes. The input layer of the networks consists of bids $b_{ij} \geq 0$ representing the valuation of bidder i for item j.

The allocation network outputs a vector of allocation probabilities $z_{1j} = g_{1j}(b), \ldots, z_{nj} = g_{nj}(b)$, for each item $j \in [m]$. To ensure feasibility, that is, the probability of an item being allocated is at most one, the allocations are computed using a softmax activation function, so that for all items j, we have $\sum_{i=1}^{n} z_{ij} \leq 1$. To accommodate the possibility of an item not being assigned, we include a dummy node in the softmax computation to hold the residual allocation probability. The payment network outputs a payment for each bidder that denotes the amount the bidder should pay in expectation for a particular bid profile.

To ensure that the auction satisfies *IR*, that is, does not charge a bidder more than her expected value for the

Figure 2. RegretNet: The allocation and payment networks for a setting with *n* additive bidders and *m* items. The inputs are bids from each bidder for each item. The revenue *rev* and expected ex post *rgt*$_i$ are defined as a function of the parameters of the allocation and payment networks *w* = (*w*$_g$, *w*$_p$).



allocation, the network first computes a normalized payment $\tilde{p}_i \in [0,1]$ for each bidder i using a sigmoidal unit, and then outputs a payment $p_i = \tilde{p}_i(\sum_{j=1}^{m} z_{ij} b_{ij})$, where the z_{ij}'s are the outputs from the allocation network.

5.3. Training

For RegretNet, we have used the *augmented Lagrangian method* to solve the constrained training problem in (7) over the space of neural network parameters w.

Algorithm 1 RegretNet Training

```
1:  **Input:** Minibatches $\mathcal{S}_1, \ldots, \mathcal{S}_T$ of size $B$
2:  **Parameters:** $\forall t, \rho_t > 0, \gamma > 0, \eta > 0, \Gamma \in \mathbb{N}, K \in \mathbb{N}$
3:  **Initialize:** $w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^n$
4:  **for** $t = 0$ **to** $T$ **do**
5:      Receive minibatch $\mathcal{S}_t = \{v^{(1)}, \ldots, v^{(B)}\}$
6:      Initialize misreports $v_i'^{(\ell)} \in V_i, \forall \ell \in [B], i \in N$
7:      **for** $r = 0$ **to** $\Gamma$ **do**
8:          $\forall \ell \in [B], i \in N$:
9:              $v_i'^{(\ell)} \leftarrow v_i'^{(\ell)} + \gamma \nabla_{v_i'} u_i^w(v_i^{(\ell)}; (v_i'^{(\ell)}, v_{-i}^{(\ell)}))$
10:     **end for**
11:     Compute regret gradient: $\forall \ell \in [B], i \in N$:
12:         $g_{\ell,i}^t =$
13:           $\nabla_w [u_i^w(v_i^{(\ell)}; (v_i'^{(\ell)}, v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; v^{(\ell)})]\Big|_{w=w^t}$
14:     Compute Lagrangian gradient (8) on $\mathcal{S}_t$ and update:
15:         $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{C}_{\rho_t}(w^t, \lambda^t)$
16:     Update Lagrange multipliers once in $Q$ iterations:
17:         **if** $t$ is a multiple of $Q$
18:             Compute $\widehat{rgt}$ on $\mathcal{S}_t$
19:             $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t \widehat{rgt}_i(w^{t+1}), \forall i \in N$
20:         **else**
21:             $\lambda^{t+1} \leftarrow \lambda^t$
22: **end for**
```

We first define the Lagrangian function for the optimization problem, augmented with a quadratic penalty term for violating the constraints:

$$\mathcal{C}_\rho(w; \lambda) = -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i\in N} p_i^w(v^{(\ell)}) + \sum_{i\in N}\lambda_i \widehat{rgt}_i(w) + \frac{\rho}{2}\sum_{i\in N}\left(\widehat{rgt}_i(w)\right)^2$$

where $\lambda \in \mathbb{R}^n$ is a vector of Lagrange multipliers and $\rho > 0$ is a fixed parameter that controls the weight on the quadratic penalty. The solver alternates between the following updates on the model parameters and the Lagrange multipliers: (a) $w^{new} \in \operatorname{argmin}_w \mathcal{C}_\rho(w^{old}; \lambda^{old})$ and (b) $\lambda_i^{new} = \lambda_i^{old} + \rho\,\widehat{rgt}_i(w^{new}), \forall i \in N$.

The solver is described in Algorithm 1. We divide the training sample $\mathcal{S}$ into minibatches of size B, estimate gradients on the minibatches, and perform several passes over the training samples. The update (a) on model parameters involves an unconstrained optimization of $\mathcal{C}_\rho$ over w and is performed using a gradient-based optimizer. The gradient $\mathcal{C}_\rho$ of w.r.t. w for fixed λ^t is given by:

$$\nabla_w \mathcal{C}_\rho(w; \lambda^t) = -\frac{1}{B}\sum_{\ell=1}^{B}\sum_{i\in N}\nabla_w p_i^w(v^{(\ell)}) + \sum_{i\in N}\sum_{\ell=1}^{B}\lambda_i^t g_{\ell,i} + \rho\sum_{i\in N}\sum_{\ell=1}^{B}\widehat{rgt}_i(w) g_{\ell,i}, \tag{8}$$

where

$$g_{\ell,i} = \nabla_w\left[\max_{v_i' \in V_i} u_i^w(v_i^{(\ell)}; (v_i', v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; v^{(\ell)})\right].$$

The terms $\widehat{rgt}_i$ and $g_{\ell,i}$ in turn involve a "max" over misreports for each bidder i and valuation profile ℓ. We solve this inner maximization over misreports using another gradient-based optimizer (lines 6–10).

As the optimization problem is nonconvex, the solver is not guaranteed to reach a globally optimal solution. However, this method proves very effective in our experiments, and we find that the learned auctions incur very low regret and closely match the structure of optimal auctions in settings where this is known.

6. EXPERIMENTS

We present and discuss a selection of experiments out of a broad range of experiments that we have conducted and that we describe in more detail in Düetting et al.[7] and the full version. The experiments demonstrate that our approach can recover near-optimal auctions for essentially all settings for which the optimal design is analytically known, that it is an effective tool for confirming or refuting hypotheses about optimal designs, and that it can find new auctions for settings where there is no known analytical solution.

6.1. Setup

We implemented our framework using the TensorFlow deep learning library.

For RochetNet, we initialized parameters α and β in Eq. (2) using a random uniform initializer over the interval [0,1] and a zero initializer, respectively. For RegretNet, we used the tanh activation function at the hidden nodes, and Glorot uniform initialization.[11] We performed cross-validation to decide on the number of hidden layers and the number of nodes in each hidden layer. We include exemplary numbers that illustrate the trade-offs in Section 6.6.

We trained RochetNet on 2^{15} valuation profiles and sampled every iteration in an online manner. We used the Adam optimizer with a learning rate of 0.1 for 20,000 iterations for making the updates. The parameter κ in Eq. (4) was set to 1000. Unless specified otherwise, we used a max network over 1000 linear functions to model the induced utility functions and report our results on a sample of 10,000 profiles.

For RegretNet, we used a sample of 640,000 valuation profiles for training and a sample of 10,000 profiles for testing. The augmented Lagrangian solver was run for a maximum of 80 epochs (full passes over the training set) with a minibatch size of 128. The value of ρ in the augmented Lagrangian was set to 1.0 and incremented every two epochs. An update on w^t was performed for every minibatch using the Adam optimizer with learning rate 0.001. For each update on w^t, we ran $\Gamma = 25$ misreport update steps with learning rate 0.1. At the end of 25 updates, the optimized misreports for the current minibatch were cached and used to initialize the misreports for the same minibatch in the next epoch. An update on λ^t was performed once every 100 minibatches (i.e., $Q = 100$).

We ran all our experiments on a compute cluster with NVDIA Graphics Processing Unit (GPU) cores.

6.2. Evaluation

In addition to the revenue of the learned auction on a test set, we also evaluate the regret achieved by RegretNet, averaged across all bidders and test valuation profiles, that is, $rgt = \frac{1}{n}\sum_{i=1}^{n} \widehat{rgt}_i(g^w, p^w)$. Each $\widehat{rgt}_i$ has an inner "max" of the utility function over bidder valuations $v_i' \in V_i$ (see (6)). We evaluate these terms by running gradient ascent on v_i' with a step-size of 0.1 for 2000 iterations (we test 1000 different random initial v_i' and report the one that achieves the largest regret). For some of the experiments, we also report the total time it took to train the network. This time is incurred during offline training, whereas the allocation and payments can be computed in a few milliseconds once the network is trained.

6.3. The Manelli-Vincent auction

As a representative example the optimal designs from economic theory that we can almost exactly recover with our approach, we discuss the Manelli-Vincent auction.[15]

A. Single bidder with additive valuations over two items, where the item values are independent draws from $U[0, 1]$.

The optimal auction for this setting is given by Manelli and Vincent.[15] We used two hidden layers with 100 hidden nodes in RegretNet for this setting. A visualization of the optimal allocation rule and those learned by RochetNet and RegretNet is given in Figure 3. Figure 4(a) gives the optimal revenue, the revenue and regret obtained by RegretNet, and the revenue obtained by RochetNet. Figure 4(b) shows how these terms evolve over time during training in RegretNet.

Both approaches essentially recover the optimal design, not only in terms of revenue but also in terms of the allocation rule and transfers. The auction learned by RochetNet is exactly DSIC and matches the optimal revenue precisely, with sharp decision boundaries in the allocation and payment rule. The decision boundaries for RegretNet are smoother, but still remarkably accurate. The revenue achieved by RegretNet matches the optimal revenue up to a <1% error term and the regret it incurs is <0.001. The plots of the test revenue and regret show that the augmented Lagrangian method is effective in driving the test revenue and the test regret toward optimal levels.

The additional domain knowledge incorporated into the RochetNet architecture leads to exactly DSIC mechanisms that match the optimal design more accurately and speeds up computation (the training took about 10 minutes compared to 11 hours for RegretNet). On the other hand, we find it surprising how well RegretNet performs given that it starts with no domain knowledge at all.

6.4. The Straight-Jacket auction

Extending the analytical result of Manelli and Vincent[15] to a single bidder and an arbitrary number of items (even with additive preferences, all uniform on [0, 1]) has proven elusive. It is not even clear whether the optimal mechanism is deterministic or requires randomization.

Figure 3. Side-by-side comparison of allocation rules learned by RochetNet (panels (a)) and RegretNet (panels (b)) for Setting A. The panels describe the probability that the bidder is allocated item 1 (left) and item 2 (right) for different valuation inputs. The optimal auctions are described by the regions separated by the dashed black lines, with the numbers in black being the optimal probability of allocation in the region.

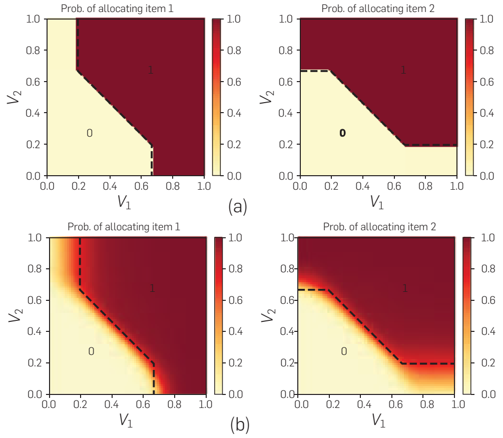
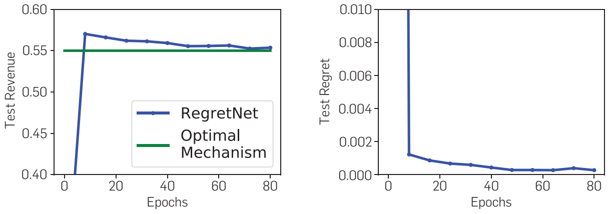


Figure 4. (a) Test revenue and regret for RegretNet and revenue for RochetNet for Setting A. (b) Plot of test revenue and regret as a function of training epochs for Setting A with RegretNet.

Distribution	Opt	RegretNet		RochetNet
	rev	rev	rgt	rev
Setting A	0.550	0.554	<0.001	0.550

(a)



(b)

Giannakopoulos and Koutsoupias[10] proposed a Straight-Jacket Auction (SJA) and gave a recursive algorithm for finding the subdivision and the prices, and used LP duality to prove that the SJA is optimal for items. These authors also conjecture that the SJA remains optimal for $m \leq 6$ general m but were unable to prove it.

Figure 5 gives the revenue of the SJA and that found by RochetNet for $m \leq 10$ items. We used a test sample of 2^{30} valuation profiles (instead of 10,000) to compute these numbers for higher precision. It shows that RochetNet finds the optimal revenue for $m \leq 6$ items and that it finds DSIC auctions whose revenue matches that of the SJA for m = 7, 8, 9, and 10 items. Closer inspection reveals that the allocation and payment rules learned by RochetNet essentially match those predicted by Giannakopoulos and Koutsoupias[10] for all $m \leq 10$. We take this as strong additional evidence that the conjecture of Giannakopoulos and Koutsoupias[10] is correct.

6.5. Discovering new optimal designs

RochetNet can also be used to aid the discovery of new, provably optimal designs. For this, we consider a single bidder with additive but correlated valuations for two items as follows:

B. One additive bidder and two items, where the bidder's valuation is drawn uniformly from the triangle $T = \{(v_1, v_2) | \frac{v_1}{c} + v_2 \leq 2, v_1 \geq 0, v_2 \geq 1\}$ where $c > 0$ is a free parameter.

There is no analytical result for the optimal auction design for this setting. We ran RochetNet for different values of c (e.g., 0.5, 1, 3, 5) to discover the optimal auction. Based on this, we conjectured that the optimal mechanism contains two menu items for $c \leq 1$, namely $\{(0, 0), 0\}$ and $\{(1, 1), \frac{2+\sqrt{1+3c}}{3}\}$, and three menu items for $c > 1$, namely $\{(0, 0), 0\}$, $\{(1/c, 1), 4/3\}$, and $\{(1, 1), 1 + c/3\}$, giving the optimal allocation and payment in each region. In particular, as c transitions from values less than or equal to 1 to values larger than 1, the optimal mechanism transitions from being deterministic to being randomized. We have used duality theory[5] to prove the optimality of this design, as stated in Theorem 6.1.

THEOREM 6.1. *For any* $c > 0$, *suppose the bidder's valuation is uniformly distributed over set* $T = \{(v_1, v_2) | \frac{v_1}{c} + v_2 \leq 2$ $v_1 \geq 0, v_2 \geq 1\}$. *Then, the optimal auction contains two menu items* $\{(0, 0), 0\}$ *and* $\{(1,1), \frac{2+\sqrt{1+3c}}{3}\}$ *when* $c \leq 1$, *and three menu items* $\{(0, 0), 0\}$, $\{(1/c, 1), 4/3\}$, *and* $\{(1, 1), 1+c/3\}$ *otherwise.*

6.6. Scaling up

We have also considered settings with up to five bidders and up to ten items. This is several orders of magnitude more complex than settings that can be addressed through other computational approaches to DSIC auction design. It is also a natural playground for RegretNet as no tractable characterizations of DSIC mechanisms are known for these settings.

The following two settings generalize the basic setting considered in Manelli and Vincent[15] and Giannakopoulos and Koutsoupias[10] to more than one bidder:

C. Three additive bidders and ten items, where bidders draw their value for each item independently from the uniform distribution $U[0,1]$.

D. Five additive bidders and ten items, where bidders draw their value for each item independently from the uniform distribution $U[0,1]$.

The optimal auction for these settings is not known. However, running a separate Myerson auction for each item is optimal in the limit of the number of bidders.[19] For a regime with a small number of bidders, this provides a strong benchmark. We also compare to selling the grand bundle via a Myerson auction.

For Setting C, we show in Figure 6(a) the revenue and regret of the learned auction on a validation sample of 10,000 profiles, obtained with different architectures. Here, (R, K) denotes an architecture with R hidden layers and K nodes per layer. The (5, 100) architecture has the lowest regret among all the 100-node networks for both Setting C and Setting D. Figure 6(b) shows that the learned auctions yield higher revenue compared to the baselines, and do so with tiny regret.

Figure 5. Revenue of the Straight-Jacket Auction (SJA) computed via the recursive formula in Giannakopoulos and Koutsoupias[10] and that of the auction learned by RochetNet, for various numbers of items *m*. The SJA is known to be optimal for up to six items and conjectured to be optimal for any number of items.

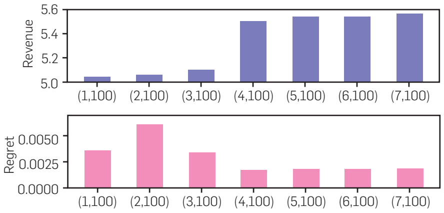
Items	SJA (*rev*)	RochetNet (*rev*)
2	0.549187	0.549175
3	0.875466	0.875464
4	1.219507	1.219505
5	1.576457	1.576455
6	1.943239	1.943216
7	2.318032	2.318032
8	2.699307	2.699305
9	3.086125	3.086125
10	3.477781	3.477722

7. CONCLUSION

The results from this research demonstrate that the methods of deep learning can be used to find close approximations to optimal designs from auction theory where they are known, to aid with the discovery of new optimal designs, and to scale-up computational approaches to optimal, DSIC auction design. Although our approach can be applied to settings that are orders of magnitude more complex than those that can be reached through other approaches to optimal DSIC design, a natural next step would be to scale this approach further to industry scale (e.g., through standardized benchmarking suites and innovations in network architecture). We also see promise for this framework in advancing economic theory, for example in supporting or refuting conjectures and as an assistant in guiding new economic discovery.

More generally, we believe that our work (together with a handful of contemporary works such as Hartford et al.,[14] Thompson et al.[25]) has opened the door to ML-assisted economic theory and practice, and we are looking forward to the advances that this agenda will bring along. C

Figure 6. (a) Revenue and regret of RegretNet on the validation set for auctions learned for Setting C using different architectures, where (*R*, *K*) denotes *R* hidden layers and *K* nodes per layer. (b) Test revenue and regret for Settings C and D, for the (5, 100) architecture.



(a)

Setting	RegretNet rev	RegretNet rgt	Item-wise Myerson	Bundled Myerson
C: 3 × 10	5.541	< 0.002	5.310	5.009
D: 5 × 10	6.778	< 0.005	6.716	5.453

(b)

References

1. Babaioff, M., Immorlica, N., Lucier, B., Weinberg, S.M. A simple and approximately optimal mechanism for an additive buyer. In *Proceedings of the 55th IEEE Symposium on Foundations of Computer Science*, 2014, 21–30.
2. Cai, Y., Daskalakis, C., Weinberg, S.M. An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the 44th ACM Symposium on Theory of Computing*, 2012, 459–478.
3. Conitzer, V., Sandholm, T. Complexity of mechanism design. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, 2002, 103–110.
4. Curry, M.J., Chiang, P.-Y., Goldstein, T., Dickerson, J.P. Certifying strategyproof auction network. In *Proceedings of the 34th Conference on Neural Information Processing Systems* (NeurIPS 2020).
5. Daskalakis, C., Deckelbaum, A., Tzamos, C. Mechanism design via optimal transport. In *Proceedings of the 14th ACM Conference on Electronic Commerce*, 2013, 269–286.
6. Daskalakis, C., Deckelbaum, A., Tzamos, C. Strong duality for a multiple-good monopolist. *Econometrica*, 85 (2017), 735–767.
7. Dütting, P., Feng, Z., Narasimhan, H., Parkes, D.C., Ravindranath, S.S. Optimal auctions through deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019, 1706–1715.
8. Dütting, P., Fischer, F., Jirapinyo, P., Lai, J., Lubin, B., Parkes, D.C. Payment rules through discriminant-based classifiers. *ACM Trans. Econ. Comput. 1*, 3 (2014), 5.
9. Feng, Z., Narasimhan, H., Parkes, D.C. Deep learning for revenue-optimal auctions with budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, 2018, 354–362.
10. Giannakopoulos, Y., Koutsoupias, E.. Duality and optimality of auctions for uniform distributions. In *SIAM J. Comput.*, 47 (2018), 121–165.
11. Glorot, X., Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.
12. Golowich, N., Narasimhan, H., Parkes, D.C. Deep learning for multi-facility location mechanism design. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, 261–267.
13. Guo, M., Conitzer, V. Computationally feasible automated mechanism design: General approach and case studies. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.
14. Hartford, J.S., Wright, J.R., Leyton-Brown, K. Deep learning for predicting human strategic behavior. In *Proceedings of the 29th Conference on Neural Information Processing Systems*, 2016, 2424–2432.
15. Manelli, A., Vincent, D. Bundling as an optimal selling mechanism for a multiple-good monopolist. *J. Econ. Theory 1*, 127 (2006), 1–35.
16. Morgenstern, J., Roughgarden, T. On the pseudo-dimension of nearly optimal auctions. In *Proceedings of the 28th Conference on Neural Information Processing Systems*, 2015, 136–144.
17. Myerson, R. Optimal auction design. *Math. Operat. Res.*, 6 (1981), 58–73.
18. Narasimhan, H., Agarwal, S., Parkes, D.C. Automated mechanism design without money via machine learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, 433–439.
19. Palfrey, T. Bundling decisions by a multiproduct monopolist with incomplete information. *Econometrica 2*, 51 (1983), 463–483.
20. Pavlov, G. Optimal mechanism for selling two goods. *B.E. J. Theor. Econ.*, 11 (2011), 1–35.
21. Rahme, J., Jelassi, S., Bruna, J., Weinberg, S.M. A permutation-equivariant neural network architecture for auction design. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence* (2021).
22. Rochet, J.-C. A necessary and sufficient condition for rationalizability in a quasilinear context. *J. Math. Econ.*, 16 (1987), 191–200.
23. Sandholm, T., Likhodedov, A. Automated design of revenue-maximizing combinatorial auctions. *Oper. Res. 5*, 63 (2015), 1000–1025.
24. Shen, W., Tang, P., Zuo, S. Automated mechanism design via neural networks. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, 2019. Forthcoming.
25. Thompson, D., Newman, N., Leyton-Brown, K.. The positronic economist: A computational system for analyzing economic mechanisms. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017, 720–727.
26. Vickrey, W. Counterspeculation, auctions, and competitive sealed tenders. *J. Finance*, 16 (1961), 8–37.

Paul Dütting (duetting@google.com), Google Research, Zürich, Switzerland.

Zhe Feng (zhe_feng@g.harvard.edu), School of Engineering and Applied Sciences at Harvard University, Cambridge, MA, USA. This work is supported in part through a Google Ph.D. Fellowship.

Sai S. Ravindranath (saisr@g.harvard.edu), School of Engineering and Applied Sciences at Harvard University, Cambridge MA, USA.

David C. Parkes (parkes@eecs.harvard.edu), Harvard University, MA, USA. This work is supported in part through NSF award CCF-1841550.

Harikrishna Narasimhan (hnarasimhan@google.com), Google Research, Mountain View, CA, USA.

DOI:10.1145/3470444

To view the accompanying paper, visit doi.acm.org/10.1145/3470446

Technical Perspective
eBP Rides the Third Wave of Mobile Health

By Josiah D. Hester

IMAGINE, FOR A second, that you suffer from hypertension (high blood pressure), diabetes, have a family history of heart disease, or have long battled with asthma. Unfortunately, most of you won't have to imagine this; according to the World Health Organization, over two billion people worldwide live with one of these chronic health conditions, so likely you or a close family member are afflicted. Treating these conditions requires frequent doctor visits and physical checks, both in and out of the clinic. Careful health maintenance and preventive care can lead to a happy and long life but is a lifelong burden on the patient and the doctor. Without it, however, these conditions can be deadly.

Why do they become deadly? The key problem, even in developed, rich countries, is access to continuous care; this can happen in a variety of ways, from inequitable healthcare systems, lack of trained professionals, distance from hospitals, lack of access to data, or patient non-adherence. These problems are now compounded by the COVID-19 pandemic, which has overloaded hospital systems and reduced health infrastructure's ability to prioritize preventive care. The bitter pill here is that preventive care is the best way to reduce this load long term, but it is just not feasible with available resources and exhausted clinicians.

For the past three decades, the computing community has asked if it can fill this preventive care gap. eBP—the automated blood pressure wearable system described in the following paper—is a sterling example of the third wave of mobile health tech to fill this gap. Where did it start? Well, in the 1960s, "Star Trek" showed the tricorder: a device that senses bio-signals, records them, and computes to prescribe treatment. In the 1980s and 1990s, before smartwatches, FitBits, or even iPods—researchers were building wearable prototypes enabling simple interaction with users and sensing of a few things like motion. Smart clothes, headbands, wristwatches, that sense, record, and compute like the tricorder. These were clunky—with poor battery life and limited ability—but proved the concept of mobile health and sparked the field. In the 2000s, everyone had a smartphone in their pocket, and the next wave of mobile health leveraged these modern versions of the "Star Trek" tricorder. Because of the phone, these mobile health approaches were the first to have broad usage. Researchers used the phone's camera, motion sensor, and significant computational ability to do things like screen for anemia and measure heart rate.

For the past decade, and especially in the last five years, the third wave of mobile health has been in full force. Taking advantage of the increases in computational power, accuracy, and efficiency of modern MEMS sensors, and major strides in applying machine learning and signal processing to biological data, a new class of wearable has emerged. These are unobtrusive and highly sophisticated devices that enable sensing of things we used to think were only possible in "Star Trek" and science fiction. More than just a FitBit, these devices have allowed us to identify and understand complex health-related human activities like eating, smoking, and sleeping, as well as accurately measure novel health markers like stress, respiration, pain, glucose levels—and in the case of eBP—blood pressure.

> eBP and the third wave of mobile health devices will eventually replace expensive clinical machines without giving up accuracy.

Blood pressure is an important health marker for a variety of conditions. Accurate measurement requires a cuff which compresses the artery on your arm for a pulse signal. eBP supports continuous blood pressure measurement with clinical accuracy, outside the clinic. It is a small device that sits in and around a person's ear. A balloon inside the ear is automatically inflated to contact the artery in the ear, then a sensor on the balloon touching the skin can gather a weak signal. I greatly simplify here, but I don't want to spoil the fun! The paper details the significant challenges in making this system work, from weak and noisy signals, uneven contact, biocompatibility, battery lifetime, and user burden.

eBP is a wonderful example of this third wave of mobile health in two ways. First, the system achieves technical and computational sophistication within the constraints of modern medicine and telehealth. We can learn a lot from eBP on designing resource constrained computers. Second, the project leverages partnerships across health and computing disciplines, giving a laser focus on the actual needs of people, and a deep understanding of the issues that arise when trying to deploy health monitoring devices.

eBP and this third wave of mobile health devices will eventually replace expensive clinical machines without giving up accuracy. Soon these devices will provide automated interventions, suggesting exercises, dietary change, or mindfulness sessions. Preventive healthcare assisted by low-cost wearables will allow our medical workers to move from crisis care to long-term, data-driven health maintenance of their patients. C

Josiah D. Hester is an assistant professor in the McCormack School of Engineering at Northwestern University, Evanston, IL, USA.

DOI:10.1145/3470446

eBP: An Ear-Worn Device For Frequent and Comfortable Blood Pressure Monitoring

By Nam Bui, Nhat Pham, Jessica Jacqueline Barnitz, Zhanan Zou, Phuc Nguyen, Hoang Truong, Taeho Kim, Nicholas Farrow, Anh Nguyen, Jianliang Xiao, Robin Deterding, Thang Dinh, and Tam Vu

Abstract

Frequent blood pressure monitoring is the key to diagnosis and treatments of many severe diseases. However, the conventional ambulatory methods require patients to carry a blood pressure (BP) monitoring device for 24 h and conduct the measurement every 10–15 min. Despite their extensive usage, wearing the wrist/arm-based BP monitoring device for a long time has a significant impact on users' daily activities. To address the problem, we developed eBP to measure blood pressure (BP) from inside user's ear aiming to minimize the measurement's impact on users' normal activities although maximizing its comfort level.

The key novelty of eBP includes (1) a light-based inflatable pulse sensor which goes inside the ear, (2) a digital air pump with a fine controller, and (3) BP estimation algorithms that eliminate the need of blocking the blood flow inside the ear.

Through the comparative study of 35 subjects, eBP can achieve the average error of 1.8 mmHg for systolic (high-pressure value) and −3.1 mmHg for diastolic (low-pressure value) with the standard deviation error of 7.2 mmHg and 7.9 mmHg, respectively. These results satisfy the FDA's AAMI standard, which requires a mean error of less than 5 mmHg and a standard deviation of less than 8 mmHg.

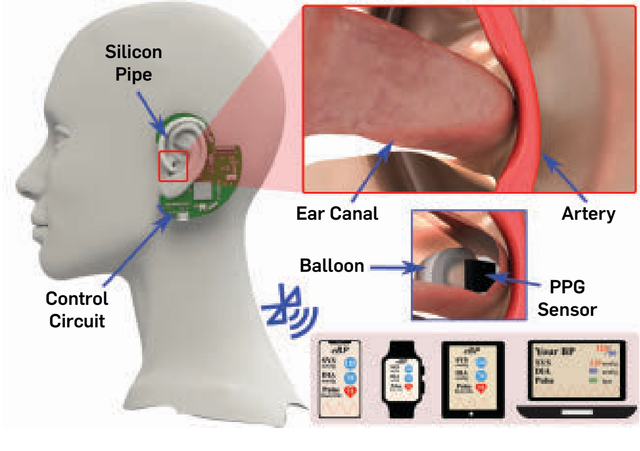
1. INTRODUCTION

BP provides doctors with insight to initiate their diagnosis. For example, chronic kidney disease, sleep apnea, and adrenal and thyroid disorders can all cause high BP, whereas low BP indicates the possibility of heart or endocrine problems, dehydration, severe infection, or even blood loss. Additionally, uncontrolled elevated BP is a major symptom of many life-threatening diseases, such as hypertension, heart failure or stroke.[4] Commonly, the reliable approach to measure BP was done by a health care practitioner using inflatable wrist cuff with a pressure gauge. Since the invention of digital BP devices, nonmedical trained users can self-measure their BP at home, as an acoustic sensor can replace the stethoscope, and a pressure sensor with a DC pump can substitute the pressure gauge and hand pump. However, these devices often cause discomfort and inconvenience for those who need **frequent BP monitoring**, such as hemodialysis (kidney failure) patients,[18] individuals with undiagnosed white coat hypertension or undiagnosed masked hypertension. There is also an increased use of frequent BP monitoring for postoperative organ transplant recipients. In such cases, BP is measured every 30 min for 24 h,[9] although each hemodialysis session takes around 4 h. Therefore, there is a significant need for an unobtrusive and comfortable BP monitoring approach. In the case of prolonged dialysis, patients hardly rest because the BP cuff constantly squeezes their arm and often hinders the wearer's mobility. Therefore, by moving the location of measuring BP to inside the ear, our device has a minimal impact in affecting the users' mobility and comfort.

In this paper, we aim to develop a novel wearable system to capture BP inside the ear called eBP, as illustrated in Figure 1. eBP resolves the aforementioned issues with its discreet design, quiet components, and convenient location.

eBP includes (1) a light-based pulse sensor attached to an in-ear inflatable pipe (or balloon), (2) an air pump, a pressure sensor, and a valve controlling module to control the balloon's contact to the in-ear skin for pulse measurement, and (3) a BP estimation algorithm. The in-ear pipe is slowly inflated by the digital pump to create small pressure on the outer ear canal until the diastolic and the systolic values are estimated.

Figure 1. eBP's overview.



The original version of this paper is entitled "eBP: A Wearable System For Frequent and Comfortable Blood Pressure Monitoring From User's Ear" and was published in *Proceedings of the 25th ACM Conference on Mobile Computing and Networking*, 2019, ACM, New York, NY, USA.

Challenges: Realizing eBP has the following challenges:

1. In-ear BP monitoring is an unexplored topic in which many of the existing techniques cannot be applied. Even the feasibility of the technique has not been confirmed.
2. The mechanism enabling the use of an inflatable balloon to measure BP from inside the ear is nontrivial. When the balloon inflates, the sensor should attach firmly to the ear canal and not slide out. In addition, applying insufficient pressure will result in an inaccurate BP measurement, although applying too much pressure may cause discomfort or hurt the ear canal.
3. The in-ear pulse signals are weak and buried under noises. In addition, the motion artifacts are difficult to remove and can impact BP measurement accuracy.
4. BP measurements are sensitive to the contact quality (i.e., pressure) between sensor and in-ear skin; yet maintaining consistent contact pressure is difficult.

Contributions: In this paper, we make the following contributions. First, we propose a novel concept of in-ear frequent BP monitoring. Second, we propose a blocking-free optical-oscillometric approach to allow the in-ear sensor to measure important parameters in BP measurements (i.e., systolic amplitude and diastolic amplitude). Third, we prototype a device with a custom-built circuit and hardware/software components for in-ear BP measurements. The light-based inflatable pulse sensor is built using an off-the-shelf catheter attached with a plethysmography (PPG) sensor.

2. FUNDAMENTALS OF BP MEASUREMENT

The section begins with a brief study of existing BP monitoring widely used nowadays. We will then point out its current limitations, setting a stage for our novel approach. Although invasive BP monitoring approach promises highly accurate results, it is costly and only available in clinics. Noninvasive techniques are far more favorable as their process is quick, low cost, and relatively simple. The noninvasive BP measurement relies on a technique called oscillatory.

Noninvasive BP requires an inflatable cuff squeezing around the arm or wrist to generate blood flow signatures. Based on these signatures captured by the pressure gauge, HCP can estimate the BP values. In particular, when the cuff pressure is equal to the systolic pressure (SBP), blood flow continues through the occluded artery, but only the highest arterial pressure can be detected. On the other hand, if the cuff pressure is lower than the diastolic pressure (DBP), the detected pulse is very weak. Oscillatory method was developed for the digital device to estimate BP from the change of pulse amplitude. It detects the maximum pulse amplitude (MAP) A_M first and applies predefined fractions of the peak amplitude ratio A_M/A_S and A_M/A_D to detect where the systolic and diastolic pressure occur and use these values to infer the pressure. A_S and A_D are the amplitude of systole and diastole, respectively. Unlike auscultatory methods, oscillatory methods do not need to completely occlude the blood vessel in order to detect the systolic BP,[10] which is well-suited for our balloon model. However, current oscillation ratios are only applicable for the arm or wrist BP measurement model. Therefore, they are not eligible for our in-ear case. Generating a new in-ear ratio requires a large-scale dataset, such as an invasive method to measure BP from inside the ear, which is infeasible. Instead, we propose a technique to measure BP without applying the characteristic ratios. To achieve this goal, we thoroughly examine the change of amplitude with respect to the change of cuff pressure. Then, we extract the key properties and formulate them into mathematical equations for processing. According to Guest commentary,[3] during the deflation:

- Pulse amplitude increases when the cuff pressure is close to the systolic level. The increment increases more quickly when the pressure reaches and passes through the systolic point.
- At the systolic and diastolic cycle cross section, the amplitude obtains its highest value (the MAP).
- Amplitude rapidly decreases once the pressure passes the MAP and moderately decreases once it reaches the diastole point. In other words, the DBP position occurs at the highest decreasing amplitude.

These observations provide key insights for composing the solutions to detect MAP, SBP, and DBP. In particular, the diastolic position is the minimum of the downslope amplitude, and MAP is the peak of the amplitude. We can derive the systolic location as being the maximum of the upslope amplitude. However, sometimes our in-ear balloon pressure might not reach the systolic phase due to comfort requirements. Therefore, we have to rely on the relational equation between MAP, SBP, and DBP[7]:

$$P_M = \beta P_S + (1-\beta)P_D, \tag{1}$$

where β is the systole ratio of the cardiac cycle and P_M, P_S, and P_D are the MAP, SBP, and DBP, respectively. Most literature reports β as a fixed value[14, 16] and is widely accepted, but each person can have a slightly different ratio dependent on age, gender, and health condition. Moreover, an incorrect estimation of β increases the estimation error, as noticed from Drawz et al.[19] In our eBP system, we propose an adaptive estimation for β based on the pulse-wave form.

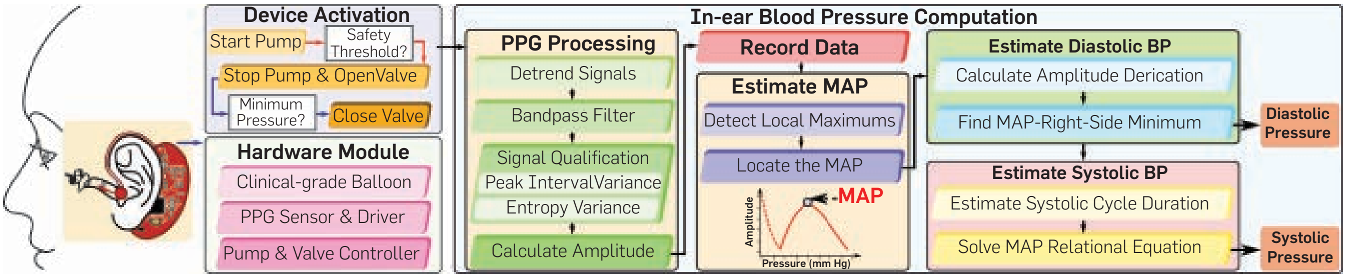
3. SYSTEM OVERVIEW

We designed our system as shown in Figure 2 to address the aforementioned challenges.

Nonratio approach for the calculation of systolic and diastolic BP. Unlike the oscillometric method, we do not apply the fixed-ratio BP because there is no valid ratio for inside the ear. For safety purposes, our pressure may not cover the SBP range. Therefore, we aim at estimating the pressure in diastole first according to its minimal downslope amplitude. Then, we substitute the DBP into the MAP-relational equation to estimate the SBP. Moreover, we propose a personalized approach to estimate the systolic fraction β instead of using the common fixed ratio.

In-ear pulse sensor with the flexible circuit. eBP uses the light-based sensing technique, named photoplethysmography

Figure 2. eBP system.



(PPG),[23] to capture the superficial pulse (BP value). The optical sensor is small and sustainable enough to be attached to the balloon. However, the state-of-the-art BP sensing technology is often designed on a printed hard circuit board. When the sensor is placed on the balloon, its surface might create sharp contact, which may hurt the user's ear. We overcome this problem by designing a flexible BP sensing circuit. This flexible circuit adapts to the balloon's deformation, making the device comfortable to use for a long period of time (Figure 3).

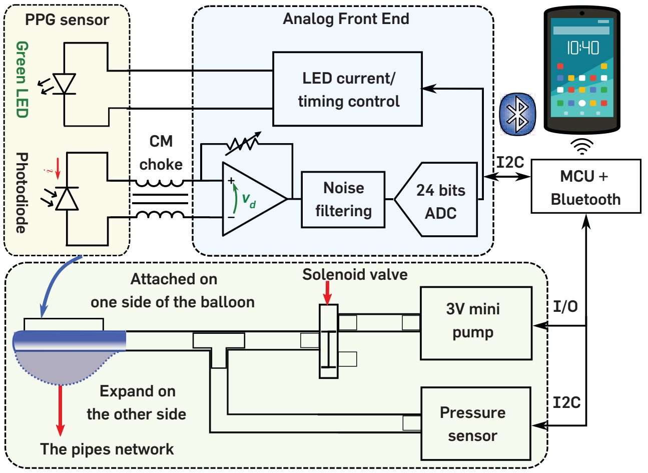
High-quality elastic balloon. The balloon, which serves our specific purpose, needs to satisfy the following criteria: biocompatibility, safety, high elasticity, consistency, and be strong and resilient to. To satisfy these conditions, we customize an off-the-shelf medical balloon often used for bladder catheterization.

In-ear PPG signal qualification. The in-ear PPG power is weaker than that of the finger, wrist, or arm. Therefore, after basic preprocessing, we employ two techniques for signal qualification, such as the **Peak Interval Variation** and **Entropy Variance**, to eliminate bad data chunks and identify the correct position inside the ear to place the sensor. For conventional signal filtering, we process every 50 ms with DC removal and a bandpass filter. This procedure helps to get rid of noise and other unwanted band signals, to disclose only the pulse waveform. With data that qualifies for this criteria, we calculate their amplitude using our modified peak-to-peak technique.

In-ear PPG signal processing. (1) *Modified peak-to-peak amplitude calculation*: Current peak-to-peak calculation is inconsistent for real-time processing due to the random order of peaks and bottoms. We propose a solution by adding a verification module to ensure the order consistency. (2) *Drift removal for Mean Arterial Pressure detection:* During the first few seconds of the balloon deflating, a large drift away from the calibrated pressure causes the false detection of maximum amplitude. We have developed a solution to detect the MAP based on its local maxima property regardless of the appearance of the drift.

Ear-worn air pump and draining components. Air pump and draining components are designed to inflate and shrink the balloon with predefined configuration. We target the miniaturized components to develop the air pump. The controller will process the signal and detect whether the pressure is sufficient. Then, following the information, it will decide if more air should be pumped in or if the valve should be opened to reduce pressure. The final product will be worn outside the ear.

Figure 3. In-ear BP module design.

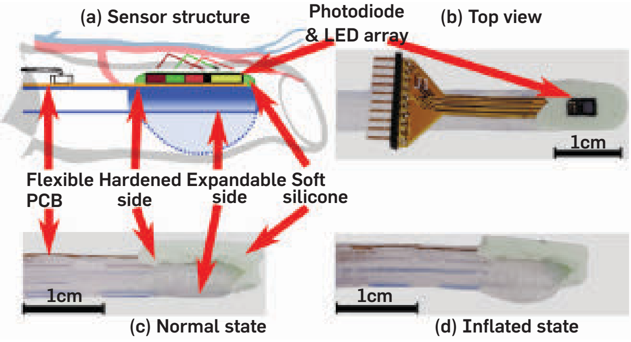


4. SYSTEM HARDWARE DESIGN

Proprietary design of the balloon attaching PPG sensor. The inflatable in-ear pulse sensor is designed by integrating a PPG sensor with the balloon of a Foley catheter made by POIESIS MEDICAL,[6] as illustrated in Figure 4. The Foley catheter is 100% medical silicon so it can be safely and comfortably inserted inside the body.[6] We found that SFH7050 PPG sensor from OSRAM[1] is the best fit for the small size of the ear canal. The sensor needs to be highly sensitive to capture the weak pulse signals from inside the ear and small enough to place into the ear canal. In particular, it has the size of 4.7 mm × 2.5 mm × 0.9 mm and performs highly accurate measurements due to its special design for the crosstalk blocking technique.[1]

The catheter balloon usually expands in all directions, easily breaking the connection between the balloon surface and the PPG sensor. Therefore, we improvise the balloon structure by hardening one side and only allowing it to expand on the other side. The PPG sensor is soldered on a thin layer (0.1 mm) of flexible PCB and then is integrated on the hardened side of the balloon catheter by using a thin layer of liquid silicone gel. After curing for 1 h at 80°C, the bonding between the PPG sensor and catheter surface becomes hardened and stays robust. Furthermore, to make

Figure 4. In-ear PPG sensor and balloon design.



the sensing unit more comfortable inside the ear, we coated Smooth-On Ecoflex 00-30 soft silicone[2] around the edge of the sensor, covering all sharp corners. The surface of the sensor was kept at by using a glass slide, which is removed once the Ecoflex is cured. Thus, the surface of the sensor offers a better sensing ability.

In-ear balloon pressure monitoring. The relationship among pressure inside the balloon, its volume, and diameter has been shown by experiments in literature to be nonlinear.[17] Especially when the diameter is in the range from 7 to 9 mm inside the ear canal, the pressure has an initial peak called the equilibrium point accompanied by a slow balloon expansion as the constituent polymer makeup of the balloon is altered. After the balloon has reached its equilibrium point, the pressure inside the balloon will keep stable or reduce, even if its volume increases. As a result, we cannot rely on pressure values to know whether the balloon has reached the wall of the ear canal or not. Instead, the quality of PPG signals is observed and the pump will be stopped when we observe clear PPG signals.

In addition, an over-threshold protection mechanism is implemented to stop pumping air when the pressure inside the balloon is over the threshold. As the size of each person's ear canal is different, with its diameters in the range from 2.4 to 17.5 mm, we want to continuously and slowly inject the air until one side of the balloon touches the skin of the user's ear canal and partially blocks the artery. However, the balloon also has its limit as to how much air it can hold. Thus, we do not want to inject too much air into it, making it permanently deformed or causing it to burst. From the balloon's specifications[6] and an experimental burst test from Mathis et al.,[17] the failure pressure of the silicone-based balloon is between 15 and 20 psi. Thus, the pressure inside the balloon is continuously monitored by the MCU and the pump will be stopped if the pressure reaches more than 10 psi, as a rule of thumb. This addresses the challenge of different ear canal sizes although maintaining the safety of our system.

Central processing controller. The central controller as shown in Figure 5 is responsible for (1) communicating to mobile devices through Bluetooth to receive commands and report sensing data, (2) driving the analog front-end IC to collect the PPG measurement to the sensor, and (3) controlling the pump/valves to control the balloon pressure for accurate PPG measurement. Overall, Figure 6 presents the eBP prototype depicting the integration of the in-ear pulse sensor with the main module.

Figure 5. eBP hardware.

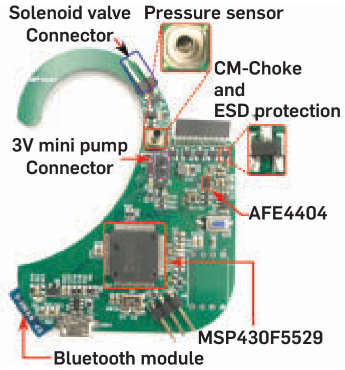
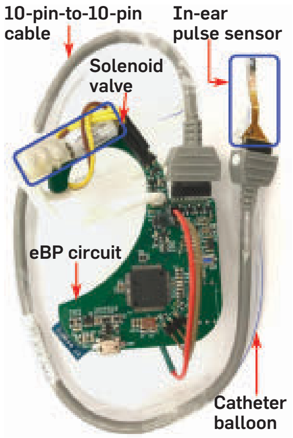


Figure 6. eBP prototype.



Power consumption. Low power consumption is an important requirement of a wearable device because it directly affects the user's mobility and comfort, which are the advantages over the conventional cuff-based BP measuring devices. Thus, all components in our designed prototype are chosen to operate with low-power consumption and also have small sizes. During a BP measurement, the MCU, AFE, pressure sensor, and Bluetooth module consume at maximum of only 4.6 mA, 325 uA, 1.7 mA, and 30 mA, respectively. The LED transmitter, valve, and mini pump draw 10 mA, 110 mA, and 150 mA, respectively. Thus, our module consumes around 303 mA although the BP measurement is running. On the other hand, only 4.95 mA is drawn when our measurement is not running. Thus, a 400 mA Li-Po battery could provide up to 1.3 h of continuous measurement (i.e., 80 measurements). However, running the device all the time is not practical or necessary. Instead, the users usually only need to measure their BP a few times per day. If the system is not running any measurement, it can last for more than 2 days (53 h) in an idle state.

5. IN-EAR BLOOD PRESSURE ESTIMATION ALGORITHMS

We propose our algorithm to handle blocking the artery inside the ear, which only blocks part of the artery and does not depend on the fixed BP ratio. In particular, eBP determines the MAP and DBP first from direct measurements and then infers SBP indirectly (Figure 7).[15]

5.1. Systolic BP measurement

We apply Equation 1 to estimate the SBP (P_S), given the pressure of MAP (P_M) and diastole (P_D). In addition, we propose an adaptive estimation for β which is the systole ratio of the cardiac cycle. We also notice that the derivation of Equation 1 will lead to the adaptive estimation of β. Therefore, we first present the mathematical model of obtaining the Equation 1 and then introduce our proposed formula to calculate β. We formulate MAP in one pulse cycle as follows: $P_M = \sum_{i=1}^{n} P(i)/n$ in discrete form or $P_M = \frac{1}{\tau}\int_0^\tau P(d)dt$ in continuous form. By assuming systole belongs to the interval (0, $\tau\beta$) and diastole is from ($\tau\beta$, τ), P_M is the total pressure average of systolic and diastolic pressure: $P_M = \frac{1}{\tau}\int_0^{\tau\beta} P(t)dt + \frac{1}{\tau}\int_{\tau\beta}^{\tau} P(t)dt$. Then, we multiply the first term and second term by β and 1 − β, respectively.

$$P_M = \beta\left[\frac{1}{\tau\beta}\int_0^{\tau\beta} P(t)dt\right] + (1-\beta)\left[\frac{1}{\tau(1-\beta)}\int_{\tau\beta}^{\tau} P(t)dt\right] \quad (2)$$

$\frac{1}{\tau\beta}\int_0^{\tau\beta} P(t)$ is the average of SBP and $\frac{1}{\tau(1-\beta)}\int_{\tau\beta}^{\tau} P(t)dt$ is the average of DBP. Note that Equation 1 is equivalent to Equation 2 by substituting $P_S = \frac{1}{\tau\beta}\int_0^{\tau\beta} P(t)dt$ and $P_D = \frac{1}{\tau(1-\beta)}\int_{\tau\beta}^{\tau} P(t)dt$. In one cycle, we detect the peak and two bottom points, and then subtract their position in sequence as shown in Figure 8. Δt_D, Δt_S, and t_C are the duration of diastolic, systolic, and the whole cycle, respectively. The systolic fraction is $\beta = \Delta t_S / \Delta t_C$. Given a frame of n cycles, we can compute β by averaging all β_i in the frame. As our system runs in real time, we only collect the first 10 cleanest frames to estimate the systolic fraction.

5.2. Mean arterial pressure detection

MAP represents the pulse pressure or the highest PPG amplitude. The precise location depends on the quality of the amplitude. This section introduces techniques to improve the MAP estimation by addressing the following issues: (1) precise peak-to-peak amplitude calculation and (2) removing the drift's effect.

Peak-to-peak amplitude calculation. The following equation provides a consistent estimation of each cycle's amplitude, denoted as amp_i, by suppressing the random order of appearance between the first peak and first bottom (Figure 9).

$$amp_i = \begin{cases} X(p_i) - X(b_i), & p_i < b_i \\ X(p_i) - X(b_i + 1), & otherwise \end{cases} \quad (3)$$

Figure 10 (a) demonstrates the PPG signal variation with respect to reducing pressure. The bottom panel displays a PPG signal sample from 150th to 170th s. Figure 10 (b) shows the corresponding amplitude using the peak-to-peak method.

Figure 7. Amplitude vs. pressure.

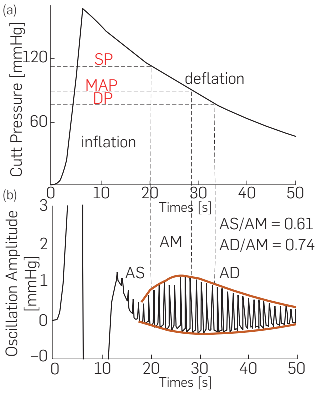


Figure 8. Systolic fraction β detection.

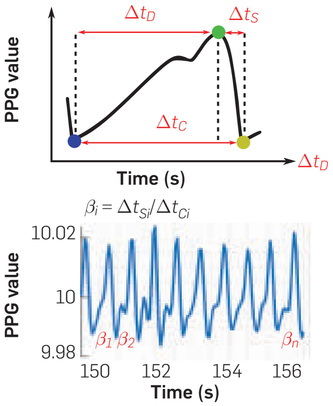


Figure 9. Illustration of the inconsistency of conventional peak-to-peak computation.

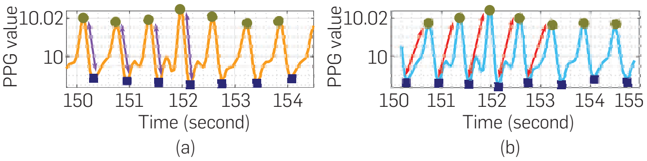
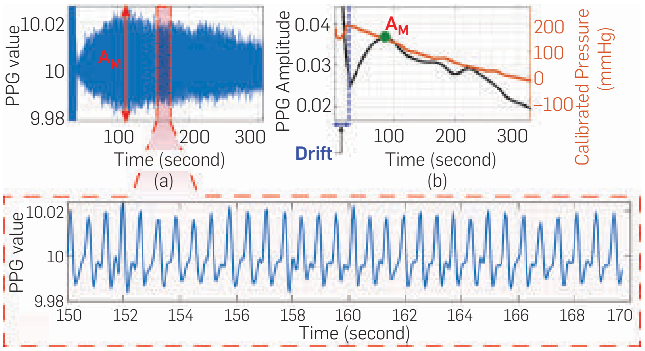


Figure 10. In-ear PPG signal (a) with corresponding amplitude and pressure (b).



Drift removal. To detect the correct MAP point instead of ones belonging to the drift, we impose an additional criterion leveraging the local maxima property. Specifically, MAP is not only the maximum amplitude point, it also indicates the pulse amplitude transient state of increasing to decreasing as shown in Figures 7[15] and 10. By contrast, points within the drift are not local maximums. Therefore, we employ the following steps to precisely detect the MAP:

1. Detect pulse amplitude's local maximums. This step confirms the removal of points belonging to the drift.
2. The highest value of local maximums corresponds to the MAP location.

5.3. Diastolic BP measurement

Based on the fundamental property of the DBP, which occurs at the highest decreasing amplitude, we formulate this as the minimum of the first derivative amplitude as shown in Figure 11. The dashed orange line represents the PPG amplitude, the solid purple is the first derivative, the dotted blue depicts the calibrated pressure, and the gray one is the PPG signal. In this example, the drift does not occur; thus, the MAP is the local maxima of the amplitude, in which its corresponding first-order derivative is approximately equal to 0. After the MAP, a rapid decrease is observed until the 43rd s, which corresponds to the minimal first order derivative and indicates the location of DBP.

6. EVALUATION

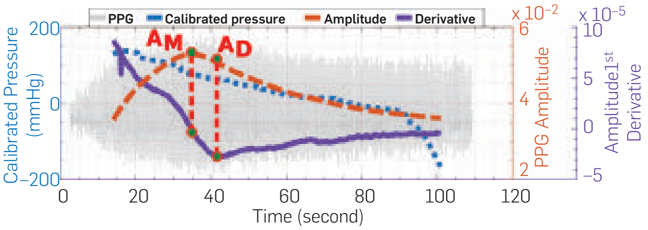
In this section, we present the set of experiments conducted to evaluate the overall performance of eBP and demonstrate the feasibility of using our BP device frequently in daily life. We first present the key results of performing BP measurements using the eBP system. Then, we evaluate different factors that can affect eBP's performance. Finally, we analyze the users' experience survey when using eBP.

6.1. Experimental methodology

We obtained the IRB approval to conduct experiments for the evaluation of eBP. The participant demographics is shown in the accompanying table. We tested eBP alongside an FDA-approved, gold standard, arm-cuff BP measurement device (KonQuest KBP-2704A[5]) (Figure 12). For assessment, we use the metric that is widely accepted by other BP studies, which consists of bias or mean error μ, a precision or standard deviation (SD) σ error, and a Pearson correlation coefficient ρ.

This experiment is tested on 35 participants of both genders and various ages. eBP participants place the in-ear balloon inside their ear. Next, the cuff of the KonQuest device is wrapped around the upper arm of each participant. We simultaneously measure the BP of each participant from our Android app running on the Samsung Galaxy S9 and the gold standard BP device. This process is repeated twice and takes about 20 min. We sterilize our device with an alcohol wipe, between each experiment, by softly cleaning the balloon tip and sensor. During the experiment, the participant has to sit still to ensure the BP reading is correct. In addition, the balloon needs to be mounted in the right position so that it will not fall out. It turns out that the ear can hold the sensor properly because the tragus helps to keep the sensor tight as shown in Figure 12.

Figure 11. First derivative of PPG amplitude discloses diastolic BP.



6.2. System performance

In this section, we evaluate eBP performance and showcase the comparative results between eBP and the Kon-Quest KBP-2704A.

Figure 13 shows the Bland-Altman diagram that describes

Demographic description of participants.

Demographic data of study population	
Age (years)	18–35 years old
Blood pressure	Systolic: 93–146, Diastolic: 53–113
Gender ratio	Male: 24, Female: 11

Figure 12. Experiment setup to compare eBP with the Kon-Quest device.

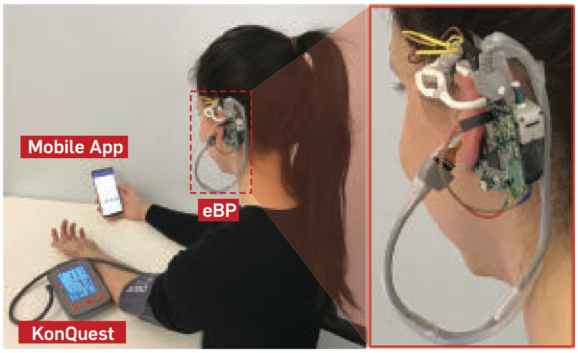
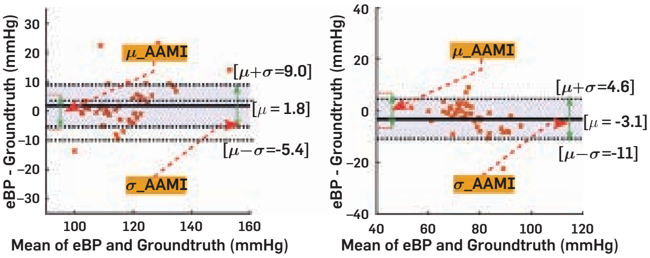


Figure 13. Bland-Altman plot comparing eBP's measurements and groundtruth.



(a) SBP (b) DBP

the average error between eBP and the ground-truth. Consequently, the mean and SD error of SBP and DBP are 1.8 mmHg and 3.1 mmHg, which is within the Association for the Advancement of Medical Instrumentation's (AAMI) requirement ($\mu_{AAMI} < 5$ mmHg).[24] In addition, our SD errors for SBP and DBP also satisfy the criteria where $\sigma_{AAMI} < 8$ mmHg.[24] On the other hand, Figure 14 displays the Pearson correlation coefficients of SBP and DBP measurements by eBP and the KonQuest device. We select five participants' data for calibration using a polynomial regression model. There were some error cases where the measurement was performed without taking sufficient stability. For example, the balloon fell out of the ear because of sweat, movement, or the ear canal was too narrow. When the balloon falls out, there is no valid pulsatile waveform detected. As a result, the system cannot predict the BP, thus, providing no data for the evaluation. The correlation shown of 0.81/1.0 for the SBP and 0.76/1.0 for the DBP represents that our system's prediction is highly correlated to that of the FDA approved device.

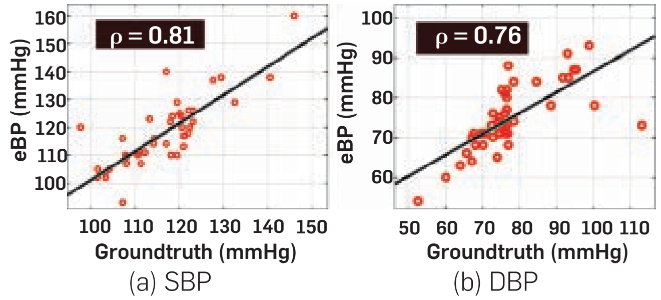
6.3. Power consumption

We measure the power consumption of both eBP hardware module and eBP app (installed on a Samsung Galaxy S9) in two scenarios: (1) during BP measurement and (2) without BP measurement. eBP hardware module power consumption is measured using Monsoon Power Monitor. eBP app power consumption is measured using AccuBattery application. Note that the power measurement of eBP hardware is done in 1 min, whereas it takes 9 min to obtain a reliable measurement from AccuBattery app. eBP hardware consumes 1279.28 and 31.34 mW during BP measurement and without BP measurement, respectively. eBP app consumes 1406 and 1119 mW during eBP measurement and without BP measurement, respectively. In summary, eBP hardware consumes 1247.94 mW (1279.28–31.34 mW) to operate the pump, the valve, the LED, and the microcontroller. eBP app consumes 287 mW (1406–1119 mW) for BP calculation.

6.4. Prediction stability

We conducted experiments to verify the robustness of our calibration procedure based on polynomial fitting. We replicated the process by taking 250 randomly picked times from the learning set. Finally, we explore the frequencies of mean and SD error as shown in Figure 15. Overall, the highest frequencies of both SBP and DBP mean error falls between 4 and 5 mmHg, which satisfies AAMI standards. Similarly, the highest frequency of SD errors is less than 8 mmHg, which also qualifies the AAMI protocol. In addition, 9 out of 35 candidates proceed 10 times of data collection to calculate the intraclass correlation coefficient (ICC). Figure 16 shows the ICC result of each candidate. The average ICC of SBP and DBP are 0.8 and 0.76, respectively.

Figure 14. Pearson correlation coefficients of eBP's estimation and groundtruth.



(a) SBP

(b) DBP

We refer the readers to the works of Bui et al.[8] for more detailed validations of optimal sensor locations, user study, and our discussion on the limitations of eBP.

7. CONCLUSION

In this paper, we presented eBP, a new method to capture BP from inside the ear, which measures the artery BP from the superficial artery near the ear canal. Existing techniques that measure BP on the arm or wrist cannot be applied to measure BP from inside the ear as the required fixed systolic and diastolic detection ratios. We developed our model to estimate the in-ear BP by observing the behavior of pulse

Figure 15. Mean and SD error in cross validation.

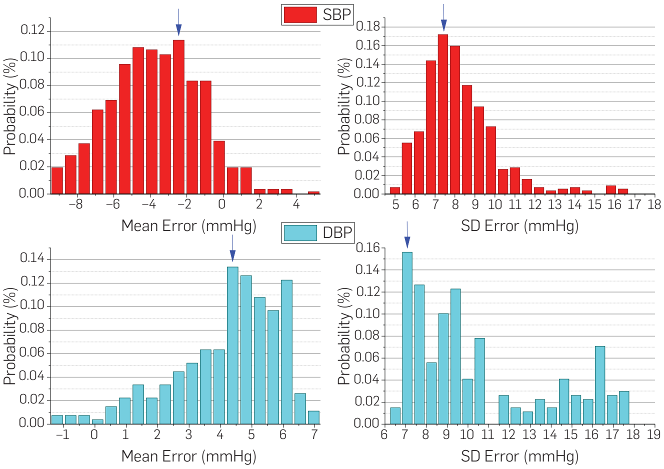
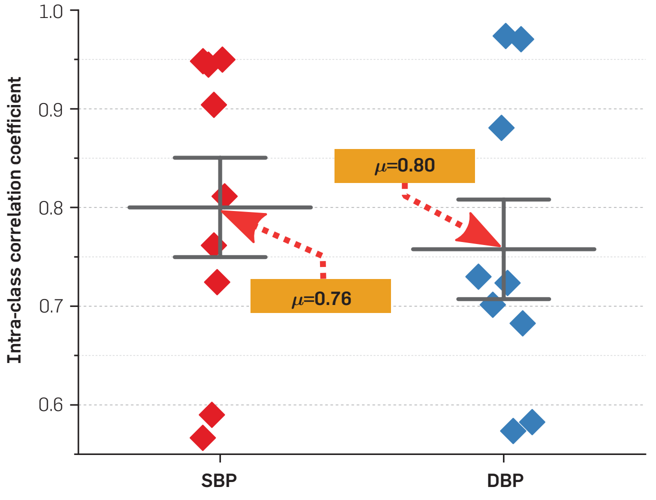


Figure 16. Intraclass correlation coefficient of eBP and groundtruth.



amplitude. Therefore, no constant parameters are required in our proposed model. In this paper, we also introduce a technique to customize an off-the-shelf catheter to become our in-ear pulse pressure sensor. We built custom hardware and software for eBP and evaluated the system through a comparative study on 35 subjects. The study shows that eBP obtains an average error of 1.8 and −3.1 mmHg and standard deviation error of 7.2 and 7.9 mmHg for systolic (high-pressure value) and diastolic (low-pressure value), respectively. These errors are within the acceptable margins regulated by FDA's AAMI protocol, which allows average BP difference of up to 5 mmHg and standard deviation of up to 8 mmHg. These promising results not only show the feasibility of an in-ear blood monitoring concept but also open up the possibility of making current gold standard cuff-based BP measurement more comfortable.

Broaden applications. Although eBP is currently a standalone device, with the continued trend of incorporating biometric monitoring into devices[12, 21] that are worn on a daily basis, there would be minimal behavioral changes required on the part of the wearer to benefit from eBP. As ear-worn sensing platforms such as Earable[22] and eSense[13] are becoming increasingly popular, the BP monitoring modality could potentially be integrated into these platforms. It enables the ability, which is not possible before, to sense and react to various dangerous diseases and conditions in daily life such as hypertension, epileptic seizures, etc. Additionally, eBP could also be incorporated with a headphone or hearing aid, both of which are ubiquitous as the World Health Organization reports that approximately 466 million people worldwide suffer from disabling hearing loss[20] and more than 365 million headphones were sold in 2017 in the US alone.[11] In addition, our proposed BP calculation algorithm can be applied to make existing cuff devices more comfortable. In the case of hardware design, the use of a medical balloon to deliver a sensor into the ear can widely benefit other applications. For example, it can improve the contact points and the conductivity of electrodes for the in-ear sensing area.

Acknowledgments

This research is based in part upon work supported by NSF CNS/CSR 1846541, NSF SCH 1602428, Google Faculty Awards 2018, the Schramm Foundation and the Colorado Advanced Industries Accelerator (AIA).

References

1. Biomon sensor sfh7050, osram opto semiconductors. https://tinyurl.com/y7g6yrbn.
2. Ecoflex 00–30, smooth-on. https://tinyurl.com/y77pfgnun.
3. Guest commentary: How blood-pressure devices work. https://tinyurl.com/y5rmewuf.
4. Hbp and the cardiovascular system. https://tinyurl.com/ybv4vlst.
5. Konquest kbp-2704a. https://tinyurl.com/y4njpa4s.
6. Poiesis medical duettetm dual-balloon 2-way urinary catheter. http://www.poiesismedical.com/products/duette/.
7. Baker, P.D., Westenskow, D.R., Kück, K. Theoretical analysis of non-invasive oscillometric maximum amplitude algorithm for estimating mean blood pressure. *Med. Biol. Eng. Comput. 35*, 3 (May 1997):271–278.
8. Bui, N., Pham, N., Barnitz, J.J., Zou, Z., Nguyen, P., Truong, H., Kim, T., Farrow, N., Nguyen, A., Xiao, J., et al. ebp: A wearable system for frequent and comfortable blood pressure monitoring from user's ear. In *The 25th Annual International Conference on Mobile Computing and Networking* (2019), Association for Computing Machinery (ACM), New York, USA, 1–17.
9. Drawz, P.E., Abdalla, M., Rahman, M. Blood pressure measurement: clinic, home, ambulatory, and beyond. *Am. J. Kidney Dis: The Official Journal of the National Kidney 60*, 3 (Apr. 2012), 449–462.
10. Geddes, L.A., Voelz, M., Combs, C., Reiner, D., Babbs, C.F. Characterization of the oscillometric method for measuring indirect blood pressure. *Ann. Biomed. Eng. 10*, 6 (Nov. 1982), 271–280.
11. GfK. Global unit sales of headphones and headsets from 2013 to 2017 (in millions), 2019. www.statista.com/statistics/327000/ worldwide-sales-headphones-headsets.
12. Hester, J., Peters, T., Yun, T., Peterson, R., Skinner, J., Golla, B., Storer, K., Hearndon, S., Freeman, K., Lord, S., Halter, R., Kotz, D., Sorber, J. Amulet: An energy-efficient, multi-application wearable platform. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems* (2016), Association for Computing Machinery (ACM), New York, USA, 216–229.
13. Kawsar, F., Min, C., Mathur, A., Van den Broeck, M., Acer, U.G., Forlivesi, C. esense: Earable platform for human sensing. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services* (2018), Association for Computing Machinery (ACM), New York, USA, 541–541.
14. Lee, S., Jeon, G., Lee, G. On using maximum a posteriori probability based on a bayesian model for oscillometric blood pressure estimation. *Sensors (Basel, Switzerland) 13*, 10 (2013), 13609–13623.
15. Liu, J., Hahn, J.-O., Mukkamala, R. Error mechanisms of the oscillometric fixed-ratio blood pressure measurement method. *Ann. Biomed. Eng. 41*, 11 (2012), 587–597.
16. Mafi, M., Rajan, S., Bolic, M., Groza, V.Z., Dajani, H.R. Blood pressure estimation using maximum slope of oscillometric pulses. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Aug. 2012), Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 3239–3242.
17. Mathis, J.M., Barr, J.D., Jungreis, C.A., Horton, J.A. Physical characteristics of balloon catheter systems used in temporary cerebral artery occlusion. *Am. J. Neuroradiology 15*, 10 (1994), 1831–1836.
18. Miskulin, D.C., Weiner, D.E. Blood pressure management in hemodialysis patients: What we know and what questions remain. *Semin. Dialysis 30*, 3 (2017), 203–212.
19. Moran, D., Epstein, Y., Keren, G., Laor, A., Sherez, J., Shapiro, Y. Calculation of mean arterial pressure during exercise as a function of heart rate. *Appl. Hum. Sci.: J. Physiol. Anthropol. 14*, 12 (1995), 293–295.
20. World Health Organization. Deafness and hearing loss, 2018. https://www.who.int/en/news-room/fact-sheets/detail/deafness-and-hearing-loss.
21. Pham, N., Dinh, T., Raghebi, Z., Kim, T., Bui, N., Nguyen, P., Truong, H., Banaei-Kashani, F., Halbower, A., Dinh, T., et al. Wake: A behind-the-ear wearable system for microsleep detection. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services* (2020), Association for Computing Machinery (ACM), New York, USA, 404–418.
22. Pham, N., Kim, T., Thayer, F.M., Nguyen, A., Vu, T. Earable–an ear-worn biosignal sensing platform for cognitive state monitoring and human-computer interaction. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (2019), Association for Computing Machinery (ACM), New York, USA, 685–686.
23. Shelley, K., Shelley, S. *Pulse Oximeter Waveform: Photoelectric Plethysmography*, 01, Walter Burns Saunders (W. B. Saunders), Philadelphia, USA, 2001, 420–423.
24. Tao, G., Chen, Y., Wen, C., Bi, M. Statistical analysis of blood pressure measurement errors by oscillometry during surgical operations. *Blood Pressure Monit. 16*, 6 (Dec. 2011), Phil Daly, London, United Kingdom.

Nam Bui, Jessica Jacqueline Barnitz, Zhanan Zou, Hoang Truong, Taeho Kim, Nicholas Farrow, Anh Nguyen, and Jianliang Xiao ({Nam.Bui, jessica.barnitz, Zhanan.Zou, Hoang.Truong, Taeho.Kim, Nicholas.Farrow, Anh.TL.Nguyen, jianliang.xiao}@colorado.edu), University of Colorado Boulder, Boulder, CO, USA.

Nhat Pham ({nhat.pham}@wolfson.ox.ac.uk), University of Oxford, Oxford, U.K.

Phuc Nguyen ({vp.nguyen}@uta.edu), University of Texas at Arlington, Arlington, TX, USA.

Robin Deterding ({Robin.Deterding}@childrenscolorado.org), Children's Hospital Colorado, Boulder, CO, USA.

Thang Dinh ({tndinh}@vcu.edu), Virginia Commonwealth University, Richmond, VA, USA.

Tam Vu ({Tam.Vu}@cs.ox.ac.uk), University of Oxford, Oxford, U.K./University of Colorado Boulder, Boulder, CO, USA.

CAREERS

Worcester State University

Assistant or Associate Professor of Computer Science (2 Positions)

About Worcester State University:
Learn more about us at
https://www.worcester.edu/about/ .

Job Description:
The Computer Science Department at Worcester State University invites applications for two tenure-track Assistant or Associate Professor positions with a start date of September 2022. We are looking for faculty dedicated to innovative, inclusive, and engaging pedagogy using high-impact practices and authentic learning.

Position 1: Software Development/Software Engineering, with expertise and ability (or willingness to learn) to teach courses in team software processes (in particular Scrum and Agile), version control, CI/CD, containerization, software design and architecture, software testing, Web APIs, and full stack development.

Position 2: Systems and Security, with expertise and ability (or willingness to learn) to teach courses in computer organization and architecture, Unix systems programming (including shell scripting), computer security/information assurance, and system and network administration.

Expectations:

- Teaching 12 credits (4 courses) per semester, usually 2-3 preparations
- Ability to teach common lower-division courses as needed, such as:
- Breadth-first Introduction to Computer Science
- Programming for Non-Majors
- Introduction to Programming
- Data Structures
- Unix Systems Programming
- Database Design and Applications
- Discrete Structures
- Teaching using high impact practices.
- Advising of major and minor students.
- Involvement in Departmental and University committees.
- Research and scholarship in the area of the candidate's expertise. Research in the Scholarship of Teaching and Learning is valued. The ability to involve undergraduate students in research is highly desirable.

Requirements:
Applicants must possess a Ph.D. in Computer Science or a terminal degree in a closely related field, conferred by September 1, 2022. Teaching experience at the college/university level is desirable. Professional experience outside of academia is desirable. Salary and rank commensurate with experience.

Additional Information:
The WSU CS Department is a vibrant department with about 220 major and 30 minor students. The Computer Science major has two concentrations-Big Data Analytics and Software Development, and two minors - Computer Science and Data Science. In addition, the department has numerous connections with other departments on campus and is an active participant in general education.

Worcester State University is an Equal Opportunity/Affirmative Action Employer. M/F/D/V. Women and minorities are strongly encouraged to apply.

Review of applications will begin early in the Fall 2021 semester.

Application Instructions:
All applicants must apply online through Interview Exchange: https://worcester.interviewexchange.com. Please submit a letter of interest, CV, and the contact information for three professional references who will write a letter of recommendation on your behalf.

Information which cannot be uploaded by the applicant may be faxed to 508-929-8163 or mailed to:

Executive Director, Chief Human Resources
Officer
Worcester State University
486 Chandler Street
Worcester MA 01602-2597

*Proof of highest degree is required upon hire, by means of verification through the National Student Clearinghouse, or by official, sealed transcript.

[CONTINUED FROM P. 128]

rules: it plays vast numbers of times and sees what improves its score."

"That's great for playing Breakout," said Lipcott, unconvinced by Carter's defense. "But what about Dupin? How did it learn to be so successful?"

"The same way. It was fed vast numbers of case details and worked out the rules for itself. You'll see when you read the book properly that in the early days of AI they tried to list all the rules for the computer, but it never worked well enough for a complex task like detection. Machine learning totally beats anything we can teach the system."

"Okay, sure. But think of that line…" Lipcott searched the page in front of her. "'*Managing to win in a way the game designers never envisaged.*' If you translate that into what Dupin does, isn't that saying, 'managing to convict someone in a way that the law is not intended to work?' Could Dupin break the rules to win?"

"Not at all," said Carter. "Look, in the old days we relied on witness statements, even though they were useless. Have you heard of the 1901 Stern experiment?"

Lipcott shook her head.

"This German professor faked a murder in his lecture room and then asked the students, who thought the crime was real, to write accounts of what they saw. They named eight different people as the murderer. We rely on solid evidence. Of course Dupin considers witness statements and monitors all conversations in this building to include interviews and our thinking on the case, but it's Dupin's ability to access substantial evidence that makes it so effective."

"I understand that. But this is an artificial intelligence that made up its own rules. What's to say that the video isn't a deepfake, or the location data corrupted? Do we really know what Dupin is capable of?"

"Let's get a coffee." Carter headed down the corridor. "It's hard to accept at first—it feels like we're not real investigators anymore. But at least we've got jobs. Think of everyone replaced by machine-learning systems. The fact is, Dupin does the spade work, and we can concentrate on what humans do best. Interpreting outcomes and interacting with people."

How can we trust an algorithm that doesn't know the rules?

"I get it," said Lipcott. "But how can we trust an algorithm that doesn't know the rules?"

"You'll get used to it," said Carter. A noisy fanfare came from her pocket. "Sorry, I should change that alert. It's the Dupin app. Looks like we've got another case."

She pulled her phone out as she walked and flicked up the details. "This is what I mean. There was a homicide just five minutes ago in E street, and Dupin has already cracked it. Have you ever been there?"

Lipcott shook her head.

"It's where the DC medical examiner is based. It's our very own Rue Morgue."

"I'm sorry?"

"The Rue Morgue in Paris. That's where Poe's Dupin solved his first crime." Carter flicked at her phone again. "The details are coming through."

For a moment, Carter seemed to slip on the polished floor of the old FBI building. The Special Agent stopped dead and put her hand on the wall to steady herself. In a flashing red rectangle, the app was presenting her with a case summary:

Suspect: *Saskia Lipcott.*
Crime: *Homicide (first degree).*
Probability of conviction: *99.99%.*

"What is it?" asked Lipcott.

Carter shook her head and stared at the combined camera and microphone that surveyed the corridor. Lipcott's words seemed to float in front of her eyes. What she did next could determine not just Lipcott's future, but her own.

She walked on to the corner, to a dead spot between cameras, took a deep breath, and mouthed, "Don't ask questions. Get ready to run."

Brian Clegg (www.brianclegg.net) is a science writer based in the U.K. His most recent books are *What Do You Think You Are?*, exploring the science of what makes you *you*, and *Quantum Computing*, offering background to this new computing paradigm.

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/3470629

Brian Clegg

Future Tense
Agent Algorithm

Crime-solving computer plays by its own rules.

AGENT SASKIA LIPCOTT grimaced at her reflection in the mirror glass, wondering again if the look she'd gone for had too much of Dana Scully from "The X-Files." "I don't get it," she said. "Why call the computer Duppin? It sounds like a kid's game."

"Not Duppin," said Special Agent Dinah Carter. "That's Doo-pan. It's from Edgar Allan Poe. He wrote the first detective stories, about a Frenchman called Auguste Dupin."

"Okay?" Lipcott sounded uncertain.

"And it's not the name of the computer; it's the machine-learning software. Didn't they give you the AI guide?"

Lipcott, who had grown up on a farm and had a very different idea of what AI stood for, raised her eyebrows. "Nope."

Carter opened her desk drawer, took out a thin book, and threw it across. *The AI Primer—Artificial Intelligence from DeepMind to Dupin.* "Let's move," she said to Lipcott. "We need to interview the suspect. It's a formality. Dupin tells us he's guilty, but it has to be done."

Lipcott followed Carter into the interview room. It was far less impressive than the ones she had seen in TV dramas—just an ordinary, bare office. A middle-aged man with thinning brown hair, wearing a sweater and jeans, was seated at a metal-topped table.

Carter drew up one of the two chairs on their side of the table and indicated Lipcott should take the other.

"Mr. Lamb, I am Special Agent Carter and this is Agent Lipcott. The purpose of this meeting is to put before you the evidence provided that will be used in your trial. You have refused legal representation. Any comments you make will be recorded and added to the evidence base. Do you understand?"

IMAGE BY LUKERUK/SHUTTERSTOCK

Lamb nodded.

Carter touched her tablet screen. "We have video placing you at the scene of an armed robbery last Monday at a jewelry store on Connecticut Avenue Northwest, between Desales and L streets. You are clearly seen committing the crime. This evidence is backed up by location data from your cellphone, showing that you travelled from your apartment to the store, arriving at the exact time the crime was committed. DNA evidence from the shotgun makes it clear that you were involved.

"Do you have anything to say?"

"Only what I've said all along," said Lamb. "I was at home all evening. I never left the apartment. I did not commit a crime. I was alone, but I spoke several times to my smart speaker."

Carter countered. "Though the system has no record of this. Do you have anything else to add?"

Lamb shrugged. "What's the point? It's a stitch-up."

"Thank you for your cooperation," said Carter. She nodded to the door. "Okay Lipcott, I'll see you outside. I just need to authorize the suspect's removal."

"That's it?" said Lipcott a few minutes later, as Carter emerged into the corridor. Lipcott had been flicking through the AI primer and now pointed at a paragraph. "Have you read this about artificial intelligence learning to play games?"

"Sure, I've read it," said Carter. "They're real good at games. That's why Dupin's perfect; detection is a kind of game. Isn't that why you became an investigator?"

"I guess," said Lipcott. "But listen to this. *'The remarkable success of AI systems is even more impressive because they were never given the rules of the game. Initially, their attempts were random failures, but over time, the machine-learning algorithm picked up strategies that were successful, often discovering loopholes in the game software and so managing to win in a way that the game designers never envisaged.'*

"Isn't that worrying?" Lipcott pressed.

"Why?" asked Carter. "It shows how clever the software is. It doesn't need to know the

[CONTINUED ON P. 127]

SIGGRAPH ASIA 2021 TOKYO

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

sa2021.siggraph.org