

Digital Creativity Support for Original Journalism

The decline in circulations and revenues resulting from the digitalization of news production and consumption has led to a crisis in journalism. Journalists have less time to research, investigate, and write original stories, leading to problems for our democratic processes and holding the powerful to account. A new digital creativity tool helps journalists discover more original story angles.

INJECT

journalism × crisis × digitalization ×

Technology
Digital Humans on the Big Screen

Viewpoint
OMSCS: The Revolution Will Be Digitized

Contributed
Why Computing Belongs in the Social Sciences

Last Byte
Q&A with Elisa Bertino

INJECT SUGGESTS

INJECT

Other angles to explore

Think about new angles using one or more of these related topics:

- [Generate associations](#)
- [Evaluate story angles](#)
- [Deliver digital editorial support](#)

Can you use their backgrounds, histories, or data about them or evidence that others have reported?

INJECT

volume
01

number
01

FIRST
ISSUE
PUBLISHED

ACM/IMS Transactions on Data Science is now available in the ACM Digital Library



ACM/IMS Transactions on Data Science (TDS) publishes cross-disciplinary innovative research ideas, algorithms, systems, theory and applications for data science. Papers that address challenges at every stage, from acquisition on, through data cleaning, transformation, representation, integration, indexing, modeling, analysis, visualization, and interpretation while retaining privacy, fairness, provenance, transparency, and provision of social benefit, within the context of big data, fall within the scope of the journal.



*Making Waves,
Combining Strengths*

CHI 2021



chi2021.acm.org

**May 8-13, 2021
Yokohama, Japan**

Departments

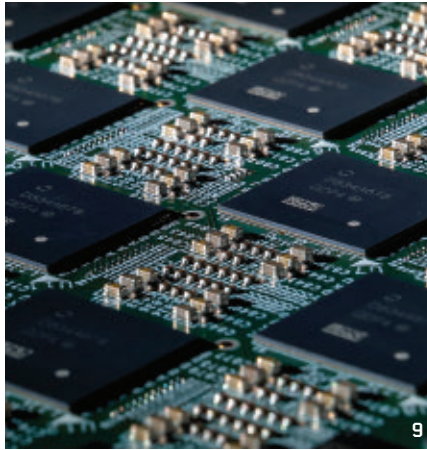
- 5 **Cerf's Up**
On the Internet of Medical Things
By Vinton G. Cerf

- 6 **BLOG@CACM**
How WWII Was Won, and Why CS Students Feel Unappreciated
John Arquilla considers how code-breaking helped end a war, while Jeremy Roschelle ponders the use of music in data science education.

Last Byte

- 104 **Q&A**
Seeing Light at the End Of the Cybersecurity Tunnel
After decades of cybersecurity research, Elisa Bertino remains optimistic.
By Leah Hoffmann

News



- 9 **Neuromorphic Chips Take Shape**
Chips designed specifically to model the neurons and synapses in the human brain are poised to change computing in profound ways.
By Samuel Greengard
- 12 **Digital Humans on the Big Screen**
Motion pictures are using new techniques in computer-generated imagery to create feature-length performances by convincingly “de-aged” actors.
By Don Monroe
- 15 **Are We Addicted to Technology?**
Experts agree that technology causes some negative behaviors, but they're divided on how bad the problem is.
By Logan Kugler

Viewpoints

- 18 **Broadening Participation**
TECHNOLOchicas: A Critical Intersectional Approach Shaping the Color of Our Future
A unique partnership seeks to address the underrepresentation and unique barriers facing Latina women and girls of color in information technology.
By Jannie Fernandez and JeffriAnne Wilder
- 22 **Kode Vicious**
Broken Hearts and Coffee Mugs
The ordeal of security reviews.
By George V. Neville-Neil
- 24 **Education**
Data-Centricity: A Challenge and Opportunity for Computing Education
Rethinking the content of introductory computing around a data-centric approach to better engage and support a diversity of students.
By Shriram Krishnamurthi and Kathi Fisler
- 27 **Viewpoint**
OMSCS: The Revolution Will Be Digitized
Lessons learned from the first five years of Georgia Tech's Online Master of Science in Computer Science program.
By Zvi Galil
- 30 **Viewpoint**
Thorny Problems in Data (-Intensive) Science
Data scientists face challenges spanning academic and non-academic institutions.
By Michael J. Scroggins, Irene V. Pasquetto, R. Stuart Geiger, Bernadette M. Boscoe, Peter T. Darch, Charlotte Cabasse-Mazel, Cheryl Thompson, Milena S. Golshan, and Christine L. Borgman



Practice



41

- 34 **To Catch a Failure:
The Record-and-Replay Approach
to Debugging**
*A discussion with Robert O'Callahan,
Kyle Huey, Devon O'Dell,
and Terry Coatta*

- 41 **Power to the People**
Reducing datacenter
carbon footprints.
By Jessie Frazelle

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



About the Cover:
With round-the-clock news and competition for readers, journalists are feeling deadline pressures like never before. This month's cover story introduces a digital creativity tool to support journalistic efforts. Cover collage by Andrij Borys Associates.

IMAGES IN COVER COLLAGE: Protest photo by Wade Jackman/Shutterstock.com. Additional stock images from Shutterstock.com.

Contributed Articles

- 46 **Digital Creativity Support
for Original Journalism**
A tool that helps journalists discover new story angles by offering insight not search results.
By Neil Maiden, Konstantinos Zachos, Amanda Brown, Dimitris Apostolou, Balder Holm, Lars Nyre, Aleksander Tonheim, and Arend van den Beld



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

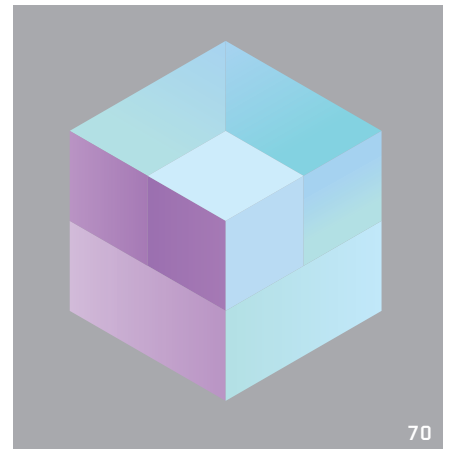
- 54 **Why Computing Belongs
Within the Social Sciences**
Fully appreciating the overarching scope of CS requires weaving more than ethics into the reigning curricula.
By Randy Connolly



Watch the author discuss this work in the exclusive *Communications* video.
<https://cacm.acm.org/videos/computing-social-sciences>

- 60 **Examining Undergraduate
Computer Science Participation
in North Carolina**
Data on CS graduation rates among six academic institutions in NC traces the demographics of those participating (or not) in the discipline.
By Fay Cobb Payton and Alexa Busch

Review Articles



70

- 70 **Threats of a Replication Crisis
in Empirical Computer Science**
Research replication only works if there is confidence built into the results.
By Andy Cockburn, Pierre Dragicevic, Lonni Besançon, and Carl Gutwin

Research Highlights

- 82 **Technical Perspective**
Entity Matching with Magellan
By Wang-Chiew Tan
- 83 **Magellan: Toward
Building Ecosystems of
Entity Matching Solutions**
By AnHai Doan, Pradap Konda, Paul Suganthan G.C., Yash Govind, Derek Paulsen, Kaushik Chandrasekhar, Philip Martinkus, and Matthew Christie
- 92 **Technical Perspective**
**Supporting Linear Algebra
Operations in SQL**
By Yannis Papakonstantinou
- 93 **Scalable Linear Algebra on
a Relational Database System**
By Shangyu Luo, Zekai J. Gao, Michael Gubanov, Luis L. Perez, Dimitrije Jankov, and Christopher Jermaine



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
Mehran Sahami and Jane Chu Prey
Practitioners Board
Terry Coatta

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

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

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

Computer Science Teachers Association
Jake Baskin
Executive Director

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.

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

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

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

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

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

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

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

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

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

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

Media Kit acmm mediasales@acm.org

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

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

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

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

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

CONTRIBUTED ARTICLES
Co-Chairs
James Larus and Gail Murphy
Board Members
Robert Austin; Kim Bruce; Alan Bundy;
Peter Buneman; Jeff Chase;
Premkumar T. Devanbu; Jane Cleland-Huang;
Yannis Ioannidis; Trent Jaeger; Somesh Jha;
Gal A. Kaminka; Ben C. Lee; Igor Markov;
Lionel M. Ni; Doina Precup; Shankar Sastry;
m.c. schraefel; Ron Shamir; Hannes Werthner;
Reinhard Wilhelm

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

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

ACM Copyright Notice
Copyright © 2020 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from permissions@hq.acm.org or fax (212) 869-0481.

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

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

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

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

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

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

Printed in the USA.





Vinton G. Cerf

DOI:10.1145/3406779

On the Internet of Medical Things

In my last column (June 2020), I wrote about my experience with COVID-19 and the challenges involved with getting medical attention. The problem is still with us, even

with the improved availability of personal protection equipment and masks. The experience of calling for a doctor's appointment and being told I could not come into the doctor's office was unsettling to say the least. A "video consultation" was all that was offered. My reaction was "Wait, you won't be getting any vital signs or other medical information that way!" This led to the natural conclusion that remote detection would be helpful in these conditions. Telemedicine has long been of interest, especially for treating patients in rural or isolated areas where physicians and hospitals may be in short supply or absent entirely. Wearable sensors have become popular items for people who want to track their daily exercise or challenge themselves to exceed past performance with new records.


Many companies make devices that can sense steps taken, pulse rate, heart beats, blood-oxygen levels, rate of motion, temperature, blood glucose levels, and weight among other metrics. Some devices are already in regular use to record continuous health conditions such as wearable heart monitors. Until now, these have made local recordings for later analysis. In the future, one can easily foresee real-time monitoring and diagnosis through the Internet. Many mobile phones support applications that gather, analyze, and present this information. There seems little doubt that many more devices will be developed for non-invasive measurement. It is entirely feasible for more invasive

devices such as pacemakers, defibrillators, and arrhythmia detectors to be linked to watches or mobile phones. I will call these, generally, the Internet of Medical Things. Adding to these, videoconferencing and high-resolution cameras on mobile phones, one can begin to imagine a significant capacity for remote medical diagnosis and triage. The possibilities get longer as more sophisticated measurements become possible taking urine, stool, and blood samples (finger pricks) for local analysis. One can find research papers on artificial olfactory systems and while this work is still in its infancy, it seems reasonable to anticipate successful manufacture of such systems in the not too distant future and which could contribute to the efficacy of telemedicine.

The utility of remote sensing has already become apparent with the COVID-19 epidemic and one might imagine that such practices may become the norm rather than the exception. Such a practice might increase the capacity to perform diagnosis, with tools such as machine learning to provide continuous monitoring and useful alerts. It would be like having a real-time, long-term and continuous doctor's appointment. Increased dependence on such tools opens the question of accurate detection and diagnosis of adverse conditions. False positive and negative detection rates would need to be minimized. Legal controversies are surely predictable, especially in the litigious U.S. Despite these risks,

however, the prospect has very positive potential.

Widespread practice of continuous monitoring could also help with the general assessment of population health, allowing for early detection of epidemic outbreaks and assistance in tracking the spread of communicable illnesses such as the SARS-COV-2 virus. Of course, this also raises the challenge of keeping personal medical information private. Cryptography and strong access control may well contribute to solutions but also pose challenges for cryptographic key management. Every access control mechanism has the potential to be a point for denial of service. Managing keys and access to them will be a predictable challenge if the Internet of Medical Things is to become a common part of public health practice.

It seems inescapable that the Internet of Medical Things will be greatly desired to aid the conduct of safe medical intervention. It will certainly drive the demand for Internet addressing, lending another argument to the importance of adding the IPv6 addressing capability throughout the global Internet. Even if a vaccine for this current pandemic is developed, there will be other pandemics and other communicable disease situations that will benefit from remote diagnosis and triage. I think this notion is here to stay. 

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

Copyright held by author.

The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.



Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/3403958

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

How WWII Was Won, and Why CS Students Feel Unappreciated

John Arquilla considers how code-breaking helped end a war, while Jeremy Roschelle ponders the use of music in data science education.



John Arquilla
Hacking the Axis

<https://bit.ly/3eCnBS6>
May 7, 2020

Observations of the 75th anniversary of the end of World War II in Europe (May 8, 1945) included remembrances of such searing events as the struggle on Omaha Beach on D-Day, the Battle of the Bulge, and at least some recognition of the enormous contribution made by the Russian people to the defeat of Fascism. Yet in all this, I suspect the role of the first “high-performance computing” capabilities of the Allies—known as Ultra in Britain, Magic in the U.S.—will receive too little attention.

The truth of the matter is that the ability to hack into Axis communications made possible many Allied successes in the field, at sea, and in the air.

Alan Turing and other “boffins” at Britain’s Bletchley Park facility built the machine—a much-improved version of a prototype developed by the Poles in the interwar period—that had sufficient computing power to break the German Enigma encoding system developed by

Arthur Scherbius. The Enigma machine was a typewriter-like device with three rotors, each with an alphabet of its own, so each keystroke could create 17,576 possible meanings (26 x 26 x 26). When a fourth rotor was added, the possibilities rose to 456,976 per keystroke.

The Germans had faith in their system, but Turing & Co. met and mastered this challenge. The timely information they decrypted had profound effects at many critical moments. When Erwin Rommel and his Afrika Korps made their final lunge toward the Nile, Ultra intercepts kept the British informed of his exact plan of attack—for which they prepared well, then repulsed. In the Battle of the Atlantic, Ultra hacks not only allowed for the rerouting of convoys away from these predators, but also enabled subhunters to turn up and attack U-boats and their supply ships at even the most remote ocean locations.

Much Ultra-hacked information was shared with the Russians, too—to some extent under cover of a “legend” that the secret material was being provided by a British-run human spy ring. This proved crucial in many Eastern Front actions,

but most notably in the massive tank battle at Kursk in July 1943, which truly broke the back of Hitler’s panzers. At this point, the Germans became convinced some traitor was leaking their most highly classified information to the Allies, but they never lost faith in Enigma.

Nor did the Japanese ever give up on their Imperial Codes, the Magic hacking of which led to the ambush of Admiral Yamamoto’s massive forces at Midway, and greatly informed the American submarine campaign against Japanese shipping. U.S. Navy “pigboats” sank over 80% of Japan’s merchant ships, and about one-third of the Imperial Navy’s warships, almost always guided by Magic hacks. Indeed, the level of detail was so great that, in all the vast Pacific, an American submarine commander often had such exact information that he knew enemy ships’ names, cargoes, even what the noon position of the ship would be on its course the following day!

Truly, the impact of this first “information war” was profound. Had the Axis powers been less complacent about the robustness of their codes, the outcomes of critical battles and campaigns could well have gone in their favor, rather than against them. The lesson for today from this very cautionary tale is that the cybersecurity of armed forces is absolutely crucial to their *physical* security, and to their prospects for victory.

So, on this 75th anniversary of a war best known and remembered for its range of startling new weapons and the sheer grit of its soldiery in battle, let us take just a moment to recognize the pioneering high-performance

computing capacity of the Allies contributed most significantly to the final margin of victory.



Jeremy Roschelle
Learning
Computational
Thinking to Dominate
the Music Industry
<https://bit.ly/2Ymlz2w>

April 22, 2020

One of my early experiences in computing involved using the Music Logo programming language to “program” something that sounded like Beethoven’s Fifth Symphony. Working with MIT professor Jeanne Bamberger (<http://web.mit.edu/jbamb/www/>), I used music as a context for programming and for inquiry into music. This process reshaped my views of what computing could be. I came to see computer science as providing a frame of analysis that could reveal the internal structure and patterns within a wide variety of human experiences, not just those that intrinsically involve computers. This experience led to my career as a learning scientist.

More recently, I have had the opportunity to use music as a context to get middle school students in New York City excited about data science. I served in a consulting role for a multi-institutional team that developed Beats Empire, a game in which a student manages an artist’s rise to fame and fortune. To help their artists, the students must demonstrate what they are learning about using data to analyze music industry trends. The game both engages students and can give teachers a sense of what students know and can do. It is available for free at <https://info.beatsempire.org/>.

Here I will share thoughts about music as a context for learning about computing.

1. Music as an authentic, accessible context

Students have complex experiences with music. They don’t just listen to music; they talk about how artists use social media. They discuss streaming services and how they recommend music. They think about themes in song titles and lyrics. They think about who listens to music where and on what devices. This rich encounter of music on computing platforms can set the stage for a learning opportunity where students go behind the scenes to see how computing influences our experience of music.

In interviewing music industry experts, we found experts could easily and cogently explain to students how data is being used to shape artist’s music and careers—and why data science in the music industry can be a great career for women and people of color. For example, see the interview “A Visit to Chartmetric” (<https://bit.ly/2BjocTs>), where I visited a company that specializes in creating analytics dashboards for artists and their managers. Chartmetric was willing to explain what they do and why they love their jobs to middle school students.

2. Students’ experience of a drive for data

As a learning researcher, I have been involved in many projects that try to involve students with realistic data. Unfortunately, as educators we often come up with “authentic” contexts that aren’t really something students ordinarily would do. In math, I know I have created a “manage a soccer team” unit where students looked at data about how fast team members can run a dash.

One thing I have learned is that in a game context, one can simulate a role where data collection is not “assigned” to students, but where they start from a purpose they care about: helping an artist grow their career. In Beats Empire, students sometimes make spontaneous decisions for their artists. For example, they can recommend a mood or theme for an artist’s song based on their intuition. But they can also collect data in the game, for example on trends in moods and themes. They look into what is popular in particular neighborhoods via an in-game map. The game is set up so paying attention to data and trends can dramatically increase the success of the player’s artist. This creates a relationship to data that is much more like the real world; data is not as a context for a specific math concept or science principle, but rather as a tool for getting better at what you care about—in this case, music. This can be exciting to students.

3. Students’ opportunity to iterate with data

It’s also very common in a math or science class to cycle through using data only once. In a science lab, you collect the data, analyze it, report it, and you are done. In this context, the coherence between the processes of collecting, storing, analyzing, and interpreting is often only in the eyes of the curricu-

lum designer or teacher, who make sure the phases of the cycle fit together. But students need to learn this, too.

It is important for students to see how the separate processes of collecting, storing, analyzing, and interpreting data constrain each other. If you want to analyze a trend of a particular kind, it’s important to collect and store the data in an appropriate way. A gaming context can create a situation where iteratively looping through processes happens quickly and is essential to the game play. Students can start to connect their choice of analysis types to actions they can take in the game. Likewise, they could decide how to collect data based on the questions they want to answer. There are important lessons to be learned in how to iteratively refine the relationship between data and an overall purpose or initiative.

Students are always learning. Meet them where they are.

Overall, the field of the Learning Sciences recognizes learning is not a special type of experience that only happens in designated settings. People are always learning. Too often, we begin by thinking about what students are *not* learning and then we try to create an artificial experience so they will learn. But it’s also possible to take a context in which many students are enjoying learning about every day, like music, sports, food, or fashion, and think about how to deepen their learning about that experience. One path is by layering computing-rich experiences into the contexts students are already motivated to learn about. Games can provide a bridge between what students like to learn and enhanced opportunities that will lead towards a career in computing.

This material is based on work supported by the National Science Foundation under Grants No. 1742011 and 1741956. Opinions, findings, conclusions, or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

John Arquilla is Distinguished Professor of Defense Analysis at the United States Naval Postgraduate School and author, most recently, of *Why the Axis Lost*; the views expressed are his alone. **Jeremy Roschelle** is Executive Director of Learning Sciences Research at Digital Promise and a Fellow of the International Society of the Learning Sciences.

© 2020 ACM 0001-0782/20/8 \$15.00

ACM Gordon Bell Special Prize for HPC-Based COVID-19 Research

Call for Nominations

The Gordon Bell Special Prize for High Performance Computing-Based COVID-19 Research will be awarded in 2020 and 2021 to recognize outstanding research achievement towards the understanding of the COVID-19 pandemic through the use of high performance computing.

The purpose of the award is to recognize the innovative parallel computing contributions towards the solution of the global crisis. Nominations will be selected based on performance and innovation in their computational methods, in addition to their contributions towards understanding the nature, spread and/or treatment of the disease.

Teams may apply for the award. Nominations will be evaluated on the basis of the following considerations:

- Evidence of important algorithmic and/or implementation innovations
- Clear improvement over the previous state of the art
- Performance is not dependent on an architecture that is specialized or cannot be replicated
- Detailed performance measurements demonstrate the submission's claims in terms of scalability (strong as well as weak scaling), time to solution, and efficiency in using bottleneck resources (such as memory size or bandwidth, communications bandwidth, I/O), as well as peak performance.
- Achievement is generalizable, in the sense that other scientists can learn and benefit from the innovations
- Although solving an important scientific or engineering challenge is important to demonstrate/justify the work, scientific outcomes alone are not sufficient for this prize.

Financial support of this \$10,000 award is provided by Gordon Bell, a pioneer in high performance and parallel computing.

Nominations for the 2020 award are due on October 8, 2020.

**For more information and to
submit nominations, please visit:**

<https://awards.acm.org/bell/covid-19-nominations>

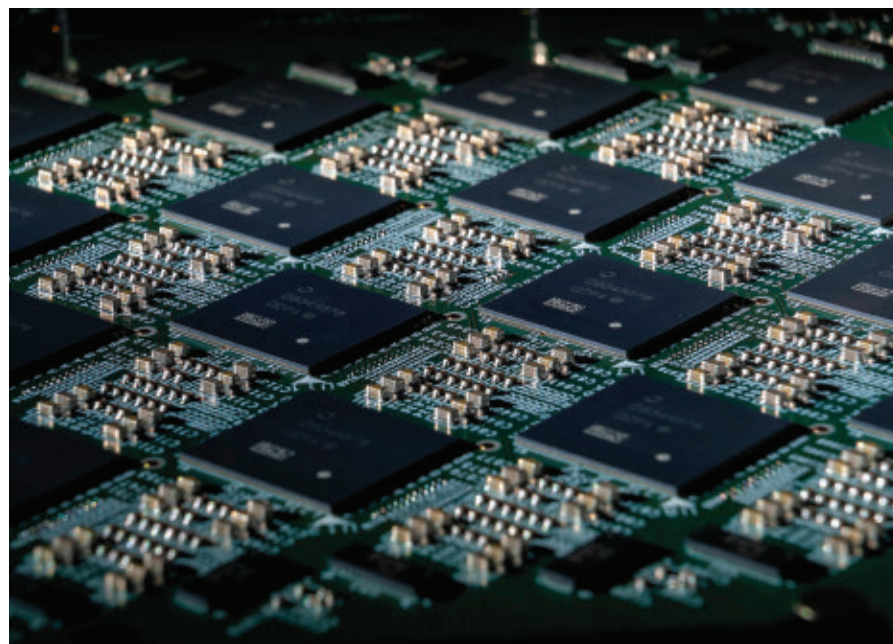


Neuromorphic Chips Take Shape

Chips designed specifically to model the neurons and synapses in the human brain are poised to change computing in profound ways.

THE ABILITY OF the human brain to process massive amounts of information while consuming minimal energy has long fascinated scientists. When there is a need, the brain dials up computation, but then it rapidly reverts to a baseline state. Within the realm of silicon-based computing, such efficiencies have never been possible. Processing large volumes of data requires massive amounts of electrical energy. Moreover, when artificial intelligence (AI) and its cousins deep learning and machine learning enter the picture, the problem grows exponentially worse.

Emerging neuromorphic chip designs may change all of this. The concept of a brain-like computing architecture, conceived in the late 1980s by California Institute of Technology professor Carver Mead, is suddenly taking shape. Neuromorphic frameworks incorporate radically different chip designs and algorithms to mimic the way the human brain works—while consuming only a fraction of the energy of today’s microprocessors. The computing model takes direct aim at the inefficiencies of existing computing frameworks—namely the von Neumann bottleneck—which forces a pro-



Intel combines 64 of its Loihi “brain-on-a-chip” neuromorphic chips to form a “Pohoiki Beach” neuromorphic system featuring eight million artificial neurons.

cessor to remain idle while it waits for data to move to and from memory and other components. This causes slowdowns and limits more advanced uses.

“Neuromorphic chips introduce a level of parallelism that doesn’t exist in today’s hardware, including GPUs and most AI accelerators,” says Chris Elia-

smith, a professor in the departments of Systems Design Engineering, and Philosophy, of the University of Waterloo in Ontario, Canada. Although today’s deep learning systems rely on software to run basic neuromorphic systems using conventional field-programmable gate arrays (FPGA), central processing units

(CPUs), and graphics processing units (GPUs), chips specifically designed to accomplish these tasks could revolutionize computing. Neuromorphic chips are packed with artificial neurons and artificial synapses that mimic the activity spikes that occur within the human brain—and they handle all this processing on the chip. This results in smarter, far more energy-efficient computing systems.

The impact of commercial neuromorphic computing could be enormous. The technology has repercussions across a wide swath of fields, including image and speech recognition, robotics and autonomous vehicles, sensors running in the Internet of Things (IoT), medical devices, and even artificial body parts.

As Adam Stieg, associate director of the California NanoSystems Institute at the University of California at Los Angeles (UCLA) puts it: “The ability to perform computation and learning on the device itself, combined with ultra-low energy consumption, could dramatically change the landscape of modern computing technology.”

Modeling the Brain

The human brain is a remarkable product of evolution. It has a baseline energy footprint of about 20 watts, while processing complex tasks in milliseconds. While today’s CPUs and GPUs can dramatically outperform the human brain for serial processing tasks, the process of moving data from memory to a processor and back not only creates latency, it expends enormous amounts of energy. A typical desktop computer burns through approximately 200 watts, while some supercomputers pull as much as 20 megawatts.

The value of neuromorphic systems is they perform on-chip processing asynchronously. Just as the human brain uses the specific neurons and synapses it needs to perform any given task at maximum efficiency, these chips use event-driven processing models to address complex computing problems. The resulting spiking neural network—so called because it encodes data in a temporal domain known as a “spike train”—differs from deep learning networks on GPUs. Existing deep learning methods rely on a more basic brain model for handling tasks, and they

must be trained in a different way than neuromorphic chips.

“If we look at biology, we see incredible energy efficiency. This is something we’re hoping to emulate in artificial systems,” says Garrick Orchard, a researcher in Intel’s Neuromorphic Computing Lab. The artificial neurons and synapses in neuromorphic chips can be stacked into layers and inserted in multiple cores. “The idea is that by taking inspiration from biology and by trying to better understand what principles are crucial for low-power computation, we can mimic these characteristics in silicon and push the boundaries of what’s possible.”

However, it isn’t just slashing energy consumption that’s appealing. Today’s CPUs and GPUs—especially when they are used in autonomous vehicles and other independent systems—typically rely on external systems, primarily clouds, to handle some of the processing. The resulting latency is a problem for on-board systems that must make split-second decisions. “You can’t collect a frame, pass it through to a deep neural net, and wait for the response when you’re traveling down a freeway at 70 miles an hour,” explains Abu Sebastian, Principal Research Staff Member at IBM Zurich. “Everything has to happen instantaneously, and that requires fast on-board processing.”

So, while the need for clouds and edge networks won’t disappear with neuromorphic chips, autonomous systems will be able to handle additional critical computing tasks on board. In areas such as image processing, this could produce exponential improvements. The latency gain of a spike-based neural network is a fundamental

Neuromorphic systems perform on-chip processing asynchronously, using event-driven processing models to address complex computing problems.

benefit—and it evolves beyond today’s GPU systems. “Due to the asynchronous data-driven mode of computing, the salient information propagates in a fast manner through multiple layers of the network. The spikes begin to propagate immediately to higher layers once the lower layer provides sufficient activity. This is very different from conventional deep learning, where all layers have to be fully evaluated before the final output is obtained,” Sebastian says.

Neuromorphic chips also have the ability to learn continuously. “Because of their synaptic plasticity and the way they learn, they can continue to adapt and evolve,” says Sebastian. In practical terms, for example, a robotic arm could learn to recognize different objects and pick them up and move them in a nuanced way. If a heavier grip is needed, the system would adjust accordingly, and if a lighter touch is required, it would also adapt. New items wouldn’t throw a neuromorphic system off-kilter; it would simply “evolve” and “at a much faster rate than a CPU could,” Orchard says.

By combining improved energy efficiency, reduced latency, and improved on-board learning, neuromorphic chips could push image recognition and speech processing to new levels of speed, efficiency, and accuracy. The technology could seed speech processing on virtually every type of device and produce new types of video cameras that operate at lower power and detect patterns and events more efficiently, Eliasmith says. Still another possible gain could take place in datacenters, which consume vast amounts of power and produce enormous carbon footprints.

The sum of these gains could produce revolutionary breakthroughs. Researchers have begun to explore the possibility of developing prosthetics that would give amputees the sensation of touch, brain-implanted chips that could aid stroke or Alzheimer’s victims, self-healing electronic skin, and even vision sensors—essentially retinal implants—that could restore vision to the blind. Scientists also are exploring probabilistic neuromorphic systems that could predict the odds of an earthquake or recession with a high level of accuracy.

Says Eliasmith, “Neuromorphic designs allow scaling that hasn’t been possible the past. We’re able to go far beyond what today’s systems can do.”

Getting Smarter

Neuromorphic chips won't replace today's CPUs and GPUs; they are more likely to be embedded next to them as separate cores. This would expand the way we use existing digital technology—particularly on the edge of the network—and provide an accelerator for niche tasks. “Today's computers are very good at what they do. They will continue to outperform neuromorphic computing systems for conventional processing tasks. The technologies are complementary and so they will co-exist,” says G. Dan Hutcheson, CEO of VLSI Research, an independent market analysis and consulting firm that tracks the semiconductor industry.

Research and development efforts are beginning to produce tangible results. For instance, Intel Labs has developed Loihi, a research chip that uses a spiking neural network architecture. The processor contains 128 neuromorphic cores, three Lakemont (Intel Quark) CPU cores, and an off-chip communications network. The chip is designed with a high level of configurability, along with cores that can be optimized for specific tasks. This makes it appealing for specialized devices. More than 80 members of Intel's Neuromorphic Research Community—including universities, government labs, neuromorphic startup companies, and Fortune 500 firms, are now experimenting with Loihi.

IBM has developed a neuromorphic chip named TrueNorth. It has 4,096 cores, each with 256 neurons that, in turn, contain 256 synapses each. The microprocessor has 1/10,000th of the power density of a conventional von Neumann processor. It achieves this efficiency with a spiking neural network. Activity in the synthetic neurons occurs only when and where it is needed. This makes the chip particularly suited to high-speed and low-energy image processing and classification tasks. Although TrueNorth is an experimental chip, IBM is continuing to actively research neuromorphic technology, including approaches that focus on learning in the chip, Sebastian says.

More than 50 other AI startups around the world are actively developing neuromorphic chips and technology for a wide array of purposes, Hutcheson says. While all of this is taking place, others are developing software and systems to

Neuromorphic technology remains in its infancy; there are no commercial products or killer applications. Yet the field is advancing rapidly and radically.

optimize chip performance. For instance, Eliasmith, who also heads a startup company called Applied Brain Research, develops algorithms and software used to program neuromorphic chips. This includes algorithms used for deep spiking networks, spiking and non-spiking adaptive controls, recurrent neural networks, on-chip learning, and spiking and non-spiking hierarchical reinforcement learning.

Meanwhile, in research labs, scientists are experimenting further with the technology. For example, at UCLA, Stieg and chemistry professor James Gimzewski have developed neuromorphic systems that can recognize rewards—similar to a rat in a maze—and artificial synapses that can “forget” by using varying input waves. Borrowing methods from human psychology, “We're building circuits that can adapt more efficiently by forgetting what isn't important,” Gimzewski explains. The pair also have developed nano-wire technology that mimics millions of connections in the brain. “This introduces a level of impermanence that allows the devices to be far more flexible,” says Stieg.

A New Model

For now, neuromorphic technology remains in its infancy. There are no commercial products, there are no killer applications. Yet, the field is advancing rapidly and radically. Commercially available chips should begin appearing within the next year or two, and the technology will likely take off in earnest within the next three to five years. Neuromorphic chips are likely to have a significant impact on edge devices and IoT systems that “must integrate dynamic

and changing information that doesn't necessary run on a single algorithm—all while conserving energy,” Stieg says.

“The world is not linear. It's not deterministic. It doesn't give definitive answers,” he concludes. “Conventional von Neumann-based computing systems deal mostly with high-speed, predictable, deterministic processes. They perform these tasks well, but struggle when things become more complex. Neuromorphic computing aims to open up an entirely new and unexplored area of computing. It could allow us to do things with computers that we couldn't have imagined in the past.”

Further Reading

Demis, E.C., Aguilera, R., Sillin, H.O., Scharnhorst, K., Sandouk, E.J., Aono, M., Stieg, A.Z., and Gimzewski, J. K. **Atomic Switch Networks — Nanoarchitectonic Design of a Complex System for Natural Computing.** *Nanotechnology*, Volume 26, Number 20. April 27, 2015. <https://iopscience.iop.org/article/10.1088/0957-4484/26/20/204003/meta>

Davies, M., Srinivasa, N., Lin, T. Chinya, G., Cao, Y., Choday, S., Dimou, G., Joshi, P., Imam, N., Jain, S., Liao, Y. Lin, C., Lines, A., Liu, R., Mathaikutty, D., McCoy, S., Paul, A., Tse, J., Venkataramanan, G., Weng, Y., Wild, A., Yang, Y., and Wang, H.

Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* Volume: 38, Issue: 1, January/February 2018, pp. 82-99. <https://ieeexplore.ieee.org/abstract/document/8259423>

DeBole, M.V., Taba, B., Amir, A., Akopyan, F., Andreopoulos, A., Risk, W., Kusnitz, J., Otero, C.O., Nayak, T.K., Appuswamy, R., Carlson, P.J., Cassidy, A.S., Datta, P., Esser, S.K., Garreau, G.J., Holland, K.L., Lekuch, S., Mastro, M., McKinsty, J., di Nolfo, C., Paulovicks, B., Sawada, J., Schleupen, K., Shaw, B.G., Klamo, J.L., Flickner, M.D., Arthur, J.V., and Modha, D.S.

TrueNorth: Accelerating From Zero to 64 Million Neurons in 10 Years. *Computer*, Volume: 52, Issue: 5, May 2019, pp. 20-29. <https://ieeexplore.ieee.org/abstract/document/8713821>

Blouw, P., Choo, X., Hunsberger, E., and Eliasmith, C.

Benchmarking Keyword Spotting Efficiency on Neuromorphic Hardware. *NICE '19: Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop* March 2019 Article No.: 1 pp. 1-8. <https://doi.org/10.1145/3320288.3320304>

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

Digital Humans on the Big Screen

Motion pictures are using new techniques in computer-generated imagery to create feature-length performances by convincingly “de-aged” actors.

ARTIFICIAL IMAGES HAVE been around almost as long as movies. As computing power has grown and digital photography has become commonplace, special effects have increasingly been created digitally, and have become much more realistic as a result.

ACM’s Turing Award for 2019 to Patrick M. Hanrahan and Edwin E. Catmull reflected in part their contributions to computer-generated imagery (CGI), notably at the pioneering animation company Pixar.

CGI is best known in science fiction or other fantastic settings, where audiences presumably already have suspended their disbelief. Similarly, exotic creatures can be compelling when they display even primitive human facial expressions. Increasingly, however, CGI is used to save money on time, extras, or sets in even mundane scenes for dramatic movies.

To represent principal characters, however, filmmakers must contend with our fine-tuned sensitivity to facial expressions. Falling short can leave viewers in the “uncanny valley,” distracted or even repulsed by a zombie-like representation. “Trying to do realistic humans is still the most difficult aspect of visual effects,” said Craig Barron, creative director at visual development and experience company Magnopus. Barron shared the 2008 Academy Award for Best Visual Effects for *The Curious Case of Benjamin Button*, in which the title character ages backward from an old man to an infant.

In the last decade, many films have included short flashbacks with younger versions of their characters. Within the last year, however, some films have used new techniques to create feature-length performances by convincingly “de-aged”



In Ang Lee’s 2019 action movie *Gemini Man*, Will Smith (right) is confronted by a younger clone of himself (left). This filmmaker went through a complicated process to “de-age” Smith for the younger role.

actors. Artificial intelligence also increasingly will augment the labor-intensive effects-generation process, allowing filmmakers to tell new types of stories.

Synthetic Performers

“The idea of de-aging has been around for a while, and companies like Lola Visual Effects have been doing an amazing job of using 2D tools to basically track young patches onto old faces to make people look convincingly a different age,” said Guy Williams of Weta Digital in Wellington, New Zealand. Williams was video-effects supervisor for Ang Lee’s 2019 action movie *Gemini Man*, in which Will Smith played an assassin confronted by a younger clone of himself. In that film, filmmakers took the process a step further, he said. “Instead of modifying an image to make a performance look younger, we erase the image, create a synthetic young performance, and place it into the shot.”

“In many ways it felt less about a visual effect and more like we were creating a reality and aiming to work out exactly how the human face works, operates, but also how it ages over time,” added Stuart Adcock, facial animation supervisor at Weta. Because of this wholesale replacement, “we were not limited to any photography,” Adcock said. “We had full freedom, so when we did need to take it somewhere else, then we were able to do that.”

To build its model, the team first ran Smith through various facial exercises, such as raising his upper lip. Such “action units,” which have overlapping, interacting effects in various regions of the face, constitute the elements of the widely used facial action coding system (FACS), originally based for detailed understanding of the skull and its muscles. After a year or so of tuning a “puppet” to reproduce “current-age Will,” they turned their attention to his younger clone.

Adcock likens this process to creating a musical score derived from a performance on one instrument. This representation allows the same musical piece to be performed on another instrument, which imparts its own characteristic sound.

To construct the “score,” the team built on its previous expertise animating fantastic characters like Gollum in *The Hobbit*, equipping Smith with facial markers and a head-mounted camera rig that recorded his expressions. These rigs have been getting sleeker and lighter all the time, Adcock said. Such “performance capture,” as contrasted with “motion capture” of bodily movements, provides extremely high-quality facial data for the animators.

Supporting the Storytellers

For renowned director Martin Scorsese, however, headgear and facial markers (dots on the actors faces) were still unacceptably disruptive to the human interactions he sought in the 2019 epic *The Irishman*. The film features Robert De Niro, Al Pacino, and Joe Pesci, all now in their late 70s, in roles spanning many ages across several decades, so a strategy for de-aging was critical. This resulted in the film being nominated for Best Special Effects and nine other Academy Awards.

Beginning several years earlier, Pablo Helman and his team from Industrial Light and Magic (ILM), the special effects house founded by George Lucas, worked with Scorsese to develop ways to create the desired images much less intrusively. To capture the performances, they built a bulky apparatus in which a normal camera was flanked by two infrared cameras and associated infrared lighting that provided synchronized, high-quality stereoscopic information needing neither visible facial markers nor changes in the lighting desired by the filmmakers.

The size and weight of this equipment posed some challenges; for example, precluding the popular hand-held Steadicam shots. Moreover, although the infrared light did not interfere with the main camera, the infrared cameras were sensitive to cigarette smoke, and could not see through vintage car windshields.

In parallel with the hardware, the ILM team developed a new software

Instead of using animators to build a puppet with DeNiro’s facial motions, the FLUX software morphed captured performance and lighting data to create the new image.

pipeline to construct the de-aged images. In addition to FACS capture from the current actors, the team acquired huge numbers of images from their previous performances over the decades, allowing them to select a facial model for each of the years in the film. Rather than reproduce the exact look of De Niro from *Taxi Driver* (1976) or *Goodfellas* (1990), however, they strove to create a de-aged version of his *Irishman* character. Moreover, instead of using animators to build a puppet with De Niro’s facial motions, their FLUX software morphed the captured performance and lighting data to create the new image.

The result was a compelling de-aged character that let the actors act—and interact.

“If you’re going to get the top actors of our time, you need to support their acting process and not create a technology or impose a technology that would somehow diminish that performance. That’s what they were able to achieve,” said Barron, who said the less-intrusive technology “allows more adoption among a wider group of people into different kind of films.”

Under the Hood

Unlike the latest superhero movie, a film like *The Irishman* succeeds when the audience forgets about its technical sophistication. Nonetheless, achieving this realism requires an enormous, diverse team innovating in both hardware and software (and a reported \$160-million budget).

“There are people that are creative, they’re thinking about design and lighting and composition and performance

ACM Member News

HUMAN-COMPUTER PARTNERSHIPS



Wendy Mackay is a research director at Inria Paris-Saclay, the French national research institute for the

digital sciences, at the Université Paris-Saclay in Paris, France. She heads up the ExSitu research group, a lab of 30 people that explores the limits of human-computer interaction.

“I have always been interested in how people interact with technology,” Mackay says, and human-computer partnerships have become a focus for her. People react to technology, she points out, but they also appropriate it and do innovative things with it; as a result, technology should be designed to support user-innovation and enhance users’ capabilities, she says.

Mackay received a bachelor’s degree in experimental psychology from the University of California, San Diego; an M.S. in experimental psychology from Northeastern University, and a Ph.D. in Management in Technological Innovation from the Massachusetts Institute of Technology.

She started her professional career at Digital Equipment Corporation (DEC), worked there in different capacities over a period of 11 years. She was working on something new at the time: human-computer interaction. “It was a new field, combining psychology and how we think about people and technology, and putting them together, with an emphasis on innovation,” she explains.

After DEC, Mackay joined academia, teaching at Denmark’s University of Aarhus, France’s University of Paris-Sud, and even Stanford University as a visiting professor.

In the future, Mackay plans to explore the physics of interactions with digital material. “We will use this to rethink how we use intelligent systems—machine learning and other types of AI—to enhance users’ capabilities, to move people forward and do more.”

—John Delaney

of storytelling, and there's the people under the hood that have to make that all work. It's really a collaboration between the technologist and the artist," said Barron, whose current work at Magnopus aims to extend storytelling to the realm of virtual and augmented reality. "It's the collaboration that determines whether the project is successful or not."

For example, the Weta team worked together to improve the "facial solver" that represents the captured positions of face markers and other features in terms of underlying action units. Although there may be different ways to break down the various training movements, they used deep learning to ensure their description matched the way their animators think about those movements. With that training, the solver could then decide, for each frame of the footage, which muscles were firing, Stuart said. "It gave us a good starting point."

"The power of AI being applied to visual effects is relatively new and there are huge potentials for that," Barron said. "To harness the power of the computer to teach it to simulate reality, whether it's through creating a performance or a synthetic human or an environment, I think will have a lot of benefit to creating more and more credible illusions."

Bringing Back the Dead

Creating lifelike representations of actors in roles they never played does raise challenging ethical issues. One is

"The power of AI being applied to visual effects is relatively new and there are huge potentials for that," Barron said.

the ability to put words in the mouths of politicians, or to put celebrities into pornographic scenes. Fortunately, such "deepfakes" are unlikely to have the Hollywood-level resources and actor cooperation used to make truly convincing fakes, although they are already good enough to cause trouble. (Some critics noted that *The Irishman* repeated some implausible claims from the book on which it was based, but worries about movies distorting historical facts are nothing new.)

Another concern is reuse of actors who are unavailable, or even dead. The 2016 "Star Wars story" *Rogue One*, for example, included a brief but controversial appearance by Peter Cushing, who had died in 1994. The most ambitious re-animation so far is the reported casting of James Dean, who died in 1955, as a costar in the film *Finding Jack*,

scheduled for release late this year.

The filmmakers obtained legal permissions to use the actors' likenesses in these films. Still, some commenters are worried about the effect of recycling actors from previous eras will make it harder for current actors to find roles, except as a blank slate onto which more famous faces are mapped.

Ironically, one of De Niro's breakout roles was as a young Vito Corleone in *The Godfather Part II*. With today's technology, he might have been demoted to a body double for a de-aged Marlon Brando. **C**

Further Reading

Seymour, M., *De-Aging the Irishman, fxguide*, December 2019, <https://bit.ly/2XZV1DE>

Tonelli, B., *The Lies of The Irishman, Slate*, August 2019, <https://bit.ly/3cATzwF>

Pioneers of Modern Computer Graphics Recognized with ACM A.M. Turing Award, ACM, March 18, 2020, <https://awards.acm.org/about/2019-turing>

The Hobbit: An Unexpected Journey VFX | Breakdown - Gollum, Weta Digital, <https://bit.ly/2UbKgg>

Dr. Ekman explains FACS (Facial Action Coding System), <https://bit.ly/3gRRakx>

How *The Irishman's* Groundbreaking VFX Took Anti-Aging To the Next Level, Netflix, <https://bit.ly/2AJFU99>

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

© 2020 ACM 0001-0782/20/8 \$15.00

Milestones

Balcan to Receive ACM Grace Murray Hopper Award

ACM named Maria Florina "Nina" Balcan of Carnegie Mellon University to receive the ACM Grace Murray Hopper Award for her contributions to minimally supervised learning.

Balcan's pioneering work in machine learning solved longstanding open problems, enabled entire lines of research crucial for modern AI systems, and set the agenda of the field for years to come.

ACM President Cherri M. Pancake said although Balcan is still in the early stages of her career, "she has already established herself as the world

leader in the theory of how AI systems can learn with limited supervision. More broadly, her work has realigned the foundations of machine learning, and consequently ushered in many new applications that have brought about leapfrog advances in this exciting area of artificial intelligence."

Balcan introduced the first general theoretical framework for semi-supervised learning, showing how to achieve provable guarantees on the performance of such techniques with concrete implications for many different types of semi-supervised

learning methods.

She also made significant contributions to active learning by establishing performance guarantees for active learning that hold even when "noise" is present in the data, and with colleagues she developed algorithms that can learn more efficiently under more specialized forms of "label noise."

Balcan proposed a theoretical foundation for understanding the general kinds of structures that can be detected by clustering, as well as characterizing the functionality of specific

clustering algorithms. She also devised novel clustering algorithms derived from these theoretical foundations, and showed applications of these algorithms in computational biology and Web search.

The ACM Grace Murray Hopper Award is given to the outstanding young (under 35) computer professional of the year, selected on the basis of a single recent major technical or service contribution. The award is accompanied by a prize of \$35,000 (financial support for this award is provided by Microsoft).

Are We Addicted to Technology?

Experts agree technology causes some negative behaviors, but they are divided on how bad the problem is.

IT'S EASY to think the world is suffering from full-blown technology addiction.

We read daily headlines about how social media platforms threaten our mental health, our relationships, and even democratic society itself. We hear smartphone addiction is the latest scourge sweeping the nation's youth, and we even see tech leaders like Chris Hughes, who co-founded Facebook, publicly call for the break-up of the firm he created because of its addictive content and features.

It certainly seems like "technology addiction" is a real condition and that it is everywhere. But the truth is a little less black and white.

Technology addiction is a broad term that isn't always well defined. It can mean any type of negative behavior across video gaming, smartphone usage, and use of social media platforms like Facebook. It is medically unclear if these negative behaviors are actually addictive, and it is difficult to tell if these behaviors are due to the way the technology in question works or because we have a hard time controlling our own use of individual technologies.

Video game addiction was added by the World Health Organization (WHO) in 2018 to its International Classification of Diseases, which the organization describes as the international standard for disease reporting. The move was welcomed by some who see video game addiction as a real disease, but it was contested by others who argued that video game addictions—and other types of technology addiction—do not meet clinical standards of addiction.

While everybody seems to agree video gaming in excess can cause harm, there is less consensus on



whether or not smartphones and consumer technology have negative effects on our behavior and, if so, how to classify these effects.

Bad Habit or Actual Addiction?

WHO says video game addiction occurs when gaming interferes with life, and the individual is unable to stop gaming despite this interference. It also says this severity of behavior must occur for a year or more to classify as an addiction.

Clearly, some people experience real physical and mental harm from overusing video games.

"For gamers who struggle with video game addiction, it's a real condition that impacts many areas of life, including school, employment, mental and physical health, and relationships," says Cam Adair, founder of Game Quitters, a video game addiction support group. Adair describes himself as a video game addict who was hooked for 10 years, playing up to 16 hours a day, until the habit caused problems in his life, including forcing him to drop out of school. Today, he speaks and writes about his recovery, and helps other video game addicts kick the habit. He sees validation for video game addiction as a harmful condition worth treating in the 75,000 people in 95

countries looking for help on Game Quitters every month.

Adair sees clear negative effects from excessive video gaming every day in the people he helps. Extreme video game addicts, he says, may neglect to eat, sleep, or to perform work or school duties. "The most common case I see is a college student, usually male, who is now beginning to fail school and can't seem to get themselves away from games," says Adair.

In Adair's view, video games themselves cause some of these problems. Some sufferers find the perception of achievement within games so addictive that they stop pursuing goals in the real world. In other scenarios, he says, the technology may be used as a distraction from actual depression or anxiety. In either case, the effects of excessive video gaming on lives are very real.

"Struggling gamers are losing their jobs, failing school, and getting divorced. The real-life impact is significant and can be devastating," he says.

Not everyone agrees that the negative effects of video gaming should be clinically labeled as an addiction.

Mental health researchers, led by psychologist Andrew Przybylski, publicly contested WHO's classification. Przybylski, an associate professor, senior research fellow, and director of research at the Oxford Internet Institute of the University of Oxford, says WHO relies too heavily on research into gambling behaviors, which are addictive, and has not reached a consensus on the symptoms of video game overuse.

There is even less agreement about the negative effects around smartphones and other consumer technology platforms.

Digital Tools, Physical Effects

While WHO has formally recognized video game addiction, it has not recognized

addictions related to smartphone use or to other consumer technologies. Neither has the Diagnostic and Statistical Manual of Mental Disorders, the U.S. ‘bible’ of psychological conditions.

However, one 2016 study led by Suliman Aljomaa of King Saud University in Saudi Arabia and published in *Computers in Human Behavior* found that, among undergraduate university students surveyed, the smartphone “addiction percentage among participants was 48%.” However, the degree of addiction differed based on factors like gender and social status.

Another study, published in the journal *BMC Psychiatry*, found problematic smartphone usage was associated with increased anxiety and depression in children and young people, although these symptoms were self-reported.

Other researchers say studies such as these are flawed.

In a 2018 study published in the *Journal of Behavioral Addictions*, Taryana Panova and Xavier Carbonell surveyed a range of literature that claimed smartphones were addictive. The researchers cite addiction’s clinical symptoms as “mood modification, tolerance, salience, withdrawal symptoms, conflict, and relapse.” However, they found much of the literature relied on self-reported results and inconsistent questions to determine if a clinical addiction was present.

Panova and Carbonell concluded problematic smartphone use is “an evolving public health concern that requires greater study to determine the boundary between helpful and harmful technology use.” However, they said, excessive smartphone use doesn’t merit the term “addiction.”

Przybylski, the psychologist who contested WHO’s classification of video games as an addiction, echoes this view. He says we aren’t even thinking about the problems posed by technology properly yet. The potential negative effects of technology deserve serious consideration, conversation, and study, he says, but that is not what is happening now. “I’m certain that the world would be a better place if we deleted most of the existing research [on technology addiction] and started anew with open, transparent, and reproducible science.”

Przybylski would like to see independent scientists investigating technology addiction show their work using robust methods, as well as soliciting participation from video game, technology, and social media companies in their studies. “It’s pretty clear that there may be something going on, but it is not clear that technology is to blame,” he says. “The current literature is a bit like blaming a runny nose for the flu.”

What the Doctors Order?

Despite the debate, excessive video gaming’s negative effects are severe enough to merit prescriptive solutions.

Game Quitters offers educational programs for gamers, parents, and medical professionals. Traditional therapy and residential programs for gaming addicts exist, too, including some run by the U.K. National Health Service. Adair says these are still being developed, and they can be expensive, with private residential treatments running \$10,000 or more per month. “That’s one reason why I believe it’s essential that free or affordable solutions are developed and provided for people struggling,” says Adair. “If someone has a video game addiction, cost should not be a barrier for them to receive help.”

Yet the “solution” to excessive smartphone and technology use, if needed at all, is unclear.

Behavioral scientist Nir Eyal, author of *Hooked: How to Build Habit-Forming Products*, argues often and vocally that technology is not addictive. Eyal also has written about how to use technology responsibly, in his book *Indistractable: How to Control Your Attention and Choose Your Life*. In both books, Eyal argues that technology use or abuse is up to the user. He recommends individuals assess how they spend their time at a granular level, to better understand what distracts them and why.

Others argue we need to address technology’s negative effects by taking back power from the massive technology companies that make smartphones, software, and platforms.

Tristan Harris, a former Google product manager who now runs the nonprofit Center for Humane Technology, which is dedicated to realigning people’s relationships with massive

technology companies, argues firms like Apple, Facebook, and Google engineer products to capture maximum attention, since their business models rely on active users and captive audiences for advertising. Because of this, popular technology tools and platforms direct our behavior in ways we can’t always control.

The Center for Humane Technology pushes for societal change through writing, speaking, and lobbying policy makers for greater oversight over tech companies.

At the end of the day, people like Adair are not sure it even matters if excessive technology usage is classified as an addiction, or who is to blame. With extreme gaming behavior, says Adair, the effects are real no matter what you call it or at whom you point fingers. “While professionals may debate the merits of a video game addiction diagnosis, the gamer themselves is actively struggling.”

Further Reading

Aljomaa, S.,
Smartphone addiction among university students in the light of some variables, *Computers in Human Behavior*, Aug. 2016, <http://bit.ly/2xf0VHh>

Eyal, N.,
Hooked: How to Build Habit-Forming Products, November 2014, <https://amzn.to/2TFBYvZ>

Panova, T. and Carbonell, X.,
Is smartphone addiction really an addiction?, *National Center for Biotechnology Information*, Jun. 12, 2018, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6174603>

Sohn, S., Rees, P., Wildridge, B., Kalk, N., and Carter, B.,
Prevalence of problematic smartphone usage and associated mental health outcomes amongst children and young people: a systematic review, meta-analysis and GRADE of the evidence, *BMC Psychiatry*, November 29, 2019, <http://bit.ly/31SmNDF>

Thompson, N.,
Our Minds Have Been Hijacked by Our Phones. Tristan Harris Wants to Rescue Them, *WIRED*, Jul. 26, 2017, <http://bit.ly/3azwCcJ>

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

SHAPE THE FUTURE OF COMPUTING. JOIN ACM TODAY.

www.acm.org/join/CAPP

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

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

ACM STUDENT MEMBERSHIP:

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

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

PAYMENT INFORMATION

Name _____

Mailing Address _____

City/State/Province _____

ZIP/Postal Code/Country _____

- Please do not release my postal address to third parties

Email Address _____

- Yes, please send me ACM Announcements via email
- No, please do not send me ACM Announcements via email

- AMEX VISA/MasterCard Check/money order

Credit Card # _____

Exp. Date _____

Signature _____

Purposes of ACM

ACM is dedicated to:

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

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

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

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

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.



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

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

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



DOI:10.1145/3408052

Jannie Fernandez and JeffriAnne Wilder

▶ Richard Ladner, Column Editor

Broadening Participation TECHNOLOchicas: A Critical Intersectional Approach Shaping the Color of Our Future

A unique partnership seeks to address the underrepresentation and unique barriers facing Latina women and girls of color in information technology.

ACHIEVING DIVERSITY IN computing is a pressing national concern in the U.S.^{8,14} Computing-related jobs will be among the fastest growing and highest-paying over the next decade according to the U.S. Bureau of Labor Statistics.¹⁶ Yet, women are significantly underrepresented in computing degrees and careers, holding only 26% of U.S. computing occupations.¹⁷ There is even a greater dearth of women of color in computing, especially Latinas. Eighteen percent of the U.S. population is Hispanic and Latinx, yet currently only 1% of the jobs in the computing workforce are occupied by Latinas. Latinx girls represented a mere 4% of all students taking the AP computer science exam in 2017.¹¹ In 2014, Hispanic women received only 2% percent of doctoral degrees in computer and information sciences.

The National Center for Women &

Information Technology (NCWIT; see <https://www.ncwit.org>) was chartered in 2004 to broaden the participation of women and girls across the technology ecosystem (K–12, higher education, and industry). NCWIT's organizational mission is centered around the concept of intersectionality: equi-

TECHNOLOchicas provides various types of resources, in English and Spanish, to help families encourage the young women in their lives to pursue computing.

ty efforts must take into consideration how race, class, gender, and other aspects of identity shape strategies and approaches to broadening participation in computing.⁶ Research supports that such approaches are a key component in ensuring the full access, engagement, and inclusion of women and girls of color in STEM and computing.^{1,2,9,14}

Employing an intersectional framework is critical in understanding how to address the underrepresentation of Latinas in computing. Latina girls and women in the U.S. are avid users of technology, but they are significantly underrepresented in its creation. There are several unique barriers facing Latinas' participation in computer science, including: a lack of access to CS classes in school and afterschool programs;⁴ lower levels of STEM and CS confidence and self-efficacy;¹⁸ a lack of family support and encouragement to pursue computing careers;³



and a lack of Latina role models in computer science.^{5,8,10,15}

In January 2015, NCWIT and Google convened a historic roundtable session in Washington, D.C., to address the challenge of how to increase Latinas' awareness, interest, and involvement in technology. Dozens of stakeholders from various sectors—including government, industry, and non-profit change leaders—participated in the 2015 meeting. Attendees spent the day identifying ways to broaden the participation of Latinas in tech, and developing key strategies to support and encourage young Latinas. Participants agreed on the value of creating a media campaign to redefine the mainstream image of computer science and information technology to make it more inclusive and appealing to young Latinas. In partnership with the Televisa Foundation,^a TECHNOchicas was born (see <https://technochicas.org/>).

Guided by the distinctive voices

^a TECHNOchicas is co-produced by NCWIT and the Televisa Foundation with support from Apple, Qualcomm, Microsoft, AT&T, Univision and the Computing Alliance of Hispanic Serving Institutions (CAHSI).

and perspectives of Latinas, TECHNOchicas was designed as a national initiative attentive to the unique barriers facing Latinas in gaining access and inclusion in computing, and to inspire the next generation of Latina technology innovators. Visibility is a key aspect of the initiative, as we understand that young Latinas “can’t be what they can’t see.” The first TECHNOchicas public service announcement (PSA) debuted on January 18, 2016, bringing positive Latina role models into the homes of millions. The program celebrates and highlights Latinas achieving success in tech who inspire young women to pursue computing. The campaign helps to show the Latinx community that their daughters can achieve the same, or even greater, success in the technology industry.

TECHNOchicas provides various types of resources, in English and Spanish, to help families encourage the young women in their lives to pursue computing. These materials include research-based guides and tips, inspirational videos, and profiles of real-life Latina role models. An exam-

ple is “Top 10 Ways Families Can Encourage Girls’ Interest in Computing,”^b which offers tips for sparking interests and taking advantage of hands-on learning opportunities as just a couple of key ways for families to help further the TECHNOchicas mission of increasing the meaningful participation of Latinas in the tech industry. Connecting a girl’s current interests to computing concepts and discussing how computing is relevant in improving everyday social issues is also critical. Online videos and profiles of more than 300 TECHNOchicas help girls see real-life, diverse Latinas in tech. These stories also help young girls understand how they can connect their own interests into the development of cutting-edge products and solutions for society at large.

Measuring Impact

“In the four years since the launch of TECHNOchicas, important strides have been made in raising awareness

^b See <https://www.ncwit.org/resources/top-10-ways-families-can-encourage-girls-interest-computing>

through our simple message of, ‘I’m Latina and I love technology.’” TECHNOchicas PSAs have aired more than 10,000 times in more than 20 Univision markets across the country reaching millions of Spanish-speaking households. There are more than 320 TECHNOchicas Ambassadors throughout the United States and Puerto Rico, a network of college-aged Latinas who serve as near-peer role models to the younger students and facilitate hands-on computing workshops, provide guidance and practical advice on how to gain access to pathways into computing, and inspire girls to become the future of the tech workforce.

TECHNOchicas has held more than 500 outreach events nationwide,

where more than 5,000 Latinas have learned directly from their peer role models. For many, these events have been their first exposure to computing, and for others it has served as encouragement to pursue a path in computing. Providing exposure to computing fields through a shared cultural lens helps students create a computing identity at a young age and helps them persist. It is also important for Latinas pursuing computing degrees to have access to mentorship and role models that share a common cultural background. TECHNOchicas Ambassadors hold events around the country to expose girls to coding and other aspects of computing careers. Events are evaluated qualitatively. Participants report that they benefit

in various ways from exposure to Latina professionals, including:

- ▶ “... learning more about Latina women in the STEM Field and how they got to where they are today.”

- ▶ “...listening to the different backgrounds of the TECHNOchicas that led to where they are now”

- ▶ “...being around so many ambitious Latinas.”

- ▶ “...listening to the different backgrounds of the TECHNOchicas that led to where they are now.”

The ambassadors themselves are also positively impacted by their experiences in the TECHNOchicas program. The following quotes exemplify the types of impact reported:

- ▶ “TECHNOchicas is an inspiring and powerful community to be a part of. I’m not only surrounded by incredible leaders in STEM, but I’m also collaborating with them to inspire and shape the next generation of Latina Engineers.”

- ▶ TECHNOchicas motivates me to continue to be the best engineer I can possibly be while helping others become the best version of themselves as well.

At its core, TECHNOchicas seeks to inspire Hispanic girls and their families. Collected qualitative data suggests this effect is being realized. After attending an outreach event, girls report being inspired. Asked what they plan to do after attending their local event, responses include persisting in the face of challenges, focusing on doing well in school, and sticking with computing or engineering because they can see now where it will get them. Representative excerpts include:

As a result of attending today’s event, I plan to...

- ▶ Learn and teach others about computing;

- ▶ Continuing to work hard in my engineering class;

- ▶ Work even harder because my opportunities are endless.

Since TECHNOchicas began, there has been a promising trend upward in the percentage of Latinas taking the AP Computer Science Principles (CSP) exam. As The College Board (2019) notes, the CSP exam was designed to invite a more diverse group of students to technology. In 2017, 2,642 Latinas took the exam, reflecting six percent of all test takers. By 2019, 6,393



Latinas sat for the test, and represented 6.7% of the testing population. While there is no direct linkage between TECHNOLOchicas and the number of Latinas taking the CSP exam, it nonetheless encouraging that Latinas are receiving support and encouragement to consider educational and career pathways in technology.

Shaping the Color of Our Future

NCWIT, as well as other change leaders working to advance the Latinx community, including Latinas in STEM,^c Hispanic Heritage Foundation,^d and the Eva Longoria Foundation,^e can change the trajectory and economic prospects of Latinx families in the U.S., who can help fill the anticipated 3.5 million computing-related job openings by 2026.⁶ By being part of this change, we hope that future generations of Latinas will face fewer barriers to pursuing careers in computing. As women of color in this space, it is both rewarding and fills us with great pride to know that we are uplifting entire communities, and an entire nation.

The TECHNOLOchicas initiative has helped us see we have the potential to influence the participation of women and girls of color from other underrepresented backgrounds (including Black and Native American) in computing. We need more national programs and initiatives that employ an intersectional approach to address the shortage of Latinas and other girls of color in tech. Recently, NCWIT spearheaded *The Color of Our Future* initiative, which anchors NCWIT programs, initiatives, and research-based resources focused on broadening the meaningful participation of underrepresented women and girls of color to positively impact the future of computing.

Yet in order to shape the future of tech, we need both intersectional and mainstream approaches to equity and representation. The presence of targeted, intersectional programs should not erase the need for more inclusive programs targeted at broadening the participation of women and other underrepresented populations across the tech ecosystem. Computing pro-

grams and the overall tech industry should be intentional and holistic in their approaches and aim to be as intersectional and inclusive as possible.

TECHNOLOchicas has empowered many Latinas in tech, giving them a sense of belonging and confidence they often lacked. Helping Latinas of all ages see the potential they have to influence the technologies that are created in the future will be game-changing for this industry. **C**

References

1. Cantor, N. If not now, when? The promise of STEM intersectionality in the twenty-first century. *Peer Review* 16, 2 (2014), 29–31.
2. Charleston, L.J. Intersectionality and STEM: The role of race and gender in the academic pursuits of African American women in STEM. *Journal of Progressive Policy and Practice* 2, 3 (2014), 17–30.
3. Denner, J. The role of the family in the IT career goals of middle school Latinas. In *AMCIS 2009 Proceedings* 334 (2009); <http://aisel.aisnet.org/amcis2009/334>
4. Denner, J., Bean, S., and Martinez, J. The girl game company: Engaging Latina girls in information technology. *Afterschool Matters* 8, (2009), 26–35.
5. Deruy, E. Where being a Latina computer scientist is the norm. *The Atlantic* (Dec. 17, 2015).
6. DuBow, W. and Ashcraft, C. *The Importance of Complexity in Attending to Intersectionality*. NCWIT, Boulder, CO, 2016.
7. DuBow, W. and Gonzalez, J.J. *NCWIT Scorecard: The Status of Women in Technology*. NCWIT, Boulder, CO, 2020.
8. Google Inc. and Gallup Inc. (2016). Diversity Gaps in Computer Science: Exploring the Underrepresentation of Girls, Blacks and Hispanics. (2016); <http://goo.gl/PG34aH>.
9. Ireland, D.T (Un)hidden figures: A synthesis of research examining the intersectional experiences of Black women and girls in STEM education. *Review of Research in Education* 42, (2018), 226–254.
10. Margolis, J. *Stuck in the Shallow End: Education, Race, and Computing*. Massachusetts Institute of Technology Press, 2008.
11. McAlear, F. *Data Brief: Women of Color in Computing*. 2018; <https://www.wocincomputing.org/wp-content/uploads/2018/08/WOCinComputingDataBrief.pdf>.
12. Participation of AP Computer Science Principles More than Doubles 3 Years After Launch (2019); <https://www.collegeboard.org/releases/2019/participation-csp-nearly-doubles>.
13. Rodriguez, S. L., Cunningham, K., Jordan, A. STEM identity development for Latinas: The role of self and outside recognition. *Journal of Hispanic Higher Education*. (2017); DOI: 10.1177/1538192717739958.
14. Scott, A. *The Leaky Tech Pipeline: A Comprehensive Framework for Addressing and Understanding the Lack of Diversity Across the Tech Ecosystem*. Kapor Center for Social Impact, 2018.
15. Scott, K.A., Sheridan, K.M., and Clark, K. Culturally responsive computing: A theory revisited. *Learning, Media, & Technology*, (2014), 1–25.
16. U.S. Bureau of Labor Statistics. *Occupational Outlook Handbook: Computer and Information Technology Occupations*, 2019.
17. U.S. Department of Labor, *Monthly Labor Review*, 2017.
18. Villa, E. Q. Engineering education through the Latina lens. *Journal of Education and Learning* 5, 4 (Apr. 2016), 113–125.

Jannie Fernandez (jannie@ncwit.org) as the Director for the K–12 Alliance and TECHNOLOchicas. Prior to her work at NCWIT, for over 10 years, she taught biology and physics, and served as Special Education Department Head for a public high school in Miami, FL, USA.

JeffriAnne Wilder (j.wilder@ncwit.org) Senior Research Scientist for NCWIT in Boulder, CO, USA, is a sociologist and leading scholar specializing in diversity, race relations and women's empowerment.

Copyright held by authors.

Coming Next Month in COMMUNICATIONS

Becoming an 'Adaptive' Expert

Improving Social Alignment during Digital Transformation

Keeping CALM: When Distributed Consistency Is Easy

Dark Patterns: Past, Present, Future

Is Persistent Memory Persistent?

Integrating Management Science into HPC Research

On the Value of Spaciotemporal Information

Flood Risk Analysis on Terrains under Multiflow Direction Model

Plus the latest news about living robots, edge AI, and virtual collaborations in the age of Covid-19.

c See <http://www.latinasinstem.com>

d See <https://hispanicheritage.org/>

e See <https://evalongoriafoundation.org/>



Kode Vicious Broken Hearts and Coffee Mugs

The ordeal of security reviews.

Dear KV,

I am working on a project that has been selected for an external security review by a consulting company. They are asking for a lot of information but not really explaining the process to me. I cannot tell what kind of review this is—pen (penetration) test or some other thing. I do not want to second-guess their work, but it seems to me they are asking for all the wrong things. Should I point them in the right direction or just keep my head down, grin, and bear it?

Reviewed

Dear Reviewed,

I have to say that I am not a fan of keeping one's head down, or grinning, or bearing much of anything on someone else's behalf, but you probably knew that before you sent this note. Many practitioners in the security space are neither as organized nor as original in their thinking as KV would like. In fact, this is not just in the security space, but let me limit my comments, for once, to a single topic.

Overall, there are two broad types of security review: white box and black box. A white-box review is one in which the attackers have nearly full access to information such as code, design documents, and other information that will make it easier for them to design and carry out a successful attack.



A black-box review, or test, is one in which the attackers can see the system only in the same way a normal user or consumer would.

Imagine you are attacking a consumer device such as a phone. In a white-box situation, you have the device, the code, the design docs, and everything else the development team came up with while building the phone; in a black-box case, you have only the phone itself. The pen-test idea

currently has credence in security circles, but, candidly, that is just a black-box test of a system. In point of fact, the goal of any security test or review is to determine if an attacker can carry out a successful attack against the system.

Determining what is or is not a successful attack requires the security tester to think like the attacker, a trick KV finds easy, because at heart (what heart?) I am a terrible person whose first thought is, "How can I break this?"

Security testing is often quite easy because of the incredibly low overall quality of software and the increasingly large number of software modules used in any product. To paraphrase Weinberg's Second Law, "If architects designed buildings the way programmers built programs, the first woodpecker that came along would destroy all of society." The difficult parts of security work are constraining the attacks to those that matter and getting past those developers with a modicum of clue who are able to build systems that at least resist the most common script kiddie attacks.

Your letter seems to imply your external reviewers are interested in a white-box review since they are asking for a great deal of information, rather than just taking your system at face value and trying to violate it. What to expect from a white-box security review, at least at a high level, should not be a surprise to anyone who has ever participated in a design review, as the two processes should be reasonably similar. The review would work in a top-down fashion, where the reviewer asks for an overall description of the system, hopefully enshrined in a design document (please have a design document); or the same information can be extracted, painfully, through a series of meetings.

Extracting a design in a review meeting takes a great deal longer in the absence of a design document but, again, looks similar to a design review. First, there must be a lot of coffee in the room. How much coffee? At least one pot per person, or two if you have KV in the room. With the coffee in place, you need a large whiteboard, at least two meters (six feet) long.

Then we have the typical line of interrogation: "What are the high-level features?" "How many distinct programs make up the system?" "What are they called?" "How do they communicate?" and for each program, "What are the major modules of this program?" KV once asked a software designer after he had filled a four-meter whiteboard with named boxes, "What's the architecture that holds all this together?" to which the answer was, "This system is too complex to have an architecture." The next sound was KV's glasses clattering on the table and a very heavy sigh. Needless to say,

The goal of any security test or review is to determine if an attacker can carry out a successful attack against the system.

that piece of software was riddled with bugs, and many were security related.

A good reviewer will have a minimal checklist of questions to ask about each program or subsystem, but nothing too prescriptive. A security review is an exploration, a form of spelunking, in which you dig into the dirty, unloved corners of a piece of software and push on the soft parts. Overly prescriptive checklists always miss the important questions. Instead, the questions should start broad and then get more focused as issues of interest appear—and trust me, they always will.

When issues are found, they should be recorded, though perhaps not in an easily portable form, since you never know who else is reading your ticketing system. You want to get inside a system and go read the bugs. If you have a bad apple or two inside the company (and what company is free of rotten apples?) and they do a search on "Security P1," they are going to walk away with a lot of fodder for zero-day attacks against your system.

Once the system and its modules have been described, the next step is to look at the module application programming interfaces (APIs). You can learn a lot about a system and its security from looking at its APIs, though some of what you will learn will never be able to be unseen. It can be pretty scarring, but it has to be done.

The APIs have to be looked at, of course, because they show what data is being passed around and how that data is being handled. There are security scanning tools for this type of work, which can be used to direct you toward where to perform code reviews, but it is often best to spot-check the APIs your-

self if you have any type of ability or intuition around security.

Lastly, we come to the code reviews. Any reviewer who wants to start here should be fired out of a cannon immediately. The code is actually the last thing to be reviewed—for many reasons, not the least of which is that unless the security-review team is even larger than the development team, they will never have the time to finish reviewing the code to sufficient depth.

Code reviews must be targeted and must look deeply at the things that really matter. It is all of the previous steps that have told the reviewers what really matters, and, therefore, they should be asking to look at maybe 10% (and hopefully less) of the code in the system. The only broad view of the code should be carried out, automatically, by the code-scanning tools previously mentioned, which include static analysis. The static analysis tools should be able to identify hot spots that the other, human reviewers have missed.

With the review complete, you should expect a few outputs, including summary and detailed reports, bug-tracking tickets that describe issues and mitigations (all while being secured from prying eyes), and hopefully a set of tests the QA team can use to verify that the identified security issues are fixed and do not recur in later versions of the code.

It is a long process littered with broken hearts and coffee mugs, but it can be done if the reviewers are organized and original in their thinking.

KV

Related articles on queue.acm.org

How to Improve Security?

Kode Vicious

<https://queue.acm.org/detail.cfm?id=2019582>

Security Problem Solved?

John Viega

<https://queue.acm.org/detail.cfm?id=1071728>

Pickled Patches

Kode Vicious

<https://queue.acm.org/detail.cfm?id=2856150>

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.

Copyright held by author.

▶ Mark Guzdial, Column Editor

Education

Data-Centricity: A Challenge and Opportunity for Computing Education

Rethinking the content of introductory computing around a data-centric approach to better engage and support a diversity of students.

ON A GROWING number of campuses, data science programs offer introductory courses that include a non-trivial amount of programming. The content of such courses overlaps that of traditional computer science introductory courses, but neither course subsumes the other. This situation creates challenges for students. Introductory courses should help students decide which disciplines to pursue further, but misaligned data science and computer science introductions leave students unable to switch between areas without starting over. This in turn puts pressure on departments to determine how to accommodate students as they explore their curricular options. In universities where finances follow student enrollments, overlaps such as these can also lead to turf wars and other tensions that adversely impact students.

We view “data science” as the process of answering questions about the world through the application of (usually statistical) computational methods to data. Data scientists benefit from some computing background. In recent years we have also seen the rise of the related profession of “data engineers,” who need a substantial computing background. In addition, the central role of data across computing demands CS majors with basic data-science skills. Therefore, for

the sake of this broad spectrum of students, it is time to rethink the content of introductory computing. We believe the approach described in this column—*data-centric introductory computing*—can support and engage students with diverse interests.

Introductory Data Science as a Critique of Introductory Computing

A conventional CS1 course,¹ which emphasizes control structures and imperative programming, aligns poorly with the learning goals of aspiring data engineers or data scientists. Data science uses rich functions that transform and compute with sophisticated data, both of which are far from the focus of CS1. The explicit manipulation of aggregated data—such as cleaning, splitting, or refactoring—is best done through operations that more closely resemble queries and higher-order functions,⁴ both of which occur much later in traditional curricula. Data science works with *real* datasets that touch on real-world problems, rather than artificial starter problems based on numbers, strings, and arrays. In all of these ways, conventional CS1 courses appear inauthentic or irrelevant for data-facing students. Moreover, the end goal of conventional early CS courses is usually to prepare students for more CS courses, rather than to prepare them to continue to learn addition-

al data-relevant programming content as their interests demand.

However, CS1 courses that eschew these trends don’t only fail to engage the budding data scientist: they also do a disservice to computer science students. There are innumerable datasets that cover a very broad span of human interests, from politics and economics to sports and voting in music competitions. Through their use, computing can be seen even by novices for what it now is: a field that engages with humans and the world. This alters some of the standard perceptions that keep away many groups of students. The introduction of data also makes it easy to concretely illustrate the social consequences and perils of data-driven decision making, even to novices. Thus, *we should view the rise of data science curricula as a criticism of computing curricula, and work to address our shortcomings.*

Why Not a Separate Data Science Track?

Given these weaknesses of CS1 (which often extend into CS2), perhaps the rise of parallel data science tracks (let’s call them DST) presents the solution? Sadly, the DST courses we have reviewed take a rather ad hoc view of computing and programming. There is little emphasis on program design, software testing, or the impact of data structure on com-

plexity. These are not merely important issues: they are critical. Even small programs may end up influencing policy decisions or research findings. We have to be able to trust the code and the results.

Many data scientists, and especially data engineers, will want to take more (traditional) CS courses such as cloud computing, security and privacy, software engineering, visualization, and databases. The limited computing preparation that a DST would offer leads to one of three outcomes, all unhappy:

- Each of these courses needs to be reconstructed in a separate “data science department” or equivalent, resulting in a huge duplication of labor and cost (which is simply infeasible at many institutions).

- Students will arrive in these upper-level computing courses without the relevant computing prerequisites. These students will have vastly poorer educational outcomes and create a significant burden for faculty.

- Those upper-level courses will have to scale back their prerequisites, resulting in weaker curricula for traditional computing students.

In general, many students who are offered both DST and CS1 options lack sufficient understanding of either discipline to predict which focus best aligns with their interests. In educational systems (like in the U.S.) where students can easily move between disciplines in college, the choice between DST and CS1-CS2 is premature.

Reform Introductory Computing!

We suggest that there is a strong alternative: to integrate data-science components into introductory computing. This requires a significant rethinking of what introductory computing looks like, and is a major departure from the traditional CS1-CS2 structure. What might this rethinking entail?

We should begin the curriculum in “data science” with some basic data engineering. That is, students begin right away with datasets of some complexity reflecting real-world questions. Perhaps surprisingly, we believe that even the choice of representation matters, and recommend focusing specifically on data represented in *tabular* formats. This offers many advantages:

- Innumerable real-world data is published as tables. Students can pick

data sets of interest to them (within some constraints), enabling them to personalize their education and feel more invested in it. Exercises become much less artificial.

- Even quite young students understand tables instinctively,² and have experienced them in everything from middle-school math classes to spreadsheets.

- Despite this, tables are a fairly sophisticated data structure: an unbounded homogenous sequence of bounded heterogeneous structures. Getting to this point in a traditional computing curriculum takes quite a while.

- Tables (especially tidy⁵ ones) are inherently parsed, eliminating a complex and imprecise step that greatly complicates other interesting data formats like text.

In short, tables are a “sweet spot” in introductory computing.

Doing useful work also requires some basic statistics. However, most students enter college with a rudimentary knowledge of some operations such as central tendencies. In our experience, even mathematically nervous non-majors are able to successfully apply them with reasonable amounts of support. Teaching basic visualization generates meaningful artifacts beyond code. Furthermore, showing students core programming concepts such as conditionals (for selecting data or iden-

tifying data-entry errors) and functions (for running experiments on datasets with similar column structure) not only enables them to complete projects with experimental data analysis, it also grounds typical programming content in a rich and meaningful context.

From Data to the Rest of Computing

Tables are, of course, not a universally appropriate data structure. Once students start engaging with the world, they soon encounter many other kinds of natural structures to represent, whether sport competition brackets or genealogical information or travel networks (respectively, usually, trees, DAGs, and graphs). While these can be encoded in tables, this requires varying degrees of inelegance, which students can easily see.

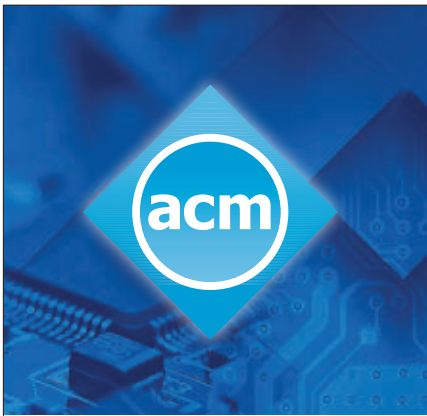
This, of course, is where computing really shines. For decades we have built theory and languages for complex data structures: to represent them, program over them, and reason about them. Thus, these data structures—which represent a limitation of tables—are a perfect point of transition to traditional computer science, through data structures and systematic programs over them. Not only are non-tabular datatypes more natural in these cases, they also offer different affordances, and can even confer significant performance advantages. We can get to these points potentially earlier than we would in traditional CS1-CS2; with much greater motivation; and with students already well-equipped to do useful work through their exposure to tables. From a computing perspective, this should be regarded as a win-win.

While data scientists arguably do not need all these data structures, we believe exposing students to the limitations of tables through concrete data-facing examples is valuable in and of itself. This also represents a branch point in the curriculum. Those certain they do *not* want to learn more computing can move on to statistics and other disciplines (armed with a solid introduction to programming and computing), while the rest can proceed with a more traditional computing path.

Data-Centricity

In short, we propose a rethinking of traditional CS1-CS2 with rudimentary data science preceding, and fluidly leading

The end goal of conventional early CS courses is usually to prepare students for more CS courses, rather than to prepare them to continue to learn additional data-relevant programming content as their interests demand.



Advertise with ACM!

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



Request a media kit with specifications and pricing:

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



In short, we propose a rethinking of traditional CS1-CS2 with rudimentary data science preceding, and fluidly leading to, more traditional data structures.

to, more traditional data structures. We call this approach *data-centric computing*, with the formula

$$\text{Data-centric computing} = \text{data science} + \text{data structures}$$

(in that order: this + is not commutative). We have been prototyping this approach for a few years at Brown University, with great success, and have learned a good deal from it.

This shift certainly presents challenges. It requires a fairly different introductory pedagogy that has not been developed well in the computing education community. Everything from programming methodologies to problem decomposition to topic ordering, and the interleaving of statistics with programming, is open and needs study. Mathematics education techniques like notice-and-wonder,³ which was popularized for data reasoning by the *New York Times*,^a becomes relevant. Students need to be taught to contend with precision and accuracy; we have to determine the right way to introduce data cleansing; and so on. We therefore call on the computing education community to rethink its approaches, prescriptions, and open questions.

The Promise

Moving introductory computing to data-centricity can open many doors. Other subjects, given an accessible, useful introductory computing course, may rethink their approach. A new generation of students will be able to com-

municate across disciplines in the language of computation. Computing's application to a wide variety of disciplines will stand up front and center, without compromising the core of our subject. Our goal, therefore, should be to create *at scale* the new kinds of partnerships that currently only happen in very specialized settings.

A data-centric introduction to computing also buys students time in curricula that give them choice. They do not have to prematurely commit to "data science" or "computing," which they are likely doing on the basis of what they have heard from parents, friends, and newspaper articles. Instead, they can experience the two subjects together for themselves before having to choose between the two (and, even if they pick a third, they will have a useful, applied understanding of computing).

Even in a time of exploding enrollments, we owe it to our students to ask these questions and act on the answers. It is easy to think we are successful in a hot marketplace, but the integration of data science could create curricula of lasting value even in a weak one. At some (perhaps more resource-rich) institutions this will take the path of whole-cloth reform; at others (more resource-starved), educators will have to find an incremental path. All this provides an opportunity and imperative for educators and researchers, and (reflecting the breadth of the field) contributions can come from people with many different kinds of expertise (pedagogy, statistics, visualization, programming languages, and more). □

References

1. *Computer Science Curricula 2013 Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society (Dec. 2013); <https://bit.ly/3drQjA9>
2. Konold, C., Finzer, W. and Kreetong, K. Modeling as a core component of structuring data. *Statistics Education Research Journal* 16, 2 (Nov. 2017), 191–212.
3. Ray-Riek, M. *Powerful Problem Solving: Activities for Sense Making with the Mathematical Practices*. Heineman, 2013
4. Wickham, H. *Advanced R*. Second edition. Chapman and Hall/CRC The R Series, 2019.
5. Wickham, H. Tidy Data. *Journal of Statistical Software* 54, 10 (2014); DOI: 10.18637/jss.v059.i10

Shriram Krishnamurthi (sk@cs.brown.edu) and **Kathi Fisler** (kfisler@cs.brown.edu) are both faculty members in computer science at Brown University, Providence, RI, USA. They also co-direct the Bootstrap outreach project, which translates versions of these ideas to middle- and high-school contexts.

a <https://nyti.ms/3dwwaNe>

Viewpoint

OMSCS: The Revolution Will Be Digitized

Lessons learned from the first five years of Georgia Tech's Online Master of Science in Computer Science program.

THE ONLINE MASTER of Science in Computer Science (OMSCS), a degree program at the College of Computing (CoC), Georgia Institute of Technology, grew out of a conversation I had with Sebastian Thrun (co-founder of online learning platform Udacity) in September 2012. We set as our goal to expand access to quality learning opportunities by using massive open online course (MOOC) technology to mitigate obstructions of time, space, and financial ability. This called for a fundamental, revolutionary shift from the prevailing paradigm of higher education, in which a brand is bolstered by exclusion and high tuition fees.

As a preliminary step, I, as dean of CoC, convened a faculty working group, chaired by Kishore Ramachandran, from the School of Computer Science. Its members were notably concerned with maintaining the quality of CoC's academic content, the logistics, and the student and faculty experience—a focus on quality that would become a key principle of the program. The document the group created became the operational manual for putting the program into practice. The working group, in turn, engaged with the faculty through a series of deliberations and town hall meetings, and in spring 2013 the faculty voted to move forward. The program then



earned the support of Georgia Tech's President and Provost, who advocated for it with the Board of Regents of the University System of Georgia. While the faculty debated and planned, Thrun and I sought funding to cover the costs of preparing, organizing, and introducing the first courses. In January 2013, AT&T provided a \$2 million gift, and added \$2 million more a year later. AT&T's generous support signaled to Georgia Tech the potential of the program, and enabled OMSCS to have positive net income from the start. When, in May 2013, the Board of

Regents approved the degree, we began preparing the first courses, using Udacity's platform and their course design and production experience. Each of the initial five courses cost approximately \$300,000 to develop. In January 2014, OMSCS was launched with 380 students.

Progress and Service

We took the words emblazoned on the seal of Georgia Tech—"Progress and Service"—as our mission. To start with, we committed the program to a unique admissions policy—GRE is not

required, and instead, OMSCS students have to obtain grade B or higher in two courses from a specified list in their first year to be officially admitted (for admission requirements see <http://www.omscs.gatech.edu/program-info/admission-criteria>). While the selectivity of the on-campus Master in Computer Science program (MSCS) is slightly higher than 10%, 70.7% of the more than 26,000 OMSCS applicants were admitted. Added to the novel admissions policy, I insisted on keeping OMSCS tuition affordable—less than \$7,000 for the full degree, payable by course, rather than \$40,000 for a public on-campus program, or \$70,000 or more in a private university.

CoC's MSCS offers a degree on completing a course option (10 courses, 30 credit hours), thesis or project options (each counting for nine credit hours). The course option is the only one available to OMSCS students as it is difficult to scale up the others. However, individual OMSCS students obtained a degree following one of the latter options. According to Goel and Joyner³ the data strongly suggests there is nothing inferior about the online course experience, students regularly rate their online courses as better than on-campus courses they have taken, and they regularly match or exceed the performance of their on-campus counterparts.

The demographics of OMSCS differ from MSCS: the average age of a starting OMSCS student is 32 as compared with 22 in MSCS, the majority of OMSCS students are domestic (67.1% in Spring semester 2019), while MSCS' is international (55.4%), most work a full-time job and their backgrounds are more diverse (in the academic years 2017–2018 and 2018–2019 70% of the applicants lacked undergraduate CS degree, 17% had a non-CS MS, and 5% had a Ph.D.). OMSCS attracted slightly more underrepresented minorities (14% vs. 10%). Goodman et al.⁴ showed most applicants would not pursue an advanced degree at all if it were not online and highly affordable and that OMSCS provides the first rigorous evidence online education can increase educational attainment. The different demographics of our online and residential programs have shown that the former has not cannibalized the latter, for which the number of applications has more than doubled.

OMSCS' growth has been phenomenal—by Spring 2019 term OMSCS offered a total of 30 courses in four specializations to 8,662 students (for a current list of courses and course previews, see <https://www.omscs.gatech.edu/current-courses>). Goodman et al.⁴ predicted OMSCS will be responsible for at least 7% increase in the number of master's degrees in computer science attained each year in the U.S. In fact, it now exceeds 10%.

The implications of OMSCS have not gone unnoticed, and the program has been widely recognized both inside and outside of higher education. In 2017, the University Professional and Continuing Education Association gave OMSCS its National Program Excellence Award. In 2017, Georgia Tech appeared on *Fast Company's* list of most innovative companies in the world—the third university recognized on the list, and the first recognized for education rather than research—on the strength of the OMSCS program. And, the OMSCS was cited in more than 1,200 news articles, including more than 50 in the *Chronicle of Higher Education* and *Inside Higher Ed*.

OMSCS' success has inspired similar programs at other universities—more than 40 MOOC-based M.S. degree programs, more affordable than their on-campus counterparts, have been launched recently by more than 30 universities. Georgia Tech too has launched two additional online master's degrees.

What Has Not Gone the Way We Thought It Would?

From the start we determined to expand the program judiciously, out of

The implications of OMSCS have not gone unnoticed, and the program has been widely recognized both inside and outside of higher education.

respect and responsibility to the “pioneers”—the first student cohort. Thus, we admitted only half of the accepted students in the first semester, delaying the entry of the rest to the next semester. All went well.

In the first year, the Teaching Assistants (TAs) were on-campus students, but as the number of students rose we were faced with the dearth of TAs. In the second year, we advertised for TAs among OMSCS students and were surprised, even astounded, by their overwhelming positive response. Research by Joyner⁵ found online TAs were far more likely to be intrinsically or altruistically motivated. The students, typically computing professionals with full-time jobs and family, serve as TAs even after graduating—for modest pay, and sometimes even on a pure volunteer basis: in Spring semester 2019, 65 out of the 282 TAs were OMSCS alumni, in addition to 109 current OMSCS students.

Another issue was student services (for example, the Office of Student Life, registration, admissions, bursar, career services, and so forth), many of which were not designed for our scale. By careful integration and cooperation we assure OMSCS students receive the same services and policies as on-campus students. For example, an older student body (that is, with families, aging parents, young children) requires more support for unforeseen health or other personal emergencies, and a geographically dispersed student body is likelier to be affected by natural disasters of which the university administration is not otherwise aware. To reduce the burden on the school's central Office of Student Life, which is responsible for validating and responding to requests for special accommodation, an application is used to track such cases and (with input from instructors) to determine the response.

Aware that student retention is critical for the success of OMSCS, we carefully monitor its metrics. 60.5% of those who started in the years 2014–2017 have graduated or were still enrolled in the Spring 2019 term. Approximately 15%–20% may drop out during any semester, yet all but 5%–6% return in the next semester. Students might need or wish to withdraw or leave the program for various reasons—some did not intend to obtain a degree in the first place, but rather update/upgrade

their skills, and others perhaps had not anticipated the *rigorosity* of the program. At five years, it is too early yet to deduce retention, but we ascribe the relatively high retention thus far in part to the peer social connectedness.

Among the frequently cited criticisms of online education is the lack of opportunity to experience the benefits of teacher-student and student-student interactions. Our concerns were allayed as we soon discovered OMSCS engendered a palpable spirit of community and of service. OMSCS students have created and led more than 70 online forums—entirely student-run social communities based on shared geography, interests, or background—where students can network, ask for help, and form the kinds of relationships they might form on campus. Students across states and countries (all 50 U.S. states, 120 countries) coordinate projects across time zones, share solutions, and support each other—and offer advice to prospective applicants. The OMSCS program’s sense of community effortlessly spans the globe.

Lessons Learned

We learned OMSCS serves a large unmet demand, underserved by the institutions of higher education. Faculty and administration in some institutions of higher education have been skeptical of online programs being able to provide as high quality education as residential ones, concerned that offering those programs will devalue their residential programs, and mindful of possible slippage in popular rankings that privilege student-faculty ratios.¹ We learned an inclusive admission policy does not detract from students’ attainment, nor from the college reputation.

Bacow et al.¹ acknowledged “very few institutions are using either the savings from online education or the incremental revenue to reduce the price of education to students.” At the time OMSCS was established, many institutions charged tuition equal to or greater than the tuition charged to residential programs. We learned that high-quality low-cost online degrees are realizable and viable—OMSCS has been financially self-maintaining since its third year (in the first two years it was in the black thanks to AT&T), and thus far has produced cumulative net income to Georgia Tech of \$13 million.

We believe the MOOC-based technology that powers the OMSCS degree can expand the availability of computer science in K–12 schools.

We learned that for a radical change in higher education to happen it must be led and supported by the faculty. Carmean and Friedman² described how, at other institutions “...money, control, job security, tradition, and quality” precipitated into discussions “... within a faculty divided against itself, [that] too often disintegrate into questions of governance and control.” We avoided “the online education tin-dertbox” by respecting and addressing our faculty’s concerns.

Expanding Tomorrow’s Opportunity


The realignment of today’s workforce with tomorrow’s economy requires more than plugging shortages of master’s degrees. The shortfall in technology education permeates every level of study. There are 1.6 million students in K–12 education in the state of Georgia alone, but only 95 qualified CS teachers. We believe the MOOC-based technology that powers the OMSCS degree can expand the availability of computer science in K–12 schools. The Constellations Center for Equity in CoC is developing a hybrid model of online and in-person instruction that will allow far more students access to quality CS education.

CoC already offers its MSCS students the use of the OMSCS videos. It offers undergraduates the choice of an online version of an introductory computing course (Introduction to Computing with Python). In the Spring 2019 semester, 55% of the undergraduates made that choice and reported liking it as much or more than their in-person courses. Two more introductory courses were offered undergraduates in the Fall 2019 semester. These courses will

be available to University System of Georgia students and to Georgia high school students. The addition of online courses may help the students reduce their time on-campus, attain graduation sooner, and reduce the cost of college education by taking introductory courses before reaching Georgia Tech and then combine online learning with internships, co-ops, and while working.

Georgia Tech and CoC are dedicated to finding ways to use technology to expand the impact of CS education. As a public university, our responsibility is both to the students we serve and to the nation, and OMSCS pioneers the change in the educational landscape with both responsibilities in mind. Our future depends upon it.

Postscript

In the Spring 2020 term, 9,597 students enrolled in OMSCS, almost 1,500 graduated in this academic year. In December 2019 the conference “Reimagine Education” presented OMSCS with the Gold Award for the best distributed/online program for nurturing 21st-century skills. In May 2020 the University of Cambridge announced it will move all lectures online for the full 2020–2021 academic year. 

References

1. Bacow, L.S. et al. Barriers to Adoption of Online Learning Systems in US Higher Education, 2012; <https://www1.udel.edu/edtech/e-learning/readings/barriers-to-adoption-of-online-learning-systems-in-us-higher-education.pdf>
2. Carmean, C. and Friedman, D. Conjecture, Tension, and Online Learning, 2014; <https://er.educause.edu/articles/2014/2/conjecture-tension-and-online-learning>.
3. Goel, A. and Joyner, D.A. Using AI to teach AI: Lessons from an online AI class. *AI Magazine* 38, 2 (2017), 48–58.
4. Goodman, J., Melkers, J., and Palais, A. Can online delivery increase access to education. *Journal of Labor Economics* 37, 7 (2019), 1–34.
5. Joyner, D.A. Scaling expert feedback: Two case studies. In *Proceedings of the Fourth Annual ACM Conference on Learning at Scale*, Cambridge, MA, (2017).

Zvi Galil (galil@cc.gatech.edu) is the Frederick G. Storey chair of computing and Executive Advisor to online programs at Georgia Institute of Technology, where he is emeritus dean of computing, Atlanta, GA, USA.

I thank Sebastian Thrun, who suggested we create a MOOC-based master program, Udacity and AT&T for partnering with us. I thank the faculty of the College of Computing at Georgia Tech for embracing OMSCS and to the faculty and staff for their ceaseless efforts to build the program. In particular, I thank Charles Isbell (at the time executive associate dean, now dean) and David Joyner (now executive director of OMSCS) and David White who ably served as the executive director of OMSCS during its first six years. I am indebted to Rich DeMillo for his support and advice, to Moshe Vardi (former editor-in-chief of *Communications*) for inviting me to write this Viewpoint. I am beholden to David Joyner and the referees for their comments on an earlier draft.

Copyright held by author.

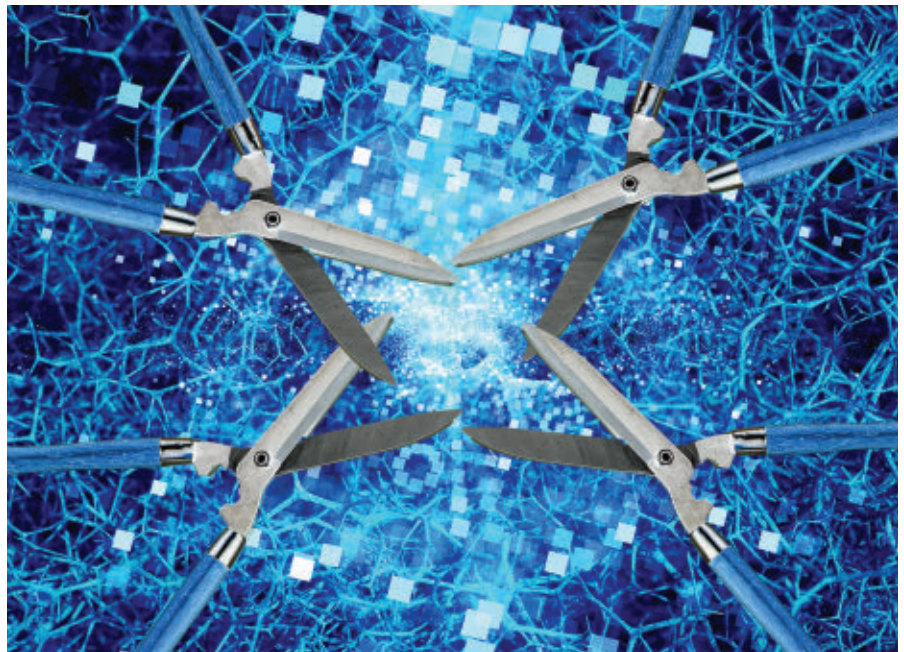
Viewpoint

Thorny Problems in Data (-Intensive) Science

Data scientists face challenges spanning academic and non-academic institutions.

AS SCIENCE COMES to depend ever more heavily on computational methods and complex data pipelines, many non-tenure track scientists find themselves precariously employed in positions grouped under the catch-all term “data science.” Over the last decade, we have worked in a diverse array of scientific fields, specializations, and sectors, across the physical, life, and social sciences; professional fields such as medicine, business, and engineering; mathematics, statistics, and computer and information science; the digital humanities; and data-intensive citizen science and peer production projects inside and out of the academy.^{3,7,8,15} We have used ethnographic methods to observe and participate in scientific research, semi-structured interviews to understand the motivations of scientists, and document analysis to illustrate how science is assembled with data and code. Our research subjects range from principal investigators at the top of their fields to first-year graduate students trying to find their footing. Throughout, we have focused on the multiple challenges faced by scientists who, through inclination or circumstance, work as data scientists.

The “thorny problems” we identify are brambly institutional challenges associated with data in data-intensive science. While many of these problems are specific to academe, some may be shared by data scientists outside the



university. These problems are not readily curable, hence we conclude with guidance to stakeholders in data-intensive research.

The Janitors of Science

Within data-intensive science, it is a truth universally acknowledged that a dataset in need of analysis must first be cleaned. This dirty job falls to the data scientist. Though the computational machinery of science has allowed new forms of scientific inquiry—and new kinds of scientists—to be developed, the machinery is fickle and only accepts pristine datasets. Yet the process of cleaning datasets is often hidden or ren-

dered invisible by disciplinary and organizational divisions.¹⁴ While even the simplest dataset must be massaged prior to use, the problem multiplies when instrument calibration degrades or automated pipelines are changed without notice. One interviewee suffered an instrument malfunction during a remote sensing experiment. Unknowingly, one in an array of sensors failed out of calibration range during a field study, but the automated pipeline continued to generate data, which had to be painstakingly cleaned in the following weeks. In scientific fields that produce comparatively small amounts of data, cleaning is often done manually in a spreadsheet,

and problems spotted visually, but with bigger data comes bigger spills that require bigger cleanups.

Continuing Education in Science

Early champions of “big data” infamously predicted an “end of theory,”¹ arguing that with enough data and computation, all research questions become simply an abstract problem of data processing. In contrast to this anti-disciplinary discourse, we see academic data scientists struggling to master the subject expertise necessary to make competent decisions about how to capture, process, reduce, analyze, visualize, and interpret research data. Domain scientists work closely with data scientists to model scientific problems, relying on common understanding to develop a team’s data pipeline and computational infrastructure. As a result, the integrity of the research process can rest with data scientists. In such settings, data scientists must develop “interactional expertise”⁵ by learning how to speak the jargon and conceptual vocabulary of a given discipline, and, more cogently, learning to ask the right questions of disciplinary scientists. Interactional expertise is not a skill that is readily taught in formal settings, particularly in traditional disciplinary degree programs. In response, data scientists gain interactional expertise in the fields in which they work by tactics such as making vocabulary lists of disciplinary jargon, quizzing colleagues in the hallway before a meeting, attending department seminars, taking classes, and reading literature of multiple domains.

The Overwhelmingness of Openness

Data-intensive science is increasingly tied to practices of, and policies for, “open science.”^{12,13} Open science spans open access publications, open datasets, open analysis code, open source software tools, and much more. The concept spreads over a myriad of tools, platforms, frameworks, and practices that change often. Conflicts arise between tools that are built on open source ecosystems and controlled by a mix of public and private entities, ranging from file formats to high-performance computing infrastructures. Managing so many overlapping mecha-

By bringing attention to these thorny problems, we aim to promote further discussion of the role of data science both inside and outside of data-intensive science.

nisms can be overwhelming, especially when data scientists are hired to take the burden of maintaining infrastructures off the backs of domain researchers.¹¹ Today’s scientific training may provide solid fundamentals for early career work, but rarely provides the skills necessary to keep pace with a fast-changing, complex ecosystem. Research groups face difficult trade-offs between migrating to new tools and maintaining old packages, versions, and formats that work well enough—and are often embedded in legacy systems that must be maintained. These trade-offs can place data scientists in uncomfortable mediating positions, similar to when they must translate between different disciplines.

Scarcity of Career Paths

Despite the rapidly growing need for data scientists in scientific research collaborations, these roles can lack specific job descriptions, and therefore a career path.⁷ Data scientists are often part of a research personnel pool that moves from project to project within a university. Few of these jobs lead to faculty positions or other secure career tracks. Even in scientific enterprises that invest in computational infrastructure for data, we rarely find career advancement systems that include data-specific tracks. Those exceptions we have encountered occur outside university departments, such as large-scale, globally distributed research projects with significant division of labor. The scarcity of career paths for those with combined expertise in a scientific domain and information technology results in a pro-

found loss of research capacity for universities. Whether individuals entered academic data science jobs as a career choice or as a byway en route to a faculty post, the lack of perceived upward mobility is resulting in departures for industry or other sectors.

Managing Infrastructures for the Long Term with Short-Term Funding

Scientific infrastructures accrete over long periods of time. Laboratories are constructed, equipment acquired, staff hired and trained, software and tools developed, journals and conferences launched, and new generations of scientists educated and graduated. Data scientists are increasingly responsible for maintaining the continuity of essential knowledge infrastructures, yet projects may outlast individual grants, leaving data scientists to operate in conditions of uncertainty about the long-term future of the infrastructure they build.⁹ This uncertainty poses complex challenges, both in terms of anticipating the needs of future users and of sustainability. In some scientific fields, the project life cycle unfolds on the scale of decades, in distinct stages such as initial conception, setting scientific goals, designing data management systems, constructing instruments and facilities, collecting data, processing data through pipelines, and releasing “science ready” data to the community. Builders of scientific infrastructure must make decisions in the present that will affect what data is collected and made available for decades, opening up some potential avenues of inquiry and closing down others.² Data-intensive science is plagued by the tyranny of small decisions; choices optimal in the short term may create a thorny nest of complications five or 10 years later.

Untangling Thorny Problems

The data-intensive science problems we have outlined here are intertwined with the organizational and funding of science within the university system.⁶ They only exist, and can only be addressed, within these larger institutional and political constraints. The specific circumstances of data science activities vary widely between and within the physical, life, biomedical, and social sciences; engineering, humanities, and other fields.



Association for
Computing Machinery

2018 JOURNAL IMPACT
FACTOR: 6.131

ACM Computing Surveys (CSUR)

ACM Computing Surveys (CSUR) publishes comprehensive, readable tutorials and survey papers that give guided tours through the literature and explain topics to those who seek to learn the basics of areas outside their specialties. These carefully planned and presented introductions are also an excellent way for professionals to develop perspectives on, and identify trends in, complex technologies.



For further information
and to submit your
manuscript,
visit csur.acm.org

Scientific practices in all of these fields are in flux, requiring new tools and infrastructures to handle data at scale, and grappling with new requirements for open science. Some individuals choose data science jobs in universities, but often the job finds them. Learning data science may be an investment that leads to a productive career, but all too often, time spent as the “data person” or “computer person” on the science team is labor not spent on dissertations, publications, or the scientific research that launches a tenure-track career.

These scientific environments have high personnel turnover rates, with individuals working in data science capacities through sequential post-doctoral fellow or grant-funded research scientist positions, or leaving for jobs in the corporate sector. Labor statistics are unlikely to capture the growth or turnover rate of these positions in science because the work is hidden behind so many different job titles. It is difficult to assess the damage to scientific progress when trusted data scientists move on to other institutions, as the losses may become apparent only months or years later. No matter how well code is documented, no paper trail can substitute for the rich domain expertise and tacit knowledge of those who conducted the science.^{4,10}

By bringing attention to these thorny problems, we aim to promote further discussion of the role of data science work both inside and outside of data-intensive science. Our list of problems is by no means exhaustive and our proposed remedies by no means complete. We offer our vignettes in the spirit of diagnosis and invite data scientists working in other fields, disciplines, and industries to contribute their own sets of thorny problems and solutions. We have written from the point of view of academic science as one permutation of data science, a term that escapes easy definition even as it advances. Much work remains. □

References

1. Anderson, C. The end of theory: The data deluge makes the scientific method obsolete. *Wired* 16. (2008); http://www.wired.com/science/discoveries/magazine/16-07/pb_theory
2. Baker, K.S., Duerr, R.E., and Parsons, M.A. Scientific knowledge mobilization: Co-evolution of data products and designated communities. *International Journal of Digital Curation* 10, 2 (2015); <https://doi.org/10.2218/ijdc.v10i2.346>
3. Borgman, C.L. *Big Data, Little Data, No Data: Scholarship in the Networked World*. MIT Press, Cambridge, MA, 2015.

4. Bowker, G.C. *Memory Practices in the Sciences*. MIT Press, Cambridge, MA, 2005.
5. Collins, H.M. and Evans, R. *Rethinking Expertise*. University of Chicago Press, Chicago, IL, 2007.
6. Edwards, P.N. et al. Science friction: Data, metadata, and collaboration. *Social Studies of Science* 41, 5 (2011), 667–690; <https://doi.org/10.1177/0306312711413314>
7. Geiger, R.S. et al. Career paths and prospects in academic data science: Report of the Moore-Sloan data science environments survey. *SocArXiv*. 2018; <https://doi.org/10.17605/OSF.IO/XE823>
8. Goodman, A. et al. Ten simple rules for the care and feeding of scientific data. *PLoS Computational Biology* 10, 4 (2014); <https://doi.org/10.1371/journal.pcbi.1003542>
9. Jackson, S.J. et al. Collaborative rhythm: Temporal dissonance and alignment in collaborative scientific work. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work (CSCW '11)* (2011), 245–254; <https://doi.org/10.1145/1958824.1958861>
10. Latour, B. and Woolgar, S. *Laboratory Life: The Construction of Scientific Facts*. Princeton University Press, Princeton, N.J., 1986.
11. Lee, C.P. Dourish, P. and Mark, G. The human infrastructure of cyberinfrastructure. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW '06)* (2006), 483–492; <https://doi.org/10.1145/1180875.1180950>
12. Levin, N. et al. How do scientists define openness? Exploring the relationship between open science policies and research practice. *Bulletin of Science, Technology & Society* 36, 2 (2016), 128–141; <https://doi.org/10.1177/0270467616668760>
13. National Academies of Sciences, Engineering, and Medicine. *Open Science by Design: Realizing a Vision for 21st Century Research*. The National Academies Press, Washington, D.C., 2018; <https://doi.org/10.17226/25116>
14. Jean-Christophe Plantin, J.-C. Data cleaners for pristine datasets: Visibility and invisibility of data processors in social science. *Science, Technology, & Human Values*, 2018; 0162243918781268. <https://doi.org/10.1177/0162243918781268>
15. Wallis, J.C., Rolando, E. and Borgman, C.L. If we share data, will anyone use them? Data sharing and reuse in the long tail of science and technology. *PLoS ONE* 8, 7 (2013); e67332. <https://doi.org/10.1371/journal.pone.0067332>

Michael J. Scroggins (mjscroggins@ucla.edu) is a postdoctoral fellow at the Department of Information Studies, UCLA, Los Angeles, CA, USA.

Irene V. Pasquetto (irenepasquetto@ucla.edu) is an assistant professor at School of Information, University of Michigan, Ann Arbor, MI, USA.

R. Stuart Geiger (stuart@stuartgeiger.com) is an assistant professor in the Department of Communication and Halicioglu Data Science Institute, UCSD, San Diego, CA, USA.

Bernadette M. Boscoe (boscoe@ucla.edu) is a postdoctoral fellow at iSchool, University of Washington, Seattle, WA, USA.

Peter T. Darch (ptdarch@illinois.edu) is an assistant professor at the School of Information Sciences, University of Illinois, Urbana-Champaign, IL, USA.

Charlotte Cabasse-Mazel (charlottecabasse@berkeley.edu) Executive Director, dhCenter UNIL-EPFL, University and Swiss Federal Institute of Technology, Lausanne, Switzerland.

Cheryl Thompson (cathompson@unc.edu) is a research data archivist at Odum Institute, University of North Carolina, Chapel Hill, NC, USA.

Milena S. Golshan (milenagolshan@ucla.edu) is a collection information specialist at the Los Angeles County Museum of Art, Los Angeles, CA, USA.

Christine L. Borgman (Christine.Borgman@ucla.edu) is Distinguished Research Professor at the Department of Information Studies, UCLA, Los Angeles, CA, USA.

This research was supported by grants to the University of California, Los Angeles, from the Alfred P. Sloan Foundation (#201514001, C.L. Borgman, PI) and grants to the University of California, Berkeley, from the Gordon and Betty Moore Foundation (Grant GBMF3834) and the Alfred P. Sloan Foundation (Grant 2013-10-27), as part of the Moore-Sloan Data Science Environments.

Copyright held by authors.

Publish Your Work Open Access With ACM!

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



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



Association for
Computing Machinery

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

**A discussion with Robert O’Callahan,
Kyle Huey, Devon O’Dell, and Terry Coatta.**

To Catch a Failure: The Record- and-Replay Approach to Debugging

WHEN WORK BEGAN at Mozilla on the record-and-replay debugging tool called rr, the goal was to produce a practical, cost-effective, resource-efficient means for capturing low-frequency nondeterministic test failures in the Firefox browser. Much of the engineering effort that followed was invested in making sure the tool could actually deliver on this promise with a minimum of overhead.

What was not anticipated, though, was that rr would come to be widely used outside of Mozilla—and not just for sleuthing out elusive failures, but also for regular debugging.

Developers Robert O’Callahan and Kyle Huey recount some of the more interesting challenges they faced in creating and extending rr, while also speculating about why it has experienced a surge in popularity more recently. O’Callahan, a Mozilla Distinguished Engineer, led the rr development effort. Huey also worked on rr while at Mozilla. Both have since left to create their own company, Pernosco, where their focus has turned to developing new debugging tools.

Helping to steer the discussion are two other engineers of note: Devon O’Dell, a senior systems engineer at Google who has never made a secret



of his keen interest in debugging, and Terry Coatta, the CTO of Marine Learning Systems.

DEVON O'DELL: What frustrations with other debuggers spurred you to start working on rr?

ROBERT O'CALLAHAN: The original motivation was that Mozilla was running lots of tests—as organizations do—and many of them failed nondeterministically, which made these things really difficult to debug. When you're running tests at large scale—say, thousands or millions of tests on each commit—low

failure rates like these can get to be pretty annoying.

You can just disable those tests, of course. But actually *fixing* them, especially since many correspond to underlying product bugs, seems far more appealing. So, we thought about how to do that and concluded that a tool able to record a test failure and then replay it as often as necessary would be just the thing—if we could build it, that is—since that would really lower the risk. Obviously, if a test needs to be run thousands of times to catch a failure, you would like for that to be automated. You would also want to be able to debug that failure *reliably*. That's why the basic

idea we started out with was: How can we record tests with the lowest overhead possible and then have the ability to replay the execution as often as needed?

TERRY COATTA: The simplistic solution here, of course, would be: “Hey, no problem. I just record *everything* this processor does.” So, just for clarity, what constitutes “low overhead” in the world of debuggers?

O'CALLAHAN: Good point. From the user's perspective, it just means the code doesn't appear to run any slower when you're recording than when you aren't. Achieving that, obviously, requires some extra work. I should add, for comparison's sake, that many tools in this

space would slow a program by a factor of 3, 5, 100—or even as much as 1,000—depending on the technology. Our goal was to get to less than a 2x slowdown.

It's certainly the case that the technology you pick is going to have a big influence on the amount of overhead. As you just mentioned, if you were to instrument so you could look at everything the CPU does, that obviously would require a lot of work.

O'DELL: What would you say it is about the rr approach that lets you achieve the execution speed you were looking for? How does that vary from other debuggers?

O'CALLAHAN: The fundamental idea is one that several systems share: If you can assume the CPU is deterministic and you're able to record the inputs to your system—say, to your virtual-machine guest or to some process—and you're also able to reproduce those inputs perfectly during replay, then the CPU will do the same thing. Hopefully, this means you can avoid actually having to monitor what the CPU does. That's the idea, anyway. And it isn't a new idea, either.


So, we record only what crosses the process boundary—for example, system calls and signals. Crossing the process boundary isn't so frequent since that will cost you time just by virtue of crossing a protection domain.

KYLE HUEY: We do record each injection of nondeterministic state. So, any time there is a syscall or some sort of asynchronous signal, we record that. But while we have to record all these things, we don't create any of our own.


O'CALLAHAN: So, here's the deal: Record-and-replay tends to be all or nothing. That means you basically need to catch all these different forms of nondeterminism. If you miss something, the behavior you will observe when you replay is probably going to diverge from what actually happened, which means you're going to end up in a very different state since programs are very chaotic. Which is to say, you really need to make sure you've nailed down all those sources of nondeterminism and have recorded every single one.

COATTA: What do you require from the hardware so as to collect all the necessary information?

O'CALLAHAN: First, there's that big assumption we have already talked



If the CPUs are deterministic and you execute one million instructions starting from some particular state while recording, you should end up in the same state in the replay.



about, which is that the CPUs are deterministic such that, whenever you run the same sequence of instructions from the same starting state, you ought to end up with the same results, the same control flow, and everything else. That's critical, and, obviously, there are some instructions that don't meet these requirements—something that generates random numbers, for example. That clearly would not be deterministic.

This means we need a way to trap any nondeterministic behaviors or instructions, or at least have some means for telling programs they should avoid these things. For example, with X86 architectures, there is a CPUID instruction you need to watch out for. So, for most modern CPUs and kernels, we did some work to ensure there is an API you can use to say, "Hey, on all CPUID instructions in this process, you need to trap and make sure they don't use RDRAND. Do your own thing instead." The same goes for hardware transactional memory, which is something modern Intel CPUs provide for.

The problem is, while the transactions themselves are OK, they can sometimes fail for spurious reasons. Which is to say, you might choose to run a transaction, and, behold, it works! It doesn't abort! But if you run it again, it's just possible the transaction *will* abort, owing to some internal cache state or something else in the environment you can't see or control. That effectively means we need to tell our programs to avoid using transactional memory.

Another thing we rely upon—and this is really important—is having some way to measure progress through a program. That is so we can deliver asynchronous events, like signals or context switches, at exactly the same point during replay as they came up during recording. If the CPUs are deterministic and you execute one million instructions starting from some particular state while recording, you should end up in the same state in the replay. But if you execute one million and three instructions in the replay, you're probably going to end up in some different state, and that would be a problem.

This is just to say that we need a way to count the number of instruc-

tions executed, ideally such that we can then, during replay, stop the program once that many instructions have been executed. This happens to be pretty much what hardware performance counters allow you to do, which means we really depend upon hardware performance counters to make this all work. And that, as it turns out, is probably also the key to getting `rr` to work with low overhead. Hardware performance counters are basically free to use—especially if you’re just counting, rather than interrupting your program to sample.

While it’s very fortunate for us that these counters are available, we also depend on them being absolutely reliable—and that’s the hard part. Basically, you need to make sure the count of events is the same each and every time you execute a particular instruction sequence. This also means, if your counter happens to be one that counts the number of instructions retired, it needs to report the same number of instructions retired as were actually executed. This can be quite a challenge since many people use performance counters just to measure performance, in which case this property isn’t really essential. If measuring performance is your use case and the counter is off by just a few instructions, it’s no big deal. But we, on the other hand, actually do care about this. The brutal truth is that a lot of these counters don’t measure exactly what they say they do. Instead, they deliver spurious overcounting or spurious undercounting.

That’s a big problem for us. So, we had to look for a reliable counter we could use with Intel devices, which were our initial targets. Ultimately, we found one that’s reliable enough—a conditional branch counter, actually. That is what we use now to count the number of retired conditional branches, and we have found it actually does exactly what the manufacturer says it does. And that’s really what makes `rr` possible. Had we found there were no counters that were accurate in this way, this project simply would not have been feasible.

COATTA: Does Intel *claim* its counters are accurate? Or does it offer no assurances at all?

O’CALLAHAN: That’s a difficult question to answer. What I can tell you is

that many of the cases where its counters deliver slightly erroneous values are documented as errata in the product datasheet. So, I believe there are people at Intel who care—perhaps not enough to fix the issue, but at least enough to divulge the problems.

COATTA: Are you concerned that the one precious performance counter you know you actually can rely upon might suddenly go nondeterministic on you?

O’CALLAHAN: Absolutely, and I’m hopeful that articles in leading computer science publications might serve to draw attention to this very concern.

O’DELL: You earlier mentioned the assumption of determinism you make in light of the obvious *lack* of determinism you find with execution orderings in multithreaded programs. It’s my understanding `rr` runs everything in a single thread more or less to counter this, but I imagine there’s probably much more to it than that.

O’CALLAHAN: To be clear, I should point out that `rr` runs everything on a single core. That is very important. We context-switch all those threads onto a single core. By doing so, we avoid data races since threads aren’t able to touch shared memory concurrently. We control scheduling that way and so are able to record the scheduling decisions, using the performance counter to measure progress. In this way, context switching becomes deterministic during replay, and we no longer have to worry about data races.

O’DELL: Does this also help you with issues such as memory reordering?

O’CALLAHAN: Yes. If we are talking about weak memory models, you ensure everything will be sequentially consistent by forcing all of it onto a single core. There is, therefore, a class of memory-ordering bugs that cannot be observed under `rr`.

COATTA: You’ve mentioned that you require the hardware to be capable of deterministically measuring progress in certain ways. But it also sounds as though there is a whole raft of constraints you face here. For example, it sounds like you need to be able to interface with the scheduler in order to understand or map all the threads down to a single core.

O’CALLAHAN: We don’t need to integrate with the operating-system scheduler. Actually, we use the `ptrace`

API to accomplish high-level control of what’s going on here. That’s a complicated API with a number of different features, but one of the things you can do with it is to start and stop individual threads. We use it to stop all the threads and then start just the ones we want to run. After that, whenever our scheduler decides, “Hey, it’s time for a context switch,” we interrupt the running threads as necessary and start a different thread. Then we pick another thread and run that using our scheduler. Essentially, the operating-system scheduler isn’t left with anything to do. We control everything, and that’s good since integrating with the operating-system scheduler is kind of a pain.

Just the same, there’s still at least some interaction with the operating-system scheduler in the sense that, whenever a program goes into a system call and the system call blocks, we need to be able to switch threads in our scheduler. That means basically setting aside the thread that was running to wait for an exit system call. In the meantime, we can also start running a different thread. We need an operating-system interface that will tell us when a thread has blocked and basically has been descheduled in the kernel.

Fortunately, there is just such an interface out there. Linux gives us these scheduling events—performance events, actually. There is a genuine Linux performance counter API that provides us with a way to be notified whenever the thread we are running blocks and triggers one of these deschedules. We use that for our event notification.

COATTA: It sounds like you have got this kind of super-detailed integration that leaves you dependent on certain very specific features of the processor. I suppose you rely on some interesting pieces of the operating system as well. As you have already suggested, this can lead to revelations of some behaviors people aren’t necessarily expecting. Which leads me to wonder if any interesting episodes have come up around some of these revelations.

O’CALLAHAN: Because of the requirement that the performance counters be perfectly accurate, some kernel changes unexpectedly hurt us. In one recent instance, there was a change

that landed in the kernel and then ended up briefly freezing the performance counters until a kernel interrupt could be entered, and that led to a few events being lost.

We seem to be the only ones who really care about this sort of thing. Anyway, we ended up finding that bug, and, hopefully, we will get it fixed before the kernel is released. That sort of thing tends to happen to us quite a bit.

HUEY: In the same vein, we found a bug in the Xen hypervisor related to a workaround to a bug in the Intel silicon that goes so far back into the dark days that nobody seems to be quite sure which CPU originally introduced it. What happened there was, if something ended up in the performance-monitoring unit's interrupt handler and was counted as zero, it automatically got reset to "1" since, apparently, if you left it at zero, it would just trigger another interrupt. And that, as you might imagine, is enough nondeterminism to break *rr* since it adds one event to the count. It took us something like a year to find that, which proved to be just no end of fun. And then they never did actually get it fixed.

O'CALLAHAN: Another aspect of this is that the sheer breadth of complexity with the X86 architectures is pretty staggering. The more instructions there are, the more ways there are to get things wrong. For example, there are a number of special instructions for accessing certain weird registers so you can ensure they have the right values during record-and-replay. And, trust me, this can prove to be very nontrivial. One register lets you know whether your process uses one of the X87 floating-point instructions from the 1980s. For our record-and-replay purposes, we need the value in that register to remain constant. If it also happens to be correct, so much the better. There are lots of weird instructions like that, which can be really eye-opening—and kind of scary.

HUEY: There definitely is a lot of complexity in the X86 space submerged just below the surface.

COATTA: As I listen to you talk about this, I'm thinking to myself, "Oh my God, this sounds like software-developer hell." You've got some super-complicated user software that relies on a whole

bunch of other super-complicated submerged bits of software. Does this ever get to you? Have you had to invest super-human amounts of effort into testing? Basically, how do you manage to cope with so much complexity?

HUEY: Well, we used to work on web browsers, so this just seems simple by comparison.

O'CALLAHAN: It's true that the stuff we're interfacing with now is kind of crazy and low level. On the other hand, it doesn't change all that rapidly. It's not as if Intel's architecture revs are coming at some incredible pace—especially not now. The kernel is also evolving pretty slowly at this point.

Because *rr* is purely a user-space tool (which was one of our design goals from the start), we depend on the kernel behavior of a bunch of APIs, but we don't really care about how those APIs are implemented so long as they don't break stuff. The kernel is also open source, so we can always see exactly what's going on there. The hardware, of course, tends to be much more opaque, but it's also more fixed.

Still, I have to admit it takes a lot of effort just to make sure things work consistently across architecture revs. It's also scary to depend on so many things, sometimes in some very subtle ways, and have absolutely no control over those things. But we actually invested a fair amount of energy in talking to people at Intel to nudge them in the right direction or at least discourage them from doing things that are sure to break stuff. For the most part, they have actually been quite helpful.

HUEY: I would add that we have done some amount of testing on the hardware. Of course, what we can do is fairly limited since, by the time we get access to any new Intel silicon, we are already essentially committed to it. Thankfully, there hasn't been a new microarchitecture released in a while, so that hasn't been a huge deal.

We also periodically run our regression test suite against the current kernel version, and we end up finding things there on a semi-regular basis—maybe once a quarter. Usually it's something relatively benign, but every now and then we'll find something more troubling: Maybe they've just bro-

ken the performance counter interface in a way that is problematic for us.

A recurring theme with *rr* is that one capability just seems to beget another. As an example, once the ability of *rr* to handle nondeterminism had been clearly demonstrated, it occurred to O'Callahan and Huey that even more nondeterminism might be employed to discover weird edge cases. In this way, *rr* has come to be used as not only a debugger, but also a tool for improving software reliability.

That is, by combining record-and-replay with the ability to harness nondeterminism, it became apparent that a program could be tested just by unleashing a torrent of nondeterminism on it to see what might come up as a consequence. Any problems bobbing to the surface then could be dealt with proactively, instead of waiting for failures to show up in production.

COATTA: You indicated the original impetus for creating *rr* was to capture hard-to-reproduce errors that surface during automated testing. But now, is it also being used in some other ways?

O'CALLAHAN: Actually, the funny thing is, while we designed *rr* to capture test and automation, it's *mostly* being used now for other things—primarily for ordinary debugging, in fact. On top of record-and-replay, you can simulate reverse execution by taking checkpoints of the program as it executes forward, and then rolling back to a previous checkpoint and executing forward to any desired state. When you combine that with hardware data watchpoints, you can see where some state in your program isn't correct and then roll back to look at the code responsible for that. Especially with C and C++ code, I think that almost amounts to a programming superpower.

Once people discovered that, they started using it and found there were some other benefits. That's when *rr* really started catching on. At this point, there's a fair number of people, both at Mozilla and elsewhere, who have taken to using it for basically all their debugging.


One of the more appealing benefits—of both rr and record-and-replay in general—is the clean separation you get between the act of reproducing a bug and the act of debugging it. You run your program, you record it, and—in doing so—you reproduce the bug. Then you can just continue supplying fresh input and debugging what comes just as often as you like. That could lead to a session that lasts a day, or even weeks—if you are up for that. But, with each pass, you will learn a bit more about what happened in that test case until you have finally got the bug figured out. That means you end up with a workflow that looks a little different from the traditional debugging workflow. People take to that, I think, since it's instinctive.

COATTA: Isn't there also another mode that gives you complete control over thread scheduling? How has that been employed as part of your debugging efforts?


HUEY: Assuming you care about the reliability of your software, once you have a useful tool for dealing with non-determinism, one relatively obvious thing to do is to throw as much non-determinism at it as you can. Basically, that means searching the execution space to find weird edge cases, which is just what Rob has attempted to do with chaos mode.

O'CALLAHAN: Right. We have already talked about how rr operates on a single core and the sorts of things that can affect program execution there. One issue is that there are certain kinds of bugs we've had trouble reproducing. In fact, we had some Mozilla Firefox test failures that would show up only intermittently—and, in some cases, only on certain platforms. To get to the bottom of that, we had sometimes run as many as 100,000 test iterations and still not be managed to reproduce a failure. That's when we started to think we probably needed to do another pass where we injected some more nondeterminism. The most obvious way to go about that was to randomize the scheduler, or at least randomize the decisions it was making.

That allowed us to study what sorts of schedules, for example, would be required to reproduce some particular bug. It also let us explore why rr wasn't producing those schedules. With a number of iterations like this, we ac-



One of the more appealing benefits—both of rr and record-and-replay in general—is the clean separation you get between the act of reproducing a bug and the act of debugging it.



tually managed to implement an improved form of chaos mode.

While we value these benefits of running in chaos mode, the overhead is high. We tried to limit that but couldn't eliminate all of it, so we decided to make chaos mode purely an option. Still, by turning on chaos mode to run your tests, you're likely to find some of your more interesting failures.

Anecdotally, a lot of people have reported they've found chaos mode to be useful for reproducing bugs that generally are very hard to reproduce. Something else we've discovered about using chaos mode that I consider to be particularly interesting is that many Mozilla tests that fail intermittently actually turn out to fail on only one platform—say, either Android or macOS. Yet when you debug them, you find it's actually a cross-platform bug that shows up on only one platform since it's either particularly slow or has some type of thread scheduler that makes it possible for the bug to be reproduced there. I should add that rr chaos mode often makes it possible to take a bug that was failing only on Android and then reproduce it on desktop Linux. This turns out to be rather useful.

COATTA: Why not always run things in chaos mode in the first place and, thus, surface failures more readily?

O'CALLAHAN: Chaos mode is clearly recommended if you're trying to reproduce an intermittent failure. But pairing chaos mode with the record-and-replay tool is also advisable. I mean, if you stripped out all the record-and-replay stuff, you would still be able to reproduce failures with some kind of controlled scheduling. But then what would you do with them?

Work to add new functionality to rr itself now seems to have concluded. But the story of renewal and extension will carry on with efforts to build new tools on top of the existing record-and-replay technology to deliver a “new debugging experience” in which developers can move beyond examining one state in one program space at a time to representing all of the program space within a database such that it can then be queried by a reworked debugger to

obtain information across time. A project to build this new debugger is already underway.

COATTA: Looking back over your effort to build *rr*, is there anything you would do differently?

O'CALLAHAN: Yes. But I should first say I think we generally made good design decisions, as most of our bets seem to have paid off pretty well. That includes our choice to focus on a single-core approach to replay even though that decision has been criticized, given that it essentially locks *rr* out of a large and growing class of highly parallel applications. The main problem is that we still don't know how to even *handle* those applications all that well. I don't think anybody has a clue when it comes to recording highly parallel applications with data races with low overhead using existing hardware and software.

By keeping our focus mostly on single-core applications, we have managed to do a pretty good job with those. It's important to have a tool that works well for a large set of users, even if it doesn't work as well for some other set of users. It's better to narrow your focus and be really good at something than to make the compromises a broader focus might require. I have no regrets about that.

I'm also fine with the decision not to do program-code instrumentation. I continue to be grateful for that every single time Intel releases a thousand new instructions and we don't have to care since we're not faced with having to add all of those to our instrumentation engine.

But, since you asked about what we'd do differently, I suppose we would probably write *rr* in Rust rather than C++ if we were starting today. Beyond that, though, I'm pretty satisfied with the core design decisions we have made. Kyle, do you see this differently?

HUEY: Not really. The things I'd want to do differently would be possible only in a different universe where saner hardware and saner kernels can be found.

COATTA: What do you see happening with *rr* as you move forward?

HUEY: We would like to add support for AMD and ARM architectures, but that's going to require silicon improvements from both of them.

O'CALLAHAN: We also need to improve our support for GPUs since *rr* doesn't work when you have sessions that directly access the GPU. To fix that, we'll need to get to where we better understand the interactions between CPU user space and GPU hardware so we can figure out how to get *rr* to record and replay across that boundary—especially when it comes to recording any GPU hardware effects on CPU user space—if that's even possible.

I think there's room to build alternative implementations of record-and-replay using different approaches, but as for *rr* itself, it's basically there already. I just don't see adding a large number of new features. Instead, I think the future will involve building on top of the record-and-replay technology, which is something we're already exploring. The basic idea is that, once you can record and replay an execution, you have access to all program states. Traditional debuggers don't really leverage that because they're limited to looking at one state at a time.

The future lies with this new idea called *omniscient debugging* that lets you represent all program space in a database and then rework your debugger such that it can make queries to obtain information across time. In theory, developers ought to be able to use this to obtain results instantaneously. That's where the next frontier lies in terms of improving the user experience.

A debugger like *rr* actually is an important stepping stone in this direction since what you really want is the ability to record some test failure with low overhead and then parallelize the analysis by farming it out to many different machines and combining the results. The effect of this would be essentially to deliver a precomputed analysis that enables a faster, more satisfying debugging experience for the developer.

COATTA: A new debugging user experience? Apart from delivering the results faster, what exactly do you have in mind?

O'CALLAHAN: One of the things you typically do with traditional debuggers is single-stepping, right? Basically, you want to trace out the control flow in functions. So, you step, step, step, step, step. You find yourself staring at a very narrow window that lets you look at the current state, which you then can manipulate through time as you try to

build a picture of the control flow. This is something you have to do by manually aggregating the data points.


Instead, what we're looking for is something that lets us observe a function execution and then—assuming we've already recorded and stored the control flow for the entire program—look at a function and immediately see which lines of the function have been executed. Which is to say, you should be able to view the control flow in an intuitive way. This, of course, will require a lot more debugger implementation.

O'DELL: It sounds like work on this might already be underway.

O'CALLAHAN: Yes, we have implemented a lot of this already. It's not out there yet, but we're working on it.

O'DELL: On a related UX [user experience] note, do you foresee some way to make it easier for people to map mental models of protocols to code?

O'CALLAHAN: Much could be done to present dynamic program execution to developers in a more intuitive way. For example, one thing we're doing with our new product is to make it much easier to explore the dynamic execution tree. That way, you can look at a function invocation and request a full list of the functions that are dynamically called from that. This maps pretty nicely to the mental models people have for programs and the ways they work. It also provides a great way to explore a program.

A lot more could be done here. If we were to build information about functions such as *malloc* and *free* into our debugger, that would also prove really powerful in terms of helping people understand exactly what their programs are doing. I think that would be pretty exciting. 

Related articles on queue.acm.org

Debugging in an Asynchronous World

Michael Donat

<https://queue.acm.org/detail.cfm?id=945134>

Advances and Challenges in Log Analysis

Adam Oliner, Archana Ganapathi,
and Wei Xu

<https://queue.acm.org/detail.cfm?id=2082137>

Reveling in Constraints

Bruce Johnson

<https://queue.acm.org/detail.cfm?id=1572457>

Copyright © 2020 held by owner/author.
Publication rights licensed to ACM.

Reducing datacenter carbon footprints.

BY JESSIE FRAZELLE

Power to the People

WHEN YOU UPLOAD photos to Instagram, back up your phone to the cloud, send email through Gmail, or save a document in a storage application like Dropbox or Google Drive, your data is being saved in a datacenter. These datacenters are airplane-hangar-sized warehouses, packed to the brim with

racks of servers and cooling mechanisms. Depending on the application you are using, you are likely hitting one of the datacenters operated by Facebook, Google, Amazon, or Microsoft. Aside from those major players, which I refer to as *hyperscalers*, many other companies run their own datacenters or rent space from a colocation center to house their server racks.

Carbon footprints. Most of the hyperscalers have made massive strides toward achieving carbon-neutral footprints for their datacenters. Google, Amazon, and Microsoft have pledged to decarbonize completely; however, none has yet succeeded in that quest.

If a company claims to be carbon neutral, this usually means it is offsetting its use of fossil fuels with renewable energy credits (RECs). A REC represents one MWh (megawatt-hour) of electricity that is generated and delivered to the electrical grid from a renewable energy resource such as solar or wind power. By purchasing RECs, carbon-neutral companies are essentially giving back

clean energy to prevent someone else from emitting carbon. Most companies become carbon neutral by *investing* in offsets that primarily avoid emissions, such as paying people not to cut down trees or buying RECs. These offsets do not actually remove the carbon the companies are emitting.

A *net zero* company actually must remove as much carbon as it emits. Though the company is still creating carbon emissions, those emissions are equal to the amount of carbon the company removes.

If a company calls itself *carbon negative*, it is removing more carbon than it emits each year. This should be the gold standard for how companies operate. None of the FAANG (Facebook, Apple, Amazon, Netflix, and Google) today claim to be carbon negative, but Microsoft issued a press release stating it would be by 2030.

Power usage efficiency, or PUE, is defined as the total energy required to power a datacenter (including lights and cooling) divided by the energy used for servers. A perfect PUE would be 1.0,

since 100% of electricity consumption would be used on computation. Conventional datacenters have a PUE of about 2.0, while hyperscalers have gotten theirs down to about 1.2. According to a 2019 study from the Uptime Institute, which surveyed 1,600 datacenters, the average PUE was 1.67.

PUE as a method of measurement is a point of contention. PUE does not account for location, which means a datacenter that is located in a part of the world that can benefit from free cooling from outside air will have a lower PUE than one in a very hot climate. PUE should be measured as an annual average since seasons change and affect the cooling needs of a datacenter over the course of a year. According to a study from the University of Leeds, “comparing a PUE value of datacenters is somewhat meaningless unless it is known whether it is operating at full capacity or not.”

Google claims an average yearly PUE of 1.1 for all its datacenters, while individually some are as low as 1.08. One of the actions Google has taken for lowering its PUE is using machine learning to cool datacenters with inputs from local weather and other factors—for example, if the weather outside is cool enough the datacenter can use it without modification as free cold air. It can also predict windfarm output up to 36 hours in advance. Google took all the data it had from sensors in its facilities monitoring temperature, power, pressure, and other resources to create neural networks to predict future PUE, temperature, and pressure in its datacenters.

This way Google can automate and recommend actions for keeping its datacenters operating efficiently from the predictions. Google also sets the temperature of its datacenters to 80°F, rather than the usual 68°F–70°F, saving a lot of power for cooling. Weather local to the datacenter is a huge factor. For example, Google’s Singapore datacenter has the highest PUE and is the least efficient of its sites because Singapore is hot and humid year-round.

Wired conducted an analysis of how Google, Microsoft, and Amazon stack up when comparing the carbon footprints of their datacenters. Google claims to be net zero for carbon emissions and publishes a transparency report of its PUE every year. While Microsoft claims it will be carbon negative by 2030, it is still carbon neutral today. It also claims to be pursuing 100% renewable energy by 2025.

Amazon, on the other hand, has the worst carbon footprint of the large tech companies. As noted previously, the location of the datacenter matters, so some Amazon regions might be greener than others because of the weather conditions in those areas or having more access to solar or wind energy. Amazon founder and CEO Jeff Bezos has pledged to get to net zero by 2040. Greenpeace seems to believe otherwise, claiming in a 2019 report that Amazon is not dedicated to that pledge since its Virginia datacenters were at only 12% renewable energy.

In 2018, Apple claimed 100% of its energy was from renewable sources. Facebook claims it will be at 100% renewable energy by the end of 2020.

While U.S. companies have followed suit on pledging to lower their carbon footprints, Chinese Internet giants such as Baidu, Tencent, and Alibaba have not.

What is Using Power in a Datacenter?

According to a study from Procedia Environmental Sciences, 48% of the power in a datacenter goes to equipment such as servers and racks, 33% to HVAC (heating, ventilation, and air conditioning), 8% to UPS (uninterrupted power supply) losses, 3% to lighting, and 10% to everything else.

HVAC requires a delicate process of making sure hot air from server exhaust does not mix with cool air and raise the temperature of the entire datacenter. This is why most datacenters have hot and cold aisles. The goal is to have the cold air flow into one side of the racks, while the hot air exhaust comes out the other side. Optimizing air flow throughout the racks and servers is essential for HVAC efficiency.

Power comes off the grid as AC power. This can be single-phase, which has two wires (a power wire and a neutral wire); or three-phase, which has three wires, each 120 electrical degrees out of phase with each other. The key difference between the two is that three-phase power can handle higher loads than single-phase. The frequency of the power off the grid can be either 50Hz or 60Hz. Voltage is any of the following: 208V, 240V, 277V, 400V, 415V, 480V, or 600V.

Since most equipment in a datacenter uses DC power, the AC power needs to be converted. This results in power losses and wasted energy adding up to around 21%–27%. To break this down, there is a 2% loss when utility medium voltage, defined as greater than 1000V and less than 100 kV, is transformed to 480VAC; a 6%–12% loss within a centralized UPS because of conversions from AC to DC and DC back to AC; and a 3% power loss at the PDU (power distribution unit) level resulting from the transformation from 480VAC to 208VAC. Standard power supplies for servers convert 208VAC to the required DC voltage, resulting in a 10% loss, assuming the power supply is 90% efficient. This is all to say that power is wasted through-



Google’s datacenter in Eemshaven, Netherlands, has been powered entirely by renewable energy since the day it opened in late 2016.


out traditional datacenters in transformations and conversions.

In an attempt to lessen the amount of wasted power from conversions, some people rely on high-voltage DC power distribution. The Lawrence Berkeley National Lab conducted a study in 2008 in which the use of 380VDC power distribution for a facility was compared with a traditional 480VAC power-distribution system. The results showed the facility using DC power eliminated multiple conversion stages resulting in a 7% decrease in energy consumption compared with a typical facility with AC power distribution. This is rarely done at hyperscale, however. Hyperscalers tend to have three-phase AC going to the rack, then convert to DC at the rack or server level.


More Power-Efficient Compute

In addition to RECs and using 100% renewable energy, there are other ways hyperscalers have made their datacenters more power efficient. In 2011, the Open Compute Project started out of a basement lab in Facebook's Palo Alto headquarters. Its mission was to design from a clean slate the most efficient and economical way to run compute at scale. This led to using a 480VAC electrical distribution system to reduce energy loss, removing anything in the servers that didn't contribute to efficiency, reusing hot aisle air in winter to heat the offices and the outside air flowing into the datacenter, and removing the need for a central power supply. The Facebook team installed the newly designed servers in the Prineville datacenter, which resulted in 38% less energy to do the same work as the existing datacenters. It also cost 24% less.

Let's dive into some of the details of the Open Compute designs that allow for power efficiency. The Open Rack design includes a power-bus bar with either 12VDC or 48VDC of distributed power to the nodes. The bus bar runs along the back of the rack vertically. It transmits power from the rack-level PSUs (power supply units) to the servers in the rack. The bus bar allows the servers to plug in directly to the rack for power, so when you service an Open Rack you do not need to unplug power cords; you can just pull



One of the actions Google has taken for lowering its PUE is using machine learning to cool datacenters with inputs from local weather and other factors.



the server out from the front of the rack. With the Open Compute designs, network connections to servers are at the front of the rack so the technician never has to go to the back of the rack (that is, the hot aisle).

Redundancy. Conventional designs have PSUs in every server. The Open Rack design has centralized PSUs for the rack, which allow for N+M redundancy, the most common deployment being N+1. This means there is an extra PSU per rack of servers. In a conventional system this would be 1+1 since there is one extra PSU in every individual server. Keeping the PSUs centralized to the rack reduces the number of power-converting components; this increases the efficiency of the system.

Right-sized PSUs. Server designers tend to choose PSUs that have enough headroom to deliver power for the maximum configuration. Server vendors would rather carry a small number of oversized power-supply SKUs than a large number that are right-sized to purpose, since economies of scale prefer the former. This leads to an oversizing factor of at least two to three times the required capacity for conventional power supplies. In comparison, a rack-level PSU will be less oversized since it is right-sized for purpose. The hyperscalers also have the advantage of economies of scale for their hardware. The typical Open Rack-compliant power supply is oversized at only 1.2 times the required capacity, if that.

Optimal efficiency. Every power supply has a sweet spot for load versus efficiency. The 80 Plus certification program measures PSU efficiency using these different grades: bronze, silver, gold, platinum, and titanium. The most power-efficient grade is titanium. The most common grade of PSU used in datacenters is silver, which has a maximum efficiency of 88%, meaning it wastes 12% electric energy as heat at the various load levels. In comparison, the 12V and 48VDC PSUs have data showing maximum efficiencies at 95% and 98%, respectively. This means the rack-level PSUs waste only between 5% and 2% of energy.

While the efficiency of the rack-level PSU is important, you still need to weigh the cost of the number of conversions being made to get the power to each server. For every unnecessary

power conversion, you are paying an efficiency cost. For example, with a 48VDC rack-level power supply, the server might need to convert the rack provided from 48VDC to 12VDC, then that 12VDC to VCORE. VCORE is the voltage supplied to the CPU, GPU, or other processing core. With its 48VDC power supply, Google advocates for using 48V to PoL (point of load) to deliver power to the servers. This means placing a DC-to-DC or linear power-supply regulator going from the rack-level PSU to the server, which would reduce the number of conversions needed to get the power to the processing cores. The 48VDC-to-DC regulators required for Google's implementation, however, are not common and come at a premium cost. It is likely that Google's motivation for opening the specs for the 48VDC rack is to drive more volume to those parts and thus drive down costs. In contrast, 12VDC-to-DC regulators are quite common and low cost.

Reading a Power-Efficiency Graph

The accompanying figure is an example of a power-efficiency graph for a power supply. You can see that the peak of the graph is where the PSU is the most efficient. Divide the output power by the input power to calculate efficiency. The x-axis of the graph measures the load of the power supply in watts, while the y-axis measures efficiency.

If you know the peak load is 120W and idle is 60W, as shown in the figure, then this power supply would be more than is needed since it can handle up to 150W. At a peak load of 120W with 230VAC, this power supply would have a maximum efficiency of around 94% and a minimum efficiency at idle of around 92% with 230VAC. You now know the losses of this specific power supply and can compare it with other supplies to see if they are more efficient. This allows you to choose the right power supply for the load.

Open Compute servers without a bus bar. Not all Open Compute servers include a power-bus bar. Microsoft's Olympus servers require AC power. The Olympus power supply has three 340W power-supply modules, one for each phase, with a total maximum output of 1,000W. Therefore, these power supplies assume all deployments are three-phase power. The minimum effi-

With the Open Compute designs, network connections to servers are at the front of the rack so the technician never has to go to the back of the rack (that is, the hot aisle).

ciency of the PSU is 89%–94%, depending on the load, placing the grade of the Olympus power supply around an 80 Plus platinum.

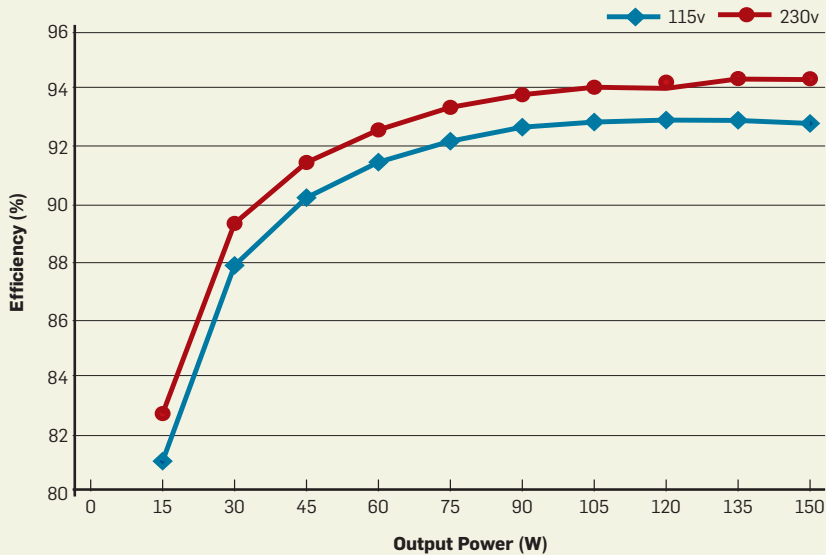
Like all technical decisions, using per-server AC power supplies versus rack-level DC is a trade-off. By having separate power supplies, different workloads can balance the power they are consuming individually rather than at a rack level. In turn, though, Microsoft needs to build and manufacture multiple power supplies to ensure they are right sized to run at maximum efficiency for each server configuration. Serviceability also requires technicians to unplug power cables and go to the back of the rack.

At the time Microsoft made the decision to use individual AC power supplies per server, the Open Rack design was at v1 (not v2 like it is today), the cost of the copper for the power-bus bar was higher, and the loss of efficiency to resistance was a factor. The Open Rack v1 design had an efficiency concern with power loss resulting from heating the copper in the bus bar. If a rack holds 24 kW of equipment, a 12VDC power-bus bar must deliver 2kA of current. This requires a very thick piece of copper, which has a significant power loss because of resistance in the bus bar.

Let's break down how to measure the relationship of power to resistance. Ohm's law declares electric current (I) is proportional to voltage (V) and inversely proportional to resistance (R), so $V=IR$. To see the relationship of power to resistance, combine Ohm's law ($V=IR$) with $P=IV$, which translates to power (P) being the product of current (I) and voltage (V). Substituting $I=V/R$ gives $P=(V/R)V=V^2/R$. Then, substituting $V=IR$ gives $P=I(IR)=I^2R$. So, $P=I^2R$ is how you can calculate the power loss resulting from resistance in the bus bar.

In making its decision, Microsoft balanced the conversion efficiency against the material cost of the bus bar and the resistive loss. Open Rack v2, however, changes the tradeoffs of the original decision. With a 48VDC bus bar, a rack that holds 24kW of equipment requires only 500A, as opposed to the 2kA required by the 12VDC power-bus bar from the v1 spec. This translates into a much cheaper bus bar and lower losses from resistance. The bus

Power efficiency graph example.



bar still has more loss than 208VAC cables, but there is an improved efficiency from the power-supply unit at the rack level, which makes it compelling. As stated earlier, however, you need to be mindful of the number of conversions needed to get the power to the components on the motherboard. If your existing equipment is 12VDC, you would want to avoid any extra conversions using that with a 48VDC bus bar. Save the 48VDC bus bar for new equipment that has 48V to PoL to avoid extra conversions.

The main difference between Microsoft's design with individual power supplies and the 24VDC and 48VDC Open Rack designs is the way the initial power is delivered to the servers. Microsoft's design distributes three-phase power to the servers individually through power supplies, while the 24VDC and 48VDC power-bus bars distribute the power delivery to the servers. Once power is delivered to the server, it is typically sent through a DC-to-DC power-supply regulator, which in turn powers the components on the motherboard. This step is shared whether the power is coming from a single power-bus bar or individual power supplies.

Another interesting bit comes into play with UPSes. As noted earlier, there are losses in efficiency because of UPSes. What does this mean in terms of a DC bus bar or individual AC PSUs? When AC power is going into

each individual server, you have two choices: a UPS on the AC before it gets distributed to the individual servers, or a UPS per server integrated into each server's PSU. Deploying and servicing individual batteries per server is a nightmare for maintenance. Because of this, most facilities that use AC power to the servers wind up using rack-wide or building-wide UPSes. Since the batteries in a UPS are DC, an AC UPS has an AC-to-DC converter for charging the batteries and a DC-to-AC inverter to provide AC power from the battery. For online UPSes, meaning the battery is always connected, this requires two extra conversions from AC to DC, and DC back to AC, with power-efficiency losses for both.

With a DC rack-level design, battery packs can be attached directly to the bus bar. The rack-level PSUs are the first AC-to-DC conversion state so there is no need for another conversion since everything from there runs on DC. The downside is that the rack-level PSU needs to adjust the voltage level to act as a battery charger. This means the servers need to accept a fairly wide tolerance on the 48V target, around $\pm 10\%$, so 40-56V isn't unreasonable. Because DC-to-DC converters are fairly tolerant about input voltage ranges, this is fairly straightforward, without any significant loss in power efficiency. It's important to note that for hyperscalers UPSes are present only to allow for a

generator to kick in—a few seconds rather than 10–15 minutes for a traditional datacenter.

With commodity servers, such as Dell or Supermicro, the cost of individual power supplies is much higher in terms of power efficiency since those PSUs do not have as high an 80 Plus grade and do have much more oversizing. They also tend to lack power-supply regulators that minimize power-conversion losses in supplying power to the components on the board. This would lead to around an 8%–12% gain in power efficiency by moving from a bunch of commodity servers in a rack to an Open Compute Project design—not to mention that the serviceability ease of the bus bar would benefit technicians as well.

By designing rack-level architectures, huge improvements can be made for power efficiency over conventional servers, since PSUs will be less oversized, more consolidated, and redundant for the rack versus per server. While the hyperscalers have benefited from these gains in power efficiency, most of the industry is still waiting. The Open Compute Project was started as an effort to allow other companies running datacenters to benefit from the power efficiencies as well. If more organizations run rack-scale architectures in their datacenters, the wasted carbon emissions caused by conventional servers can be lessened.

Acknowledgments. Thanks to Rick Altherr, Amir Michael, Kenneth Finnegan, Arjen Roodselaar, and Scott Andreas for their help with the nuances in this article. 

Related articles on queue.acm.org

Cooling the Data Center

Andy Woods

<https://queue.acm.org/detail.cfm?id=1737963>

Virtualization: Blessing or Curse?

Evangelos Kotsovinos

<https://queue.acm.org/detail.cfm?id=1889916>

Words Fail Them

Stan Kelly-Bootle

<https://queue.acm.org/detail.cfm?id=1569209>

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

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

DOI:10.1145/3386526

A tool that helps journalists discover new story angles by offering insight not search results.

BY NEIL MAIDEN, KONSTANTINOS ZACHOS, AMANDA BROWN, DIMITRIS APOSTOLOU, BALDER HOLM, LARS NYRE, ALEKSANDER TONHEIM, AND AREND VAN DEN BELD

Digital Creativity Support for Original Journalism

JOURNALISM INVOLVES THE search for and critical analysis of information.¹⁸ How journalists discover and select sources of this information is important to avoid bias, to be credible and trusted, and to create angles with which to generate new stories of value to readers.

Journalist creative thinking, to discover and generate new associations during this search and analysis of information, contributes to the generation of new stories. Journalists are known to seek opportunities to develop new creative skills with which to discover information.¹⁷ Applying these skills enables journalists to maintain control over their work.²⁵ And emerging

forms of investigative journalism demand new creative search and association skills.¹⁰

However, discovering and examining information sources about complex stories takes time—time that journalists increasingly lack as news organizations reduce staff numbers.²² The digitalization of news production and consumption has led many news businesses to become uncompetitive. Some work practices are slow to change due to conflicts with journalist professional values for autonomy.⁸ Therefore, as coping strategies, journalists often use subsets of available and familiar information sources to create stories, which in turn can reduce the diversity of angles used to report stories.

Although journalism is one of the creative industries, explicit support for the creative skills of journalists is rare. For example, it is not one of the five journalist capabilities reported in Cohen et al.,⁴ and few digital tools support journalist creativity.

INJECT was a new digital tool designed to support journalists to discover new associations with which to generate stories with angles more novel and valuable than stories published previously. It integrated creative search algorithms with which to discover information in published news stories and interactive support to form new associations with this information during

>> key insights

- **Journalists identified more with digital tools to support them to discover and generate new angles on stories more quickly than now—tools that recognized and augmented their existing creativity skills.**
- **Different creative search algorithms applied to news information operationalized the strategies for discovering new angles on stories reported by experienced journalists.**
- **Evaluations of the INJECT digital tool in three newsrooms revealed it increased the novelty of stories written by journalists, but younger journalists more open to new technologies and working more autonomously were more likely to use the tool.**

INJECT's design was informed by established cognitive models of creative thinking.

Person

INJECT's creative news index is now an important asset of more than 500 million of news information for computational manipulation.

Thing



though journalism is one of the creative industries, explicit skills in the creative skills of journalists is rare.

INJECT

journalism × crisis × digitalization ×

Technology

Digital Humans on the Big Screen

Viewpoint

OMSCS: The Revolution Will Be Digitized

Contributed

Why Computing Belongs in the Social Sciences

Last Byte

Q&A with Elisa Bertino



Thing

INJECT SUGGESTS

INJECT

Other angles to explore

Think about new angles using one or more of these related topics:

- Generate associations
- Evaluate story angles
- Deliver digital editorial support

Can you use their backgrounds, histories, or data about them or evidence that others have reported?

Place



The crawler was directed to fetch verified news stories from 1,105 predefined publishers by 380 diverse titles in 6 languages.

Organization



creative thinking. It was designed to contribute to journalist engagement in professional-level creative work, that is, work that generated income and provided a living.¹² This work included writing stories that were creative, that is, judged to be novel and valuable¹⁴ through the application of the creativity skills¹ of the journalists.

Existing Creativity Support Technologies for Journalists

Digital tools that enhance the creativity skills of journalists are rare. One exception was the *Story Discovery Engine*, which used artificial intelligence algorithms for investigative reporting.³ The *Tell Me More* system mined the Web for similar stories reported by different sources and extracted text that offered new information in the form of quotes, actors and figures.¹¹ Both of these tools had similar objectives to INJECT, but were not framed as creativity support tools for journalists. The *SocialSensor* news app surfaced fast moving trends from social media content, but revealed biases arising from such content.²³ Many different data visualization tools support journalists to make sense of, for example, social media content.⁶ However, none supported human creative thinking to discover new angles on news stories.

To work around the lack of bespoke tools for information search and analysis tasks, many journalists use generic search tools such as Google,¹³ but these lack explicit support for human creative thinking about news stories.

Unlike in journalism, digital creativity support has been implemented for professionals in other creative industries, such as the performing arts, music, and film and television. Examples of the digital support include *StoryCrate*, a collaborative editing tool developed to drive users' creative workflows within a location-based television production environment² and *Trigger Shift*, which appropriated information technologies into performance art in the theater.²¹ Other domains for which digital creativity support has been developed include theatre, scientific discovery, and caring for older people.

Therefore, to fill the gap in journalism, new digital creativity support was developed that aligned with the work practices, tools and values of journalists. The resulting tool was called INJECT.

Designing INJECT with Journalists

INJECT's design was informed by established cognitive models of creative thinking. Most of these models describe dual processes of developing and evaluating ideas to generate outcomes that are both novel and valuable.^{12,14} Developing ideas is a divergent and associative process that can be spontaneous and deliberate, and involves retrieving relevant items from memory and generating associations with new information.⁹ By contrast, evaluating ideas is more analytic, but can be interleaved tightly with developing ideas.⁷

Therefore, INJECT was designed to support journalists to discover new in-

formation, generate associations between this information and items from memory to discover new angles on news stories, and evaluate these angles quickly during story development.

To align INJECT to these work practices, tools, and values, journalists were included in the tool's design. Interviews were held with experienced and inexperienced journalists to discover problems, requirements, and constraints. Paper-based then digital wireframes of the INJECT tool were developed and presented to professional journalists. New releases of the working INJECT software were prototyped for their usability and impact with professional and student journalists.

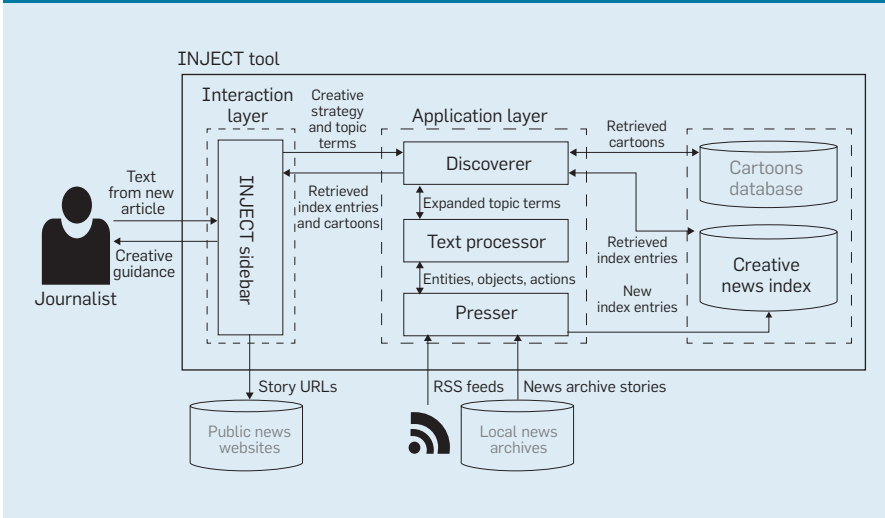
The user-centered design process uncovered three important values that most journalists held about their work—values the INJECT tool was designed to uphold.

The first value was the importance of discovering information already reported in verified newspapers, as opposed to in unverified sources, as the starting point for discovering new angles on stories. Even though it was argued that published news might constrain their creative thinking, most journalists expressed a preference for it to direct the discovery of new associations and angles. Feedback on prototypes revealed three specific types of verified news information were effective for discovering angles: 1) published news stories similar but not the same as the new story being written; 2) entities such as people, places, and organizations that might relate to these new stories, and; 3) guidance for directed creative thinking to develop the stories. INJECT was designed to direct journalists to generate new associations between information discovered in similar stories and in entities referenced in these stories.

The second value was to recognize the existing creativity skills of journalists. Many who engaged in INJECT's design initially rejected the need for digital support for their creative thinking. After all, journalism is one of the creative industries, and many chose it as a profession to be creative. Instead, journalists identified with the need to generate new angles more quickly.

The third value was that creative thinking was not separate from but part of everyday journalistic work. Indeed,

Figure 1. The INJECT tool's three-tier architecture, showing its layers, services and external information sources.



journalists sought support for more original journalism that was embedded in daily work tasks and tools, such as text editors.

The INJECT Tool

The INJECT tool was implemented with natural language processing, multi-language creative search, and interactive creativity support capabilities. It indexed content from millions of verified stories published by hundreds of news titles in multiple languages in order to provide journalists with a sufficiently large external information source from which to discover associations.

The INJECT tool's three-tier architecture is shown in Figure 1. The interaction layer was a sidebar designed to be simple, fit with existing work practices, and encourage journalists to discover new angles on stories quickly without learning new skills.

The application layer was composed of services designed to generate large numbers of possible associations between information that journalists were writing about using indexed news content from millions of already published news stories.

These services retrieved this content from INJECT's data layer, called the creative news index, which was designed so the discoverer service could undertake divergent creative searches that were more sophisticated than were possible with existing Web search and news site APIs. The index was populated by the presser service, which indexed millions of verified news stories as possible starting points for discovering new angles on stories. The text processor service was invoked by the presser to make sense of and to generate indexed content from published news, and by the discoverer to expand creative search queries.

INJECT's presser. The presser generated indexes of millions of verified news stories that could be retrieved, on request, as starting points for journalists to discover associations with which to generate new angles on stories. It had a crawler component that fetched news stories to index from open RSS feeds, and an importer component that fetched stories from accessible newspapers' archives.

The crawler was directed to fetch verified news stories from 1,105 pre-

defined RSS feeds published by 380 diverse news titles in six languages. These feeds, titles and languages were selected by INJECT's editorial team to generate indexes of diverse views and angles on news, and ranged from major daily newspapers in the U.S., regional newspapers in the Netherlands, and tabloid titles in the U.K. On a normal news day, it fetched about 15,000 verified stories. Stories from high-frequency feeds were fetched every 30 minutes, others every 12 hours. During each fetch cycle, the crawler automatically read all news stories accessible via the URLs in each RSS feed, removed navigation links, adverts and embedded media such as links, images, and videos, and sent the remaining text string, along with story's author, URL, image URL, and published date, to the text processor service. This text string, author, date, and URLs provided a rich external information source with which journalists could discover and generate new associations and angles on news stories.

The importer component was similar to the crawler but was directed to fetch stories from local JSON files. It was developed to address the need of news organizations to use their own stories as starting points for more original journalism. Like the crawler, it also sent a text string, along with author, URL, image URL, and published date, to the text processor service.

INJECT's text processor. The text processor service generated new entries to add to the creative news index by analyzing the natural language text string of each fetched news story with the following:

- ▶ Named entity extraction mechanisms to index stories using real names such as people and places. The mechanisms that treated candidate-named entities as groups of consecutive words describing a concept such as a person (for example, Tawakkol Karman), location (for example, Sana'a), organization (for example, United Nations) or object (for example, war crime). This enabled the processor to extract entities with which journalists might discover associations not described in the text, for example the entity Sana'a from the text the capital of Yemen. After experimentation with alternatives, the processor invoked the DBpedia Spotlight⁵ and Polyglot²⁰ services. Spotlight annotated

mentions of DBpedia resources using entity detection and disambiguation algorithms with adjustable precision and recall, which were used to refine INJECT's sensitivity to news content using measures such as entity prominence, topical pertinence, and disambiguation confidence. Polyglot implemented named entity extraction, speech tagging, sentiment analysis, morphological analysis, and transliteration in all of INJECT's six target languages—English, German, Dutch, French, Italian, and Norwegian. It could detect, for example, that Forente Nasjoner is Norwegian for the entity United Nations, the international organization founded in 1945;

- ▶ Automatic parser mechanisms that detected nouns and verbs to index stories using common objects and actions. The parsers split news text into sentences then applied part-of-speech tagging to mark up words as belonging to lexical, part-of-speech categories. Shallow parsing was applied to generate a machine understanding of the structure of a sentence without parsing it fully into a parsed tree form. The output was a division of the text's sentences into a series of words that, together, constituted a grammatical unit. To select candidate objects and actions from these units with which journalists might also discover associations, the mechanism applied lexical extraction heuristics on a syntax structure rule-tagged sentence. For example, the processor parsed the news headline *The Yemen war in the world's worst humanitarian crisis* to extract the nouns such as *war*, *world* and *crisis*.

INJECT's creative news index. For each fetched story, the creative news index generated a new entry composed of all extracted named entities, objects and actions and frequencies of occurrence, the author, URL, image URL, and publication date. A typical entry for a news story of 400 words was composed of between 30 and 50 entities, objects, and actions. Early prototyping of the INJECT tool revealed that indexes with this volume and type of content were sufficient to generate new associations that journalists reported could be effective for discovering new angles.

All index entries were uploaded to an external Elasticsearch cluster to be manipulated by the discoverer's cre-

ative search algorithms. Elasticsearch is a scalable open source search engine with a REST API that provides scalable, near real-time search. This performance was essential to support journalists to discover new angles on stories more quickly. In April 2020, the Elasticsearch cluster held over 17 million entries, with another 350,000 new entries being added each month.

INJECT's sidebar and discoverer. Journalists interacted with the INJECT sidebar to discover new associations and angles on stories. The sidebar was designed to provide journalists with index information and features and generate new associations with this information without opening another application. To work within the space constraints of the widget, the sidebar was implemented with mouse hoverboxes and information that journalists could use to discover associations quickly. Its design also supported journalists to flip quickly between ideation and evaluation processes during story development.

Figure 2 depicts use of the sidebar in the Google Docs editor to discover associations leading to angles for a new story about the Yemen humanitarian crisis. The sidebar was also implemented for Wordpress, Adobe InCopy text editors, Google Chrome Web browser, and content management systems that use the TinyMCE text editor, as well as a separate Web application that a journalist could reshape as the sidebar.

If the journalist highlighted text in

the editor, the sidebar invoked the text processor service to extract named entities (for example, Yemen), nouns and verbs (for example, crisis, ecological) as candidate topics to present at the top of the sidebar, see Figure 2. This feature was implemented to increase the sidebar's usability and enabled journalists to work more quickly.

The journalist could then use the icons beneath these topics to select between six predefined creative strategies that mimicked the strategies of experienced journalists.¹⁵ These strategies were implemented in the discoverer service to retrieve creative news index entries with the following:

- A. Quantified information associated with the topics;
- B. Information about people associated with the topics;
- C. Information about events associated with the background of the topics;
- D. Information about future consequences associated with the topics;
- E. Datasets and visualizations associated with the topics; and,
- F. Comical information associated with the topics.

For strategies A-E, the discoverer:

1. Disambiguated each noun topic term by discovering its correct sense in the online lexicon at WordNet using context knowledge from other terms in the query (for example, that *crisis is an unstable situation of extreme danger or difficulty* rather than a *crucial stage or turning point in the course of something*). It then expanded each term with other

terms with similar meanings (for example, the term *crisis* is synonymous with *exigency* and *flashpoint*) and included these terms in the search query. Term sense disambiguation and query expansion was implemented to retrieve index entries that were different lexically but related semantically to the topic terms, so that journalists could generate new associations based on different types of semantic similarity;

2. Invoked an Elasticsearch search via the news API with the expanded query terms and logic operators set by the journalist to control search breadth. Elasticsearch returned a set of indexed entries that achieved a threshold match score in response times acceptable to journalists;

3. Scored the returned index entries for relevance based on the frequencies of original and expanded query terms in the title of each story, to prioritize entries with headlines related to topic terms. This scoring mechanism was implemented to reflect the structure of most news stories with the most important information at the start of stories;

4. Filtered the scored index entries using constraints specified for the selected strategy, so that journalists were presented with information to form associations consistent with that strategy. For example, for quantified information (A), it filtered to retain entries with a minimum threshold of 100s of quantity, measure and value keywords, for example *Sterling*, *population* and actual numbers. For information about events associated with the background of the topic terms (C), it filtered to retain entries with more than 500 words of content and a minimum threshold of 100s of keywords indicative of background articles such as *cause*, *impact* and *studies* from sources such as the *Economist* and the *New York Times*. And for information about people (B), it generated orders of entries that reference a person entity named in a minimum number of entries.

The discoverer sent JSON representations of each remaining index entry to the sidebar to display as a news card. By contrast, for strategy F, discoverer generated simpler keyword queries that searched the caption text of over 60,000 political cartoons accessed by INJECT via an API from an external database.

Figure 2. The INJECT sidebar on the right side of the Google Docs text editor.

A journalist writing a new story about the Yemen crisis is presented with news articles and information about places, things, people and organizations with which to discover associations leading to new angles.



This automation of information discovery was designed to enable journalists to commit more cognitive resources to generating associating and evaluating ideas. The sidebar presented the retrieved information as a scrollable sequence of news cards. Journalists could select the information to view using sidebar features to sequence the news cards by relevance, date of publication or random, and to present news published within selected periods.

Each news card in the sidebar presented the title, publication, date, first sentence, and 10 randomly selected entities. Clicking on the title opened the original new story or cartoon, at source, in a new browser tab. Positioning the cursor over each rectangle presented a pop-up creativity spark generated for that places, things, people and organizations. This feature was implemented as a mouse hover-over to enable journalists to explore multiple sparks and discover different associations quickly. The sparks themselves were designed to direct the deliberate generation of associations and ideas by journalists. Each was generated by the sidebar from a predefined set of spark types to direct journalists to think about, for example, the history and relevance of places, the motives of people and their opponents, the future and emotional impact of objects, and available data about organizations.

Figure 3 shows these features in three different INJECT sidebars presented for different angles using the same information about the Yemen humanitarian crisis.

The sidebar also presented other styles of news card showing only entities, word clouds, and sparks in list form. These other styles, shown in Figure 4, were added to reduce comparisons with Google search that reduced journalists' expectations for creativity support.

Furthermore, to support journalists to evaluate as well as discover ideas, the sidebar launched Google Web searches in new browser tabs from within INJECT, to retrieve information with which to analyze and critique ideas, see Figure 5.

The INJECT tool was tested by journalists working in multiple languages. When sufficiently robust, it was evaluated in different newsrooms.

Figure 3. Three INJECT sidebars presented for the new story about the Yemen crisis, showing (from left-to-right) information to support a background angle, a people angle, and a comical angle based on political cartoons.

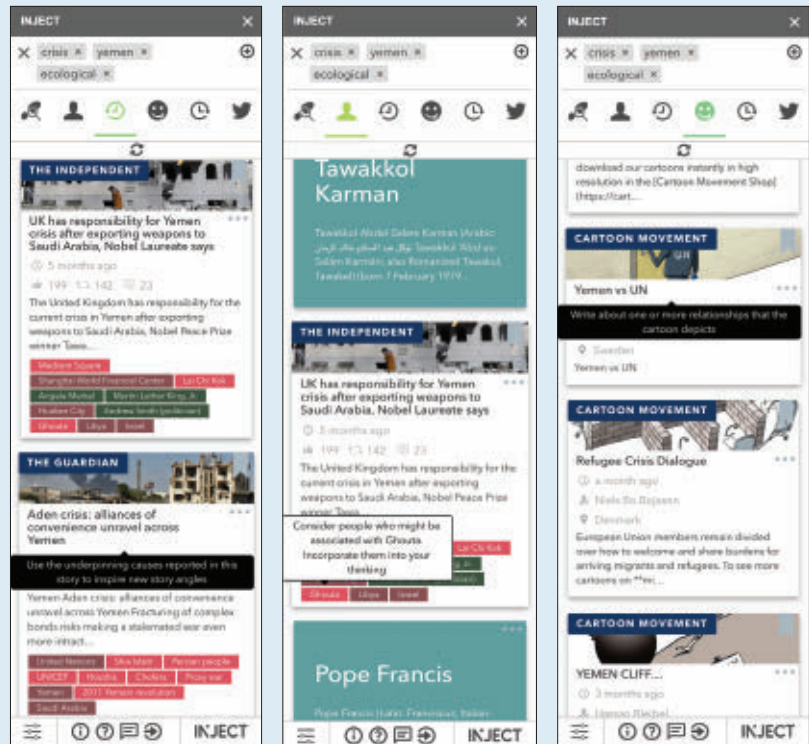
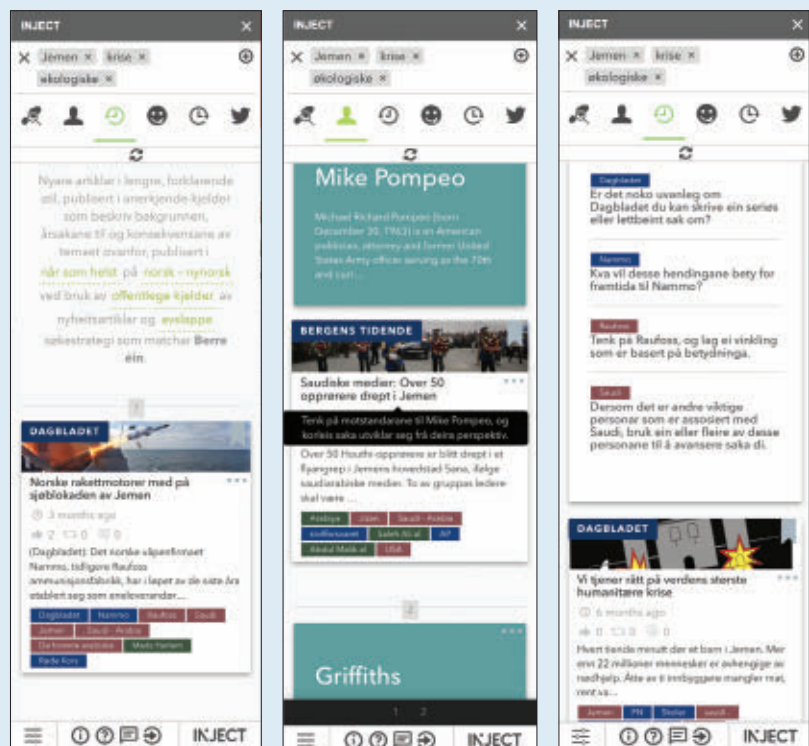


Figure 4. Three INJECT sidebars presented for the same new story about the Yemen crisis in Norwegian, showing (from left-to-right) use of a background information angle, a people angle, and use of creativity sparks in list form.



The Expert Judgment Process Used to Rate News Stories

Three of the seven judges were domain experts in journalism—associate professors of journalism at local higher education institutions. The other four had roles that equipped them with extensive local knowledge, as head of information at a regional institute in business and trade, two local business leaders in tourism, and a retired legal stenographer. All seven lived in the regions covered by these newspapers.

Each judge was assumed to be able to rate 40 news stories accurately in the available three-hour period, so each rated 20 news stories that journalists had written with support from INJECT and 20 written without it in the same period 12 months earlier. A random number generator algorithm at random.org was used to select the 40 news stories, and numbers of stories proportionate with the total number of stories written by each journalist with the support of INJECT were selected. The 40 news stories were then randomly ordered in a questionnaire using another algorithm at random.org, anonymized and presented with two 1–7 scales to capture each judge’s novelty rating and value rating of each news story.

Evaluating the INJECT Tool in Three Newsrooms

The INJECT tool was installed in the newsrooms of three regional newspapers in Norway to investigate the effectiveness of its creativity support. One research question explored was whether journalists produced news stories that were more novel and valuable with INJECT’s support.

INJECT was introduced into the daily work of four journalists in each of the three newspapers for two months in 2018, for use in Norwegian and English. The 12 journalists received INJECT training and helpdesk support and were encouraged by their editors to use INJECT. During the evaluation, the numbers of English-language entries in the creative news index increased from 2.7 million to 3.2 million and Norwegian-language entries from 260,000 to 300,000. The index also included 62,160 Norwegian-language articles from ar-

chives of the three newspapers generated by the importer component and INJECT also searched over 50,000 digital cartoons. The journalists used INJECT’s Web application version.

To investigate the research question, news stories produced by the journalists with and without the support of INJECT were rated by seven individuals with journalism expertise and/or knowledge of the regions of the three newspapers, see the sidebar “The Expert Judgment Process Used to Rate News Stories.”

INJECT was used in all three newsrooms. No major technical problems were reported. A total of 72 published stories were written with the support of INJECT by 10 of the journalists. Journalists used already-published news stories as effective starting points for new angles on stories. Based on the expert analysis, a Mann-Whitney test revealed that the novelty ratings were

greater for the news stories written with the support of INJECT (Mdn=3) than without the support of INJECT (Mdn=2), $U=6997.5$, $p<0.0001$. INJECT use was associated with an increase on the novelty of news stories, albeit from ratings that indicated low novelty of most non-INJECT news stories.

In contrast, a second Mann-Whitney test revealed the value ratings were not greater for the news stories written with the support of INJECT (Mdn=5) than without the support of INJECT (Mdn=5), $U=9156$, $p>0.05$. The average value rating of all of the news stories was 4.7 out of 7, and the lowest and highest average valued articles were 3.71 and 5.86. This was unsurprising, given that all of the news stories had passed through editorial processes.

Most of the journalists needed time to learn to use INJECT, and many reported comparisons to Google: “You need to adjust slightly, because we are used to search engines that give us the most popular hits.” INJECT use was related to journalist attitudes. Four younger journalists in one newspaper who were open to new technologies and worked more autonomously used INJECT more frequently. By contrast, more experienced journalists were less willing to adopt INJECT after the evaluation: “We seem to have certain stubbornness against using INJECT and other tools like it.”

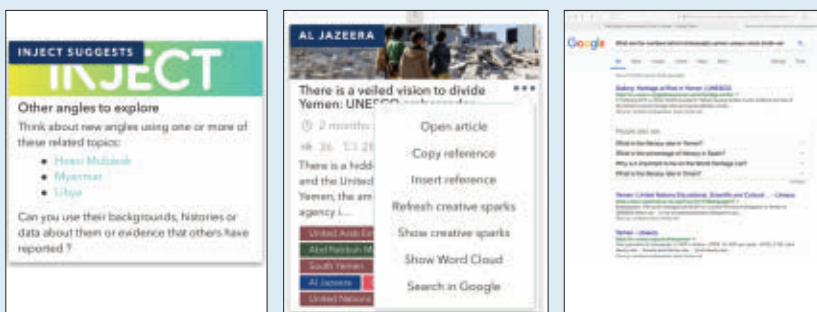
The evaluation in the three newsrooms revealed the journalists did produce news stories that were more novel if not more valuable with support from INJECT. In fact, all were published, indicating sufficient value for purpose. One interpretation of this result was the stories written without the tool’s support had value but lower novelty, that is, the stories were not creative. Articles written with the tool’s support had increased novelty but not increased value. In a strict sense, these articles were more novel rather than creative, but still had sufficient value to publish.

More results are reported in Maiden et al.¹⁵

Conclusion

Demonstrating INJECT to other news organizations reinforced our judgment that digital support for journalist creative thinking is rare.^{2,11} However, its positive reception revealed the poten-

Figure 5. INJECT features to distinguish it from Google search, including a different form of presentation of creative ideas, additional digital capabilities, and a feature to launch Google search from INJECT with the topic terms, selected angle, and title of the retrieved news item.



tial of INJECT to support journalist creative thinking.

To uphold the three journalist values uncovered during design, INJECT's interactive support was separated from indexing published news. The sidebar design enabled journalists to access INJECT's guidance in as few as two clicks, without leaving the text editor. It demonstrated how to establish digital support for creative thinking as part of journalists' daily work tools, although more evaluations are needed.

INJECT's creative news index is now an important asset of more than 500 million pieces of news information for computational manipulation. New computational analyses under development will detect patterns, biases and angles on news shown to be novel, and hence creative. One will analyze differences in topic reporting in different languages to generate angles underreported in a target language. Rolling out new INJECT versions with these features will support news businesses to remain competitive and fulfil their role in liberal democracies.

Acknowledgments

The research reported in this paper was supported by the EU-funded H2020 723328 INJECT innovation action. 

References

- Amabile, T.M. and Pratt, M.G. The dynamic componential model of creativity and innovation in organizations: Making progress, Making meaning. *Research in Organizational Behavior* 36 (2016), 157–183.
- Bartindale, T., Valentine, E., Glancy, M., Kirk, D., Wright, P. and Olivier, P. Facilitating TV production using StoryCrate. In *Proceedings of the ACM Conference on Creativity and Cognition*, 2013, 193–202; <http://doi.acm.org/10.1145/2466627.2466628>
- Broussard, M. *Artificial Intelligence for Investigative Reporting* 3, 6 (2015), 814–831.
- Cohen, S., Hamilton, J.T. and Turner, F. Computational journalism. *Commun. ACM*, 54, 10 (Oct. 2011), 66–71.
- DBpedia Spotlight (Dec. 21, 2017); <https://github.com/dbpedia-spotlight/>
- Diakopoulos, N., Naaman, M. and Kivran-Swaine, F. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *Proceedings of 2010 IEEE Symp. Visual Analytics Science and Technology*, 115–122.
- DiPaola, S. and Gabora, L. Incorporating characteristics of human creativity into an evolutionary art algorithm. In *Proceedings of the Genetic and Evolutionary Computing Conference*. D. Thierens, ed. (University College London, July 7–11, 2007) 2442–2449.
- Ekdale, N., Singer, J.B., Tully, M. and Harmsen, S. Making change: Diffusion of technological, relational, and cultural innovation in the newsroom. *Journalism and Mass Commun. Q.* 92, 4 (2015), 938–958.
- Finke, R.A., Ward, T.B. and Smith, S.M. *Creative Cognition: Theory, Research, and Applications*. MIT Press, Cambridge, MA, 1992.
- Gynnild, A. Journalism innovation leads to innovation journalism: The impact of computational exploration on changing mindsets. *Journalism* 15, 6 (2014), 713–730.
- Jacobelli, F., Birnbaum, L. and Hammond, K.J., Tell me more, not just 'more of the same'. In *Proceedings of Intelligence User Interfaces*. ACM, New York, USA, 2010, 81.

- Kaufman, J.C. and Beghetto, R.A. Beyond big and little: The four c-model of creativity. *Rev. General Psychology* 13, 1 (2009).
- Machill, M. and Beiler, M. The importance of the Internet for journalist research. *Journalism Studies* 10, 2 (2009); [doi:10.1080/14616700802337768](https://doi.org/10.1080/14616700802337768)
- Maher, M.L. and Fisher, D. Using AI to Evaluate Creative Designs. In *Proceedings of the 2nd Intern. Conf. Design Creativity 1* (2011), 45–54.
- Maiden, N., Brock, G., Zachos, K. and Brown, A. Making the news: Digital creativity support for journalists. In *Proceedings of ACM SIGCHI Conference* (Montreal, Canada, Apr. 20–27, 2018), Article 475.
- Maiden, N., Zachos, K., Brown, A., Nyre, L., Holm, B., Tonheim, A., Hesselting, C. and Apostolou, D., Evaluating the use of digital creativity support by journalists in Newsrooms. In *Proceedings of ACM Creativity and Cognition Conf.* (San Diego, CA, June 23–26, 2019); <https://doi.org/10.1145/3325480.3325484>
- Malmelin, N. and Virta, S. Managing creativity in change: Motivations and constraints of creative work in a media organization. *Journalism Practice* 10, 6 (2016); <https://doi.org/10.1080/17512786.2015.1074864>.
- McNair B. *The Sociology of Journalism*. Arnold, London, U.K., 1998.
- Michalko M. *Thinkertoys: A Handbook of Creative-Thinking Techniques*. Random House Inc., New York, USA, 2006.
- Polyglot's documentation, 2017; <http://polyglot.readthedocs.io/>
- Schofield, T., Vines, J., Higham, T., Carter, E., Atken, M. and Golding, A. Trigger shift: Participatory design of an augmented theatrical performance with young people. In *Proceedings of the ACM Conf. Creativity and Cognition*, 2013, 203–212; <http://doi.acm.org/10.1145/2466627.2466640>
- Sjøvaag, H. Homogenisation or differentiation? The effects of consolidation in the regional newspaper market. *Journalism Studies* 15, 5 (2014), 511–521.
- Thurman, N. et al. Giving computers a nose for news. *Digital Journalism* 4, 7 (2016), 838–848.
- Tolmie, P. et al. Supporting the use of user generated content in journalistic practice. In *Proceedings of the 2017 CHI Conference*. ACM, New York, USA, 3632–3644.
- Witschge, T. and Nygren, G. Journalism: a profession under pressure? *J. Media Business Studies* 6, 1 (2009), 37–59.

Neil Maiden (Neil.Maiden.1@city.ac.uk) is Professor of Digital Creativity in the Cass School of Business at City University of London, U.K.

Konstantinos Zachos is a Research Fellow in the Cass School of Business at City University of London, U.K.

Amanda Brown Amanda is a Research Fellow in the Creativity in Professional Practice, Cass Business School of City University London.

Dimitris Apostolou is an associate professor of Information Systems in the University of Piraeus and Senior Researcher in the Institute of Communication and Computer Systems of NTUA, Athens, Greece.

Balder Holm is Higher Executive Officer in the Department of Information Science and Media Studies at the University of Bergen, Norway.

Lars Nyre is a professor of media design, journalism and technology at the University of Bergen, Norway.

Aleksander Tonheim is a designer at Stacc X, Bergen, Norway.

Arend van den Beld manages the social justice organization Free Zone and media organization VJM & Cartoon Movement, which he founded in 2007 and now counts over 500 journalists in 100 countries.

Copyright held by authors.



Watch the authors discuss this work in the exclusive *Communications* video. <https://caom.acm.org/videos/digital-creativity>

Distinguished Speakers Program

A great speaker can make the difference between a good event and a WOW event!

Students and faculty can take advantage of ACM's Distinguished Speakers Program to invite renowned thought leaders in academia, industry and government to deliver compelling and insightful talks on the most important topics in computing and IT today. ACM covers the cost of transportation for the speaker to travel to your event.

speakers.acm.org



Association for Computing Machinery

DOI:10.1145/3383444

Fully appreciating the overarching scope of CS requires weaving more than ethics into the reigning curricula.

BY RANDY CONNOLLY

Why Computing Belongs Within the Social Sciences

ON OCTOBER 23, 2008, Alan Greenspan, the Chair of the U.S. Federal Reserve, was testifying before Congress in the immediate aftermath of the September 2008 financial crash. Undoubtedly the high point of the proceedings occurred when Representative Henry Waxman pressed the Chair to admit “that your view of the world, your ideology, was not right,” to which Greenspan admitted “Absolutely, precisely.”¹⁷ Fast forward 10 years to another famous *mea culpa* moment in front of Congress, that of Mark Zuckerberg on April 11, 2018. In light of both the Cambridge

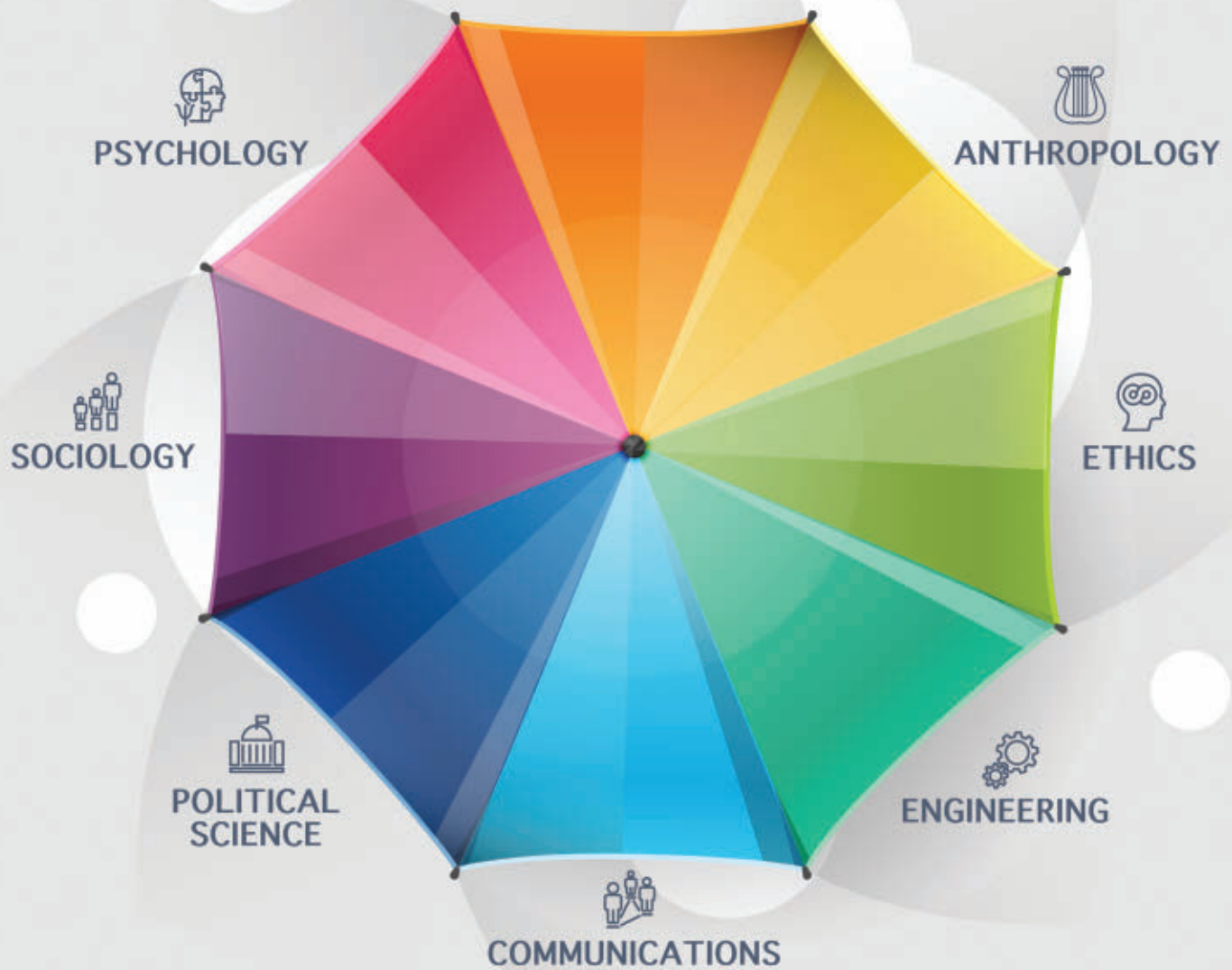
Analytica scandal and revelations of Russian interference in the 2016 U.S. election, Zuckerberg also admitted to wrong: “It’s clear now that we didn’t do enough to prevent these tools from being used for harm. That goes for fake news, foreign interference in elections, and hate speech, as well as developers and data privacy.”¹⁵

As far as *mea culpas* go, Greenspan’s was considerably more concise, but also much more insightful as to the root problem. Greenspan admitted the problem was not due to misguided user expectations, or to poorly worded license agreements, or to rogue developers. Instead he recognized the problem lay in a worldview that seemed to work for a while ... until it didn’t. In the immediate aftermath of the financial crisis, there were calls for reforms, not only of the financial services industry, but also within universities, where it was thought that unrealistic models and assumptions within economics departments²⁰ and business schools¹¹ were also responsible for inculcating a worldview that led to the crisis. It is time for us in computing departments to do some comparable soul searching.

This article is one attempt at this task. It argues the well-publicized social ills of computing will not go away simply by integrating ethics instruction or codes of conduct into computing curricula. The remedy to these ills

» key insights

- **The social ills of computing will not go away simply by integrating more ethics instruction or codes of conduct into computing curricula.**
- **A better approach to addressing these problems would be to move the academic discipline of computing away from engineering-inspired curricular models and supplement it with the methods, theories, and perspectives of the social sciences.**
- **In practice, computing is already moving tentatively into the methodological and theoretical pluralism of the social sciences, but this movement has not been fully recognized within academic computing.**



instead lies less in philosophy and more in fields such as sociology, psychology, anthropology, communications, and political science. That is, because computing as a discipline is becoming progressively more entangled within the human and social lifeworld, computing as an academic discipline must move away from engineering-inspired curricular models and integrate the analytic lenses supplied by social science theories and methodologies. To this end, the article concludes by presenting three realistic recommendations for transforming academic computing in light of this recognition.

The World View of Computing

Academic departments are not one-dimensional monoliths, so right at the start I must acknowledge there is a wide range of perspectives and

beliefs at play in any individual computing department. It's also true that computing, like any academic discipline, has somewhat arbitrary intellectual demarcations: its boundaries are less like high fences and more like a series of irregular stones that mark the rough borderlands around its domain. Computing, as a new field, initially laid out a very preliminary series of markers to help distinguish it from mathematics and engineering. Perhaps the most important of these was computing's unique disciplinary way of thinking and practicing, which, as narrated by Tedre and Denning,²² was variously labeled as "algorithmizing," "algorithmic thinking," "algorithmics," and, more recently, as "computational thinking." Over the decades, the claims made about the utility and power of this mode of thinking became increasingly ambitious and by the 2000s it became

common to argue that everyone can benefit from thinking like a computer scientist.²³ Analogous movements, such as Computer Science for All, Can-Code, and Computing at School, are all motivated by the premise that computational thinking will be a necessary part of all future work and thus it is essential that children learn it in school.

Within academia, computing and computational thinking has also followed an expansionist arc. The Digital Humanities—that is, using computational approaches and technologies within the humanities—was seen by its advocates as a way to refresh the humanities by modernizing its methods, moving it out of dusty dark libraries and into the clean, bright air of the datacenter.⁵ Computational social science is another recent curricular experiment in adopting the methodologies and techniques of computing.¹⁹

Finally, within computing itself, research has expanded significantly beyond both the analysis and creation of algorithms and the design and implementation of hardware and software architectures, to now include using these computational lenses to examine social, psychological, and cultural phenomenon more generally. ACM's *Transactions* series now covers health care, computing education, economics and computation, and social computing, and are just a sampling of this new research growth within academic computing.

Along with this expansion have come bold claims about computing's ability to better understand and explain the social world without needing background in social theory, economic models, or psychological concepts.^{1,18} While there is no doubt that many fruitful insights into social phenomenon are being made and will continue to be made through the adoption of computational approaches, it is striking how the flow of ideas appear to be in just one direction. For instance, an article in *Nature* argued it is completely legitimate for a computer scientist to study social phenomenon even with little knowledge of traditional social science methods and theories due to the insights provided by large datasets.¹² Another recent article wryly noted, it is telling that those in the traditional social sciences are often called upon to embrace computational approaches, but "computational experts dealing with social phenomenon are rarely called to conversely embrace traditional sociological thought."²

This triumphalist, almost quasi-colonizing mentality that computing appears to have in relation to other disciplines, is, at least partially, to blame for computing's current fraught relationship with other societal actors. This mentality perhaps made sense when the discipline was initially staking its academic claims in the 1970s and 1980s, or when the discipline was undergoing the harrowing student registration crisis of the first years of the 2000s. But a too-strong belief that computing provides a privileged insight, a methodologically superior set of techniques and approaches that can be applied universally and which supplies truth propositions unblemished

by social institutions, human failings, or antiquated theories, is the ideology that leads to tech executives testifying to Congress about how it all went wrong. Not only is it academically arrogant, it's short-sighted as well, because instead of *replacing* social science approaches, academic computing would be immeasurably improved by *supplementing* its own with the methods, theories, and perspectives of the social sciences. Indeed, one could even go further and make the claim that not only would computing be improved by more social science, but that computing today actually *is* a social science.

Why Computing Is a Social Science

Broadly speaking, the social sciences are a range of academic disciplines that studies human society and human individuals in the context of society. Long established fields such as sociology, economics, anthropology, psychology, and political science no doubt first come to mind when thinking about the social sciences. But disciplines such as education, law, linguistics, geography, gender studies, communications, archeology, and even business school fields such as management, marketing, and human resources can all potentially be categorized as falling under the broad net of social science. While this diversity of specialized fields can be an obstacle when it comes to generalizing about its nature, this diversity is both a strength and a reflection of the complexities of its domain of study. This is a point that needs to be reiterated. One of the key insights (and values) of the social science of the past half century is its embrace of complexity. That is, methodological and theoretical pluralism is what defines both the social sciences in general, but also its subject, humans in social, political, economic, and cultural contexts. This is seemingly quite different from the natural and engineering sciences, where the predictability of its subjects can be better assumed and thus a single methodological approach for making and evaluating knowledge claims is possible.

I would like to argue that in practice computing is already starting to move out of the methodologically singular natural/engineering sciences

and moving tentatively into the methodological pluralism of the social sciences, but that this movement has not been fully recognized within academic computing.

One can get a preliminary sense of the social scientific nature of computing by looking at one manifestation of its social scientific nature, namely, how computing is already deeply implicated in relations of power. As renowned sociologist Manuel Castells noted, power relations are "the foundational relationship of society because they construct and shape the institutions and norms that regulate social life."⁸ One of the key insights of contemporary social science has been its recognition of the role and influence of power and politics throughout our lives, our society, our institutions, and our technologies of knowledge.

In the contemporary world, power rarely relies on coercion, but instead is enacted through persuasion—that is, by the construction of meaning through knowledge production and distributed by communication systems. Scholars in the 1970s and 1980s, for instance, focused their power analysis on newspapers, radio, and TV, but in the past decade a wide range of scholars from fields as diverse as law, sociology, economics, and communications are now focused on the truth- and power-constructing regimes of data and the algorithms that process it. Power "is operationalized through the algorithm, in that the algorithmic output cements, maintains or produces certain truths."³ Or, simply, "Data are a form of power."¹⁴


As computing professionals, we often see ourselves as problem solvers in some manner. We might be using a type of algorithmic reasoning, to say, find a bug, document a process, design a data structure, or engineer a redundancy system. Very few of us would think that we are also doing politics. "I'm just creating something cool / solving a problem for my client / doing my job." It is often true that one's computing work is relatively innocuous in terms of its relationship to power. But it's not *always* true.

"In the future, how we perceive the world will be determined more and more by what is revealed to us by digital systems ... To control these is the


essence of politics.”²¹ This has already been recognized within legal studies, where scholars such as Karen Yeung, Shoshana Zuboff, Anthony Casey, and Anthony Nisbett, have made compelling arguments that algorithms are already transforming the rule- and standard-based nature of law and justice, to a privatized and force-based one implemented via algorithms. Recommendation algorithms, automated sanctioning systems, reactive violation detection and prediction systems, and nudge architectures are replacing the human agency built into our legal and political systems with an architecture of unknowable black boxes allowing the one-way surveil and control of people without any corresponding contestation.²⁴ In Casey and Niblett’s analogy,⁷ we are moving from a society of rooms, some of which have Do Not Enter signs (and thus can be ignored or violations forgiven), to a society of rooms with locked doors. As such, our range of possible action will no longer be controlled by law, but instead be controlled by code. That is, we will increasingly be disciplined by policies devised by cybersecurity professionals, using algorithms implemented by computer scientists, making use of data analytics provided by data scientists, and engineered to run hyper-efficiently by software engineers. It’s no wonder that James Susskind ends his 2018 book on the future of politics with an exhortation to computer professionals: “The future of politics will depend, in large part, on how the current generation of technologists approaches its work. That is their burden whether they like it or not.”²¹ But this reckoning will not happen unless we also are willing to make changes to computing curricula that reflects computing’s expanding role in shaping the future of our societies.

Three Recommendations for Transforming Computing

Despite the title of this essay, I’m not actually advocating for the institutionalized transfer of computing departments into social science faculties—such a move is no doubt highly impractical and implausible—but rather for a change in mentality, a recognition that the field now and in the future will have more affinities with the concerns of the academic social



The triumphalist, almost quasi-colonizing mentality that computing appears to have in relation to other disciplines, is, at least partially, to blame for computing’s current fraught relationship with other societal actors.



sciences, and fewer with the natural sciences or engineering. To get there, I have three recommendations:

Recommendation 1: Embrace other disciplines’ insight. First, computing must divest itself of its colonizing mentality toward other disciplines and to instead recognize that theoretic frameworks from outside computing have value and would indeed improve computing. Take, for instance, the subfield of data science. It has been especially good at identifying patterns in heterogeneous data sets. But to *explain* patterns and correlations “requires social theory and deep contextual knowledge.”¹⁶ Computer scientists are also increasingly finding themselves working in social and psychological domains. This work can be improved by theories and approaches already in place in those fields. A better understanding of human psychology, power, and the incentive structures in society, may have allowed us to avoid some of the socio-technical problems we face today. The lack of deep security measures in the initial Internet protocols, for instance, betrays the hopeful, but naïve understanding of human motivation held by the early pioneers of the Internet. The legitimization crises facing democracies today is at least partly a consequence of the social fragmentation enabled by digital platforms created by programmers with a minimalist understanding of what new communications modalities can do to an unprepared audience.⁴ Finally, consider the relatively newfound appreciation within AI research about how pre-existing human biases can pollute the training data using within machine learning. Perhaps less surprise would have been encountered had those working within AI been required to take, say, a course in anthropology. For almost 50 years, the most introductory anthropology training has endeavored to instill a recognition that cultural differences and perceptions of otherness biases the observations of researchers. And, yet, in AI research, we are now only starting to recognize this fact because of an institutionalized blindness to the accumulated insights of a century of social research.


For too long within computing we have instead a tendency to rely on pop-culture theories about inevitable

technology-driven social change that painted an attractive and self-satisfied veneer over our work. Moving forward, we need to do better, and be willing to inform both our work and our thinking, with the more nuanced, historically grounded, empirically supported thinking of the social sciences. We would all benefit from remembering the perspective articulated by Peter Denning: “I am now wary of believing what looks good to me as a computer scientist is good for everyone.”¹⁰


Recommendation 2: Replace some computing courses with social science ones. The best way to achieve my first recommendation is to embrace my second recommendation: modestly reduce the number of computing courses in our programs and in the ACM curricular recommendations in order to accommodate some mandatory social science courses.

I can already hear the rebuttal. “Surely there is no room for additional non-computing courses ... we don’t have enough curricular room even to cover all the essential computing topics!” I have been actively involved in the design of two of my university’s computing programs, and I too remember well that feeling of having too many topics and not enough course spots. Regardless, the perception that topic X and topic Y absolutely must make it into the curriculum are sometimes more a reflection of individual faculty desires rather than a reflection of informed pedagogy or the hireability of students.

Take, for instance, the topic of Web development. By far, it is the main source of employment for CS graduates, and yet Web development has shrunk to being just one of several sub-areas within the elective-only Platform-Based Knowledge area in the ACM 2013 CS curricula guidelines. Indeed, many CS programs do not include *any* Web topics in their curricula, a point of some astonishment by those outside the CS academy.⁹ So if we, as computing curricula experts, are willing to let our students graduate without what are arguably the most important skills needed for successful employment because we think they can learn it on their own in the workforce, then surely there are one or two



We need to do more to fully educate our computing graduates than simply teach them deontological vs. utilitarian algorithms for ethical trolley problems.



other computing topics that can also be learned after graduation, thereby opening up potential space in the curricula for non-computing courses.

But for this to happen, ACM curricular recommendations must lead the way. Future ACM curricula must acknowledge that computing students need more than just computer and mathematics courses. They must acknowledge that graduates in the 2020s will face greater responsibilities and the intellectual worldview of graduates must broaden as a recognition of how computing is both shaping and shaped by political, social, economic, and cultural institutions. If the ACM is unwilling to make these changes, the current social structure of the discipline will endlessly recreate itself, and the social ills enabled by computing will continue to surprise its creators.

Another rebuttal to this recommendation might be that “we already have a computer ethics course.” While an important first step for sure, we need to do more to fully educate our computing graduates than simply teach them deontological vs. utilitarian algorithms for ethical trolley problems. I’m not minimizing the vital work done by groups such as the ACM Committee on Professional Ethics, Computing Professionals for Social Responsibility, and Computing for the Social Good.¹³ The problem with computing ethics is that at present it stands by itself in the computing curricula. By only having a single mandated course about the relationship of computing to the wider human and social world, how can it not but strike a student that this is peripheral (and hence irrelevant) knowledge? Just one look at the curriculum and a student will no doubt get the impression that the ethics course is not all that important in comparison to courses such as numeric theory, algorithm evaluation, and programming.

This is the natural consequence of the engineering model that computing curricula seems to inhabit. That is, the belief there is so much computing and mathematics content to be learned that there is no room for anything else. As a result, we normalized the belief that the world is irrelevant next to computing precisely through the structure of our curriculum. It is sometimes said that workers of organizations adopt a

world view that is a reflection of the organizational structure of their workplace. Our students do so as well, except in this case, it's their academic discipline's organization. This is a problem though that we can fix ... or at the very least make an attempt at doing it better.

Recommendation 3: Embrace multidisciplinary through faculty hiring.

My final recommendation involves more boldly moving our discipline toward multidisciplinary. Computing has sometimes struggled with maintaining a balance between academic disciplinary coherence on the one hand, with career-oriented students, on the other, who are mainly interested in the professionally relevant topics. In this regard, computing is quite similar to its cousin in the social sciences, the discipline of communications. That field weathered a series of crises brought on by technological change and by the contrasting pulls of faculty and student interests, by embracing multi-disciplinary opportunities.

Craig Calhoun, in his 2011 plenary address on communications as a social science, argued using a metaphor from ecology that porous edges are better than sharp boundaries when it comes to newer academic disciplines such as communications. Edges are zones where ecosystems overlap, and where biodiversity and biodensity are much higher than in the central areas of any one ecosystem. "There are more songbirds at the edges of forests than in the middle."⁶ This is what computing also needs as an academic discipline: to move to the edge and to participate in the rich academic biodiversity that happens where computing interacts with other disciplines. Some researchers in CS are already there. But rather than make this an exotic vacation, it should be our discipline's home flora and fauna. And we should not inhabit this edge with a colonizing ideology that sees computational thinking as the best way to understand and inhabit this world. Indeed, the whole point of inhabiting an edge is to take strength from multiple sources, from multiple world views, from multiple methodologies and theoretic angles, and not to pave it over with a single approach.

How can this be achieved? One way would be to hire more tenurable

faculty into computing departments who are specialists in the human and social side of computing. We don't need to limit ourselves to CS Ph.D.s. There are communications, sociology, law, psychology, anthropology, and other Ph.D.s out there whose dissertation topics are clearly computing related or even computational in approach.

Conclusion

Computing professionals and academics have helped create something awesome over the past half century. Awesome is truly the appropriate word, especially if we are cognizant of its etymological heritage. "Awesome" is derived from the ancient Greek word *deinon*, and this word captures better the full dimensions of computing's awesomeness. To be *deinon* is to be both wondrous and terrifying at the same time. "There are many *deinon* creatures on the earth, but none more so than man" sings the chorus in Sophocles' tragedy *Antigone*.

Within computing we have generally only focused on the wondrous and have ignored the terrifying or delegated its reporting to other disciplines. Now, with algorithmic governance replacing legal codes, with Web platform enabled surveillance capitalism transforming economics, with machine learning automating more of the labor market, and with unexplainable, non-transparent algorithms challenging the very possibility of human agency, computing has never been more *deinon*. The consequences of these changes will not be fully faced by us but will be by our children and our students in the decades to come. We must be willing to face the realities of the future and embrace our responsibility as computing professionals and academics to change and renew our computing curricula (and the worldview it propagates). This is the task we have been given by history and for which the future will judge us. □

References

1. Anderson, C. The end of theory: The data deluge makes the scientific method obsolete. *Wired* 16, 7 (2008), 7–16.
2. Bartlett, A. et al. The locus of legitimate interpretation in big data sciences: Lessons for computational social science from -omic biology and high-energy physics. *Big Data & Society* 5, 1 (June 2018); <https://doi.org/10.1177/2053951718768831>.
3. Beer, D. The social power of algorithms. *Information, Communication & Society* 20, 1 (Jan. 2017), 1–13; <https://doi.org/10.1080/1369118X.2016.1216147>.

4. Bennett, W.L. and Pfetsch, B. Rethinking political communication in a time of disrupted public spheres. *J. Communication* 68, 2 (Apr. 2018), 243–253; <https://doi.org/10.1093/joc/jqx017>.
5. Berry, D.M. The computational turn: Thinking about the digital humanities. *Culture Machine* 12, (2011).
6. Calhoun, C. Plenary: Communication as social science (and more). *Intern. J. Communication* 5, (2011), 18.
7. Casey, A.J. and Niblett, A. The death of rules and standards. *Indiana Law J.* 92, (2016).
8. Castells, M. A sociology of power: My intellectual journey. *Annual Review of Sociology* 42, 1 (July 2016), 1–19; <https://doi.org/10.1146/annurev-soc-081715-074158>.
9. Connolly, R. Facing backwards while stumbling forwards. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2019), 518–523.
10. Denning, P.J. Remaining trouble spots with computational thinking. *Commun. ACM* 60, 6 (May 2017), 33–39; <https://doi.org/10.1145/2998438>.
11. Giacalone, R.A. and Wargo, D.T. The roots of the global financial crisis are in our business schools. *J. Business Ethics Education* 6, (2009), 147–168.
12. Giles, J. Computational social science: Making the links. *Nature* 488, 7412 (Aug. 2012), 448–450; <https://doi.org/10.1038/488448a>.
13. Goldweber, M. et al. Computing for the social good in education. *ACM Inroads* 10, 4 (Dec. 2019), 24–29; <https://doi.org/10.1145/3368206>.
14. Iliadis, A. and Russo, F. Critical data studies: An introduction. *Big Data & Society* 3, 2 (Dec. 2016); <https://doi.org/10.1177/2053951716674238>.
15. Kang, C. and Rose, V. Zuckerberg faces hostile Congress as calls for regulation mount. *New York Times* (Apr. 11, 2018).
16. Kitchin, R. Big data, new epistemologies and paradigm shifts. *Big Data & Society* 1, 1 (July 2014); <https://doi.org/10.1177/2053951714528481>.
17. Leonhardt, D. Greenspan's mea culpa. *New York Times* (Oct. 23, 2008).
18. Mayer-Schönberger, V. and Cukier, K. *Big Data: A Revolution that Will Transform How We Live, Work, and Think*. Houghton Mifflin Harcourt, 2013.
19. Shah, D.V. et al. Big data, digital media, and computational social science. *The Annals of the American Academy of Political and Social Science* 659, 1 (May 2015), 6–13; <https://doi.org/10.1177/0002716215572084>.
20. Shiller, R.J. How should the financial crisis change how we teach economics? *The J. Economic Education* 41, 4 (Sept. 2010), 403–409; <https://doi.org/10.1080/00220485.2010.510409>.
21. Susskind, J. *Future Politics: Living Together in a World Transformed by Tech*. Oxford University Press, 2018.
22. Tedre, M. and Denning, P.J. The long quest for computational thinking. In *Proceedings of the 16th Koli Calling Intern. Conf. Computing Education Research* (New York, NY, USA, 2016), 120–129.
23. Wing, J.M. Computational thinking and thinking about computing. *Philosophical Trans. Royal Society A: Mathematical, Physical and Engineering Sciences*. 366, 1881 (2008), 3717–3725; <https://doi.org/10.1098/rsta.2008.0118>.
24. Yeung, K. 'Hypernudge': Big data as a mode of regulation by design. *Information, Communication & Society* 20, 1 (Jan. 2017), 118–136; <https://doi.org/10.1080/1369118X.2016.1186713>.

Randy Connolly (rconnolly@mtroyal.ca) is a professor at Mount Royal University, Calgary, Alberta, Canada.

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



Watch the author discuss this work in the exclusive *Communications* video. <https://cacm.acm.org/videos/computing-social-sciences>

DOI:10.1145/3372122

Data on CS graduation rates among six academic institutions in NC traces the demographics of those participating (or not) in the discipline.

BY FAY COBB PAYTON AND ALEXA BUSCH

Examining Undergraduate Computer Science Participation in North Carolina

FORMER U.S. PRESIDENT Obama put forth the initiative ‘CSForAll’ in order to prepare all students to learn computer science (CS) skills and be prepared for the digital economy. The ‘ForAll’ portion of the title emphasizes the importance of inclusion in computing via the participation and creation of tools by and for diverse populations in order to “avoid the consequences of narrowly focused AI (computing and other) applications, including the risk of biases in developing algorithms, by taking advantage of a broader spectrum of experience, backgrounds, and opinions.”¹⁰ Throughout this report, the Obama administration highlighted the number one priority, and challenge, of the field of CS: to equip the next generation with CS knowledge and skills equitably in preparation for the currency of the digital economy.

An increase in government funding is part of the initiative for CSForAll. Of the \$4 billion pledged in state funding, only \$100 million is sent directly to the K–12 school system.¹⁷ The rest of the funding is set aside for research and initiatives involving policymakers to help expand CS opportunities. In just one year, the National Science Foundation (NSF) and Corporation for National and Community Service (CNCS) were called to make \$135 million in CS funding available.¹⁷ The initiative also called for “expanding access to prior NSF supported programs and professional learning communities through their CS10k that led to the creation of more inclusive and accessible CS education curriculum including “Exploring CS and Advanced Placement (AP) CS Principles.” According to Smith,¹⁷ more than 30 school districts began expanding their CS programs at the start of this initiative. A majority of this federal funding for research, such as from the NSF, is awarded to higher education.¹¹

The potential benefit in funding higher education institutions to contribute to CSForAll lends itself to research. Researchers show that engaging students in CS at the K–12 level will not solve the shortage problem if CS programs at the university level cannot scale.^{8,9} Broadening Participation in Computing (BPC) Alliances, originally comprised of 10 (now 8) initiatives, connect educational institutions of different levels, backgrounds, and resources with the collective mission to broaden participation in the CS systems and

» key insights

- **NC CS trends offer insights into undergraduate enrollment and completion among female, Black, Hispanic and Native American students.**
- **Minority serving institutions and smaller PWIs can serve as models to broadening participation.**
- **There is a need for intersectional data collection to improve our understanding of enrollment and graduation trends.**



workforce. Some BPC Alliances work toward this goal through reforming statewide systems. For example, Georgia Computes! connected the University of Georgia with the state's middle schools to recruit students for high school and college CS courses by training high school teachers to teach CS courses (including APCS), and by training college faculty to conduct summer camps and teach high school retention curricula.³ As of 2010, the state of Georgia experienced a 68% increase in high school

APCS course offerings, a 57% increase of women and a 300% increase of Hispanic students taking the APCS exam since Georgia Computes first began.³ Similarly, Into the Loop connects UCLA with Los Angeles Unified School District (one of the largest and most diverse in the U.S.) in order to focus on developing curricula and teacher training for broadening participation in CS.⁷

The Alliance for the Advancement of African American Researchers in Computing (A4RC) and Advancing

Robotics Technology for Societal Impact (ARTSI) were two partnerships created to connect students at Historically Black Colleges and Universities (HBCUs) with the resources of R1 institutions. We note these two alliances no longer exist.³ They served an important purpose to further enable and support students to successfully complete CS undergraduate programs at HBCUs and facilitated their matriculation in graduate school at other institutions. A4RC united the two types of institutions to connect students and faculty in year-round research collaborations, including methods courses, spring visits, and summer research opportunities. ARTSI had a similar yet more specific focus on robotics.³

Currently, eight NSF Alliances (that is, Access Computing, CAHSI, ECEP, iAAMCS, Exploring Computer Science, NCWIT, STARS and CRA-WP) exist to broadening participation in computing. Based in North Carolina, the Students & Technology in Academia, Research, and Service (STARS) Alliance includes universities, K-12 school districts, and community colleges. The goal of STARS is to pool resources and energy while propagating effective practices of broadening participation for underrepresented groups at the local level.³

The STARS Alliance includes North Carolina State University, UNC Charlotte, North Carolina A&T University, Duke University, and UNC Greensboro. Because of North Carolina's reputation and involvement in BPC, we expect undergraduate CS programs in NC to be some of the best equipped to broaden participation at both the undergraduate and K-12 levels. Therefore, we turn to several of these higher education institutions (that is, North Carolina A&T, NC State, UNC Charlotte, and Duke University) who have partnered with STARS and compare them to other, similar NC institutions with CS programs (Wake Forest University and UNC Chapel Hill) to inform the field if and how access to CS can be broadened at the college level. Specifically, we examine the demographics of students completing undergraduate CS degrees from 2007-2017 to see the trends over the 10-year horizon.

Table 1. Institution type and degrees offered by institution.

Institution	Institution Type			Degrees Offered		
	Public	Private	HBCU	PhD	M.S./A.	B.S./A.
Duke		+		+	+	+
NC A&T	+		+	+	+	+
NCSU	+			+	+	+
UNC, Chapel Hill	+			+	+	+
UNC, Charlotte	+			+	+	+
Wake Forest		+			+	+

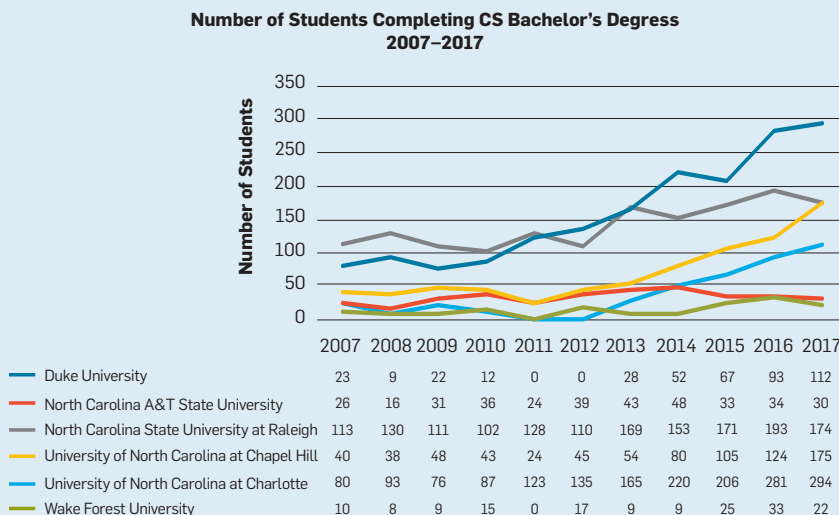
Note. += descriptive of institution

Table 2. Participation in BPC initiatives by institution.

Institution	BP Alliance Partner?	Alliance(s)
Duke	Yes	ARTSI
NC A&T	Yes	ARTSI, A4RC, STARS
NCSU	Yes	STARS
UNC, Chapel Hill	No	
UNC, Charlotte	Yes	STARS
Wake Forest	No	

Note. += descriptive of institution

Figure 1. Number of students completing CS bachelor's degrees between 2007 to 2017.



Background:**Broadening Participation**


Prior studies^{1,2,4,11,12,16} offer evidence regarding both the importance and challenges associated with broadening participation in computing. Creating firm foundations to the path of CS at the undergraduate level undoubtedly rests on social/psychological, structural, and systemic barriers. The *structural barriers* include disparities in access to and availability of rigorous computer science curricula, lack of access to peer networks/mentors/sponsors, and bias in selection processes and coursework. *Social/psychological barriers* include perceptions of who should (should not) participate, lack of cultural hooks and relevance to engage and retain students, and misconceptions about what computing is. Lastly, the *systemic barriers* are policies, practices or procedures that preclude equity in access and/or diverse pathways to computing.¹

Work by Gates and her colleagues⁴ in the context of the Hispanic population indicates that peer-led learning and affinity research group models have been successful interventions. These methods focus on academic preparation and peer-driven activities to create agency and resilience among students. In addition, culturally relevant and responsive pedagogy includes interventions, such as aligned curricula, a multi-CS course sequence, exposure to diverse CS role models, peers and instructors, and in-school and out-of-school leadership growth opportunities and equally.¹⁶


Withstanding these barriers and myriad of interventions, computing acumen² can be an equalizer to workforce opportunities in the broader society. To be an equalizer, however, broadening participation efforts must recognize the preparatory privilege associated with families that could provide parental knowledge, guidance, summer camp opportunities, in-home computers, software, even private tutoring.⁷

Institutions

In the last 10 years, researchers have acknowledged North Carolina (NC) for taking considerable measures to broaden participation in CS education. Some have recognized NC high



Creating firm foundations to the path of CS at the undergraduate level undoubtedly rests on social/psychological, structural, and systemic barriers.



schools for having a technology literacy requirement for graduation.²¹ The state also prides itself in offering classroom and online courses to all students.⁵ For example, education programs in the fundamentals of CS are available through NC's statewide career technical education programs and North Carolina Virtual Public School.²¹ NC is also recognized as one of 14 states participating in the Southern Regional Education Board (SREB) initiative.¹⁸ Funded by the Gates Foundation, this initiative developed out of SREB's Strengthening Statewide College/Career Readiness Initiative (SSCRI).

North Carolina was also an early adopter as one of only 17 states that permitted an Advanced Placement (AP) computer science course to satisfy a core math or science high school graduation requirement.⁵ This is an important distinction. By allowing AP CS to fulfill high school core requirements, North Carolina sends the message that it prioritizes computer science and recognizes it as an important part of K-12 education curricula. The state showed further support of AP CS courses in 2014 when it began paying for AP examination fees instead of requiring students to do so out-of-pocket. The idea was to cover testing fees for low-income students in order to encourage more students to take AP tests and obtain college credit for high school courses.¹⁵ This is important for AP CS as research has shown that students who take an AP computer science course are 4.5 times more likely to major in CS than those who do not.⁵ By collectively offering graduation credit for AP CS courses and paying for students to take AP exams, North Carolina has arguably taken *initial* steps towards broadening exposure and access to computer science for its students.

Due to their reputable leadership in broadening access at the K-12 level, it may be unsurprising that higher education institutions in NC receive funds to scale their CS programs. For example, Google awarded CS Capacity grants to eight universities across the U.S. in 2015. These grants provided funding for the past three years (concluding in 2018) to assist participating institutions in implementing "innova-

ative, inclusive, and sustainable approaches to address current scaling issues in university CS educational programs.”⁸ Duke University, North Carolina State University, and the University of North Carolina-Chapel Hill were CS Capacity grant recipients.

With this significant recognition and momentum, we examined how NC is actually faring in broadening participation at the undergraduate level. Therefore, we explore how many students have completed CS degrees at the following institutions in the last 10 years: Duke University, North Carolina State University, UNC Chapel Hill, North Carolina A&T University, Wake Forest University, and UNC Charlotte.

We specifically explore the participation of female, Black, Hispanic and Native American students. It should be noted this sample of schools includes two private institutions and an HBCU (see Table 1). The sample also includes those that are involved in BPC initiatives and those that are not (see Table 2). These universities are described in further detail here:

Duke University is a private, non-profit, research university located in Durham, NC. It offers BS, MS, and Ph.D. degrees in CS. Duke’s CS department is housed in their College of Engineering. Their department has participated in one of the original 10 BPC initiatives, ARTSI, which partnered

with HBCUs on the specific topic of robotics. Duke is a member of the STARS Alliance.

North Carolina A&T University is a public, research HBCU located in Greensboro, NC. It offers BS, MS, and Ph.D. degrees in CS. Its CS department is housed in the College of Engineering. NC A&T participated in A4RC and ARTSI and is a current STARS Alliance member.

North Carolina State University (NCSU) is a public research university located in Raleigh, NC. As proudly stated on their website, it is home to one of the nation’s oldest CS departments and offer degrees at the BS, several master’s options, and Ph.D. levels. The CS department at NCSU is housed in its College of Engineering. NCSU is also a partner in STARS, one of the original and still funded BPC Alliances.

The University of North Carolina at Chapel Hill is a public research university located in Chapel Hill, NC. It offers CS degrees at the bachelor’s, master’s, and Ph.D. levels. The CS department at UNC Chapel Hill is housed in their College of Arts and Sciences.

The University of North Carolina at Charlotte is a public research university located in Charlotte, NC. It offers CS degrees at the bachelor’s, master’s, and Ph.D. levels. Their CS department is housed in the College of Computing and Informatics. UNC Charlotte is also a STARS Alliance member.

Wake Forest University is a private research university institution located in Winston-Salem, NC. While it offers CS bachelor’s and master’s degrees, unlike the other institutions in our sample, it does not have a Ph.D. program. Wake Forest has not and does not participate in any of the BPC initiatives. It may also be worth noting its CS program is one of the youngest in our sample.

Current Study

Based on the literature noted here, we explored the following questions:

► In the presence of CS curricula availability at six undergraduate institutions in North Carolina, how can public educational data from the last decade inform the field about CS accessibility among female, Black, His-

Figure 2. Percentage of Black students completing CS bachelor’s degrees between 2007 to 2017.

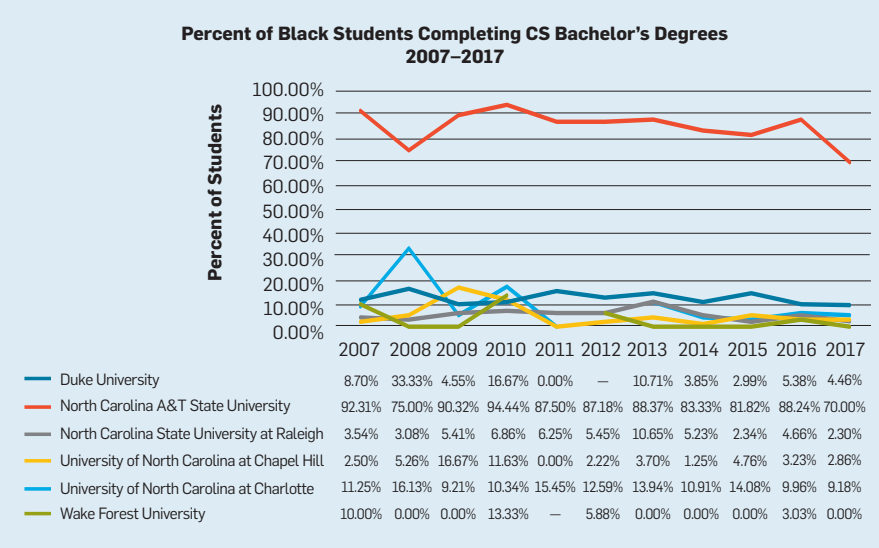
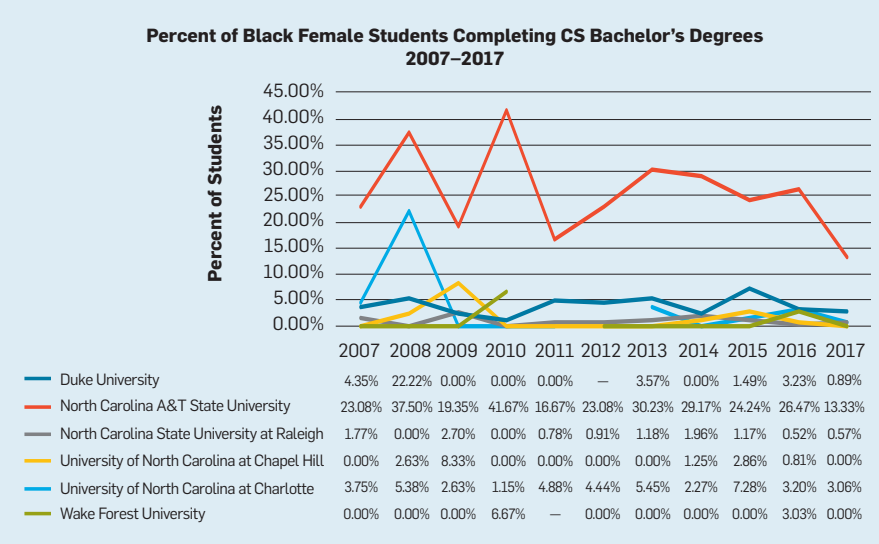


Figure 3. Percentage of Black females completing CS bachelor’s degrees between 2007 to 2017.



panic or Latino, and Native American students at the undergraduate level?

► How can these trends at the undergraduate level inform the field about CS accessibility among female, Black, Hispanic or Latino, and Native American students at the high school level?

To answer these questions, we downloaded and summarized data from the Integrated Postsecondary Education Data System (IPEDS) data collection. IPEDS data consist of statistics on postsecondary institutions regarding tuition and fees, number and types of degrees and certificates conferred, number of students applying, number of students enrolled, number of employees, financial statistics, graduation rates, student financial aid, and academic libraries. We specifically explored the number of computer science bachelor's degrees conferred by gender and race/ethnicity from 2007–2017.

Trends in the Last Decade

There was a total of 5,025 CS degrees completed from 2007–2017 across all six institutions. Overall, the number of students completing CS degrees has increased from 2007 to 2017 (see Figure 1). Wake Forest and North Carolina A&T had the smallest amount of growth. The largest growth occurred at UNC Charlotte with an increase from 2007 to 2017 of 214 students. While there has been an increase in the number of White and Asian students completing CS degrees in the last 10 years for these schools, little has changed in the number of Black students at any of the schools (see Figure 2).

Here, we show mostly percentage trends with the exception of Figure 1. We include the raw numbers (or absolute numbers) in an online appendix (<http://dl.acm.org/citation.cfm?doid=3372122&picked=formats>), which captures the scope of each institution. This helps to reduce bias in reporting or providing misleading interpretations of the data. Figure 1 shows a general increase in the number of students completing CS undergraduate degrees.

The largest percentage of students completing CS degrees were Black students from North Carolina A&T, aver-

aging an 85.32% completion rate (see Figure 2). The numbers remain small for Black students at all the other schools. In fact, most CS bachelor's degrees completed by Black students at non-HBCUs are less than 20% though the raw data (see online appendix) shows an upward trend at UNC-Charlotte and North Carolina A&T. Across all non-HBCU institutions, more White students completed CS bachelor's degrees than all other ethnicities. However, gender trends remain consistent across all institutions in this study. We found that a total of 215 Black males and 94 Black females completed CS bachelor's degrees at North Carolina A&T between

2007–2017 over the 10-year horizon (see Figure 3). This data shows that Black female completion is somewhat jagged with some spikes to 25 and 26 Black males in more recent years.

The lack of Native American students completing CS bachelor's degrees across these schools is particularly staggering. Despite NC being a state with one of the largest Native American populations,²² almost no Native American students have completed CS degrees across these schools in the last decade. The percentage of Native American students completing CS degrees is below 3.5% across all six institutions. This represents 12 students total over the time horizon as

Figure 4. Native American males completing CS bachelor's degrees between 2007 to 2017.

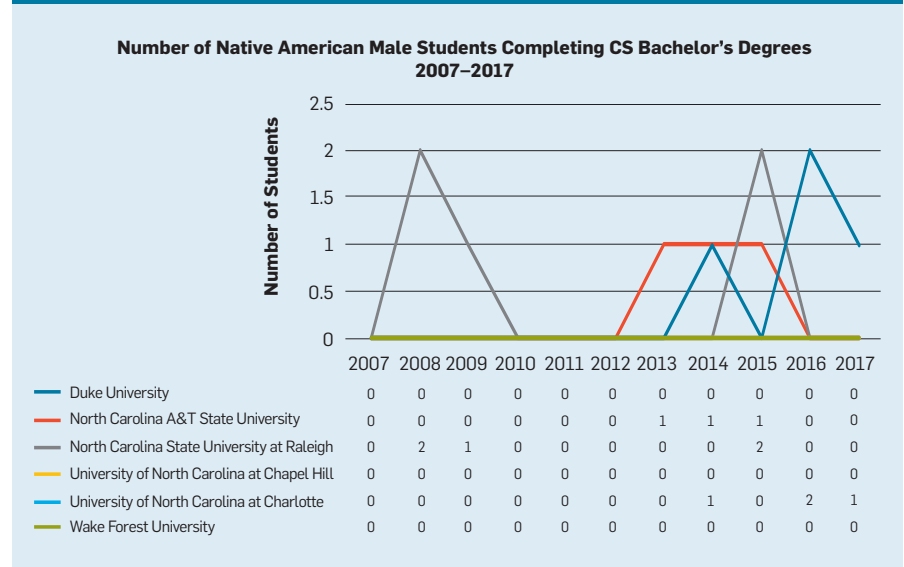
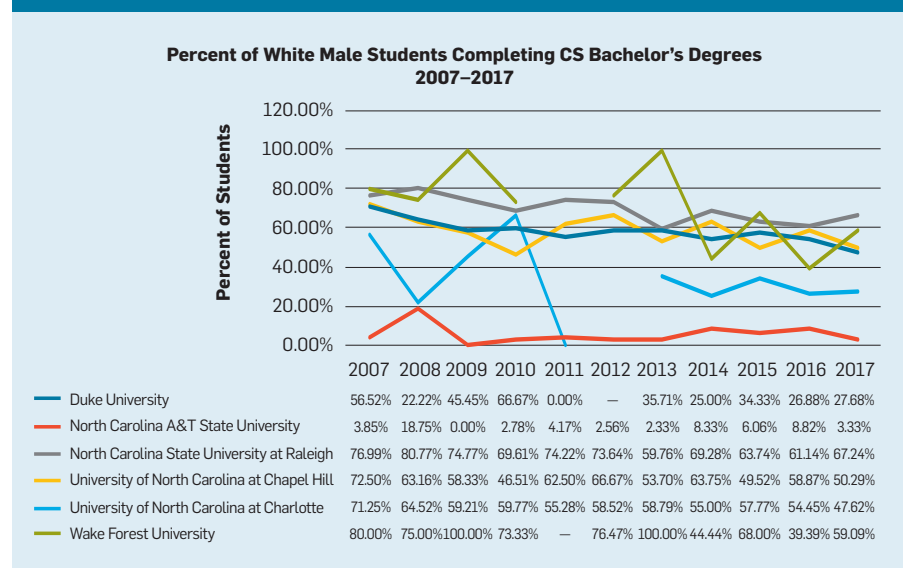


Figure 5. Percentage of White males completing CS bachelor's degrees between 2007 to 2017.



noted in Figure 4. None of these students are female, meaning zero Native American females completed CS bachelor's degrees across these institutions in the last 10 years.

Duke and UNC Chapel Hill had the largest percentage of Asian students completing CS degrees. The raw data indicated that Duke and UNC Chapel Hill, respectively, graduated 122 and 126 Asian students over the 10-year horizon. For all other institutions, less than 15% of students completing CS degrees in the last 10 years were Asian. Also, in the last 10 years, less than 12% of students completing CS degrees identified as Hispanic across all insti-

tutions, with the exception of Wake Forest in 2014. The numbers were considerably worse for women. Women identifying as Hispanic only made up only 4.55% of those who completed CS degrees in each institution.

Across all institutions, males completed more CS degrees than females. While the number of males is increasing, the number of females is not increasing at the same rate. At the majority of schools (with the exception of the one HBCU), CS degrees were completed by White males. White men accounted for 100% of all CS bachelor's degrees completed at Wake Forest University in both 2009 and 2013 (see

Figure 5). While the number of completed CS bachelor's degrees appears to be increasing across these institutions (see Figure 1) in NC, the diversity seems to follow suit as the number of White males decreases as the data illustrates (Figure 5). This diversity, however, is still met with a lack of representation among ethnic groups as the raw data in the online appendix indicates.

In comparison, Figure 6 shows the data for White females. For this group, data were missing for Duke University in 2012 and Wake Forest University in 2011. In 2013, none of the institutions had double-digits percentage completion rates for White females. From North Carolina A&T State University, White females completed CS undergraduates in 2007 (3.85%), 2012 (2.56%) and 2017 (3.33%). The largest one-year percentages for this demographic occurred in 2014 (33.33%) and 2008 (25%) at Wake Forest University. Overall, this data does not show a consistent pattern of sustained growth for the group.

Prior work¹³ indicated that NC was one of the states with the fastest growing Hispanic population. From 2000 to 2010, North Carolina had a 141% change in its Hispanic population. In 2014, North Carolina ranked eleventh in the Hispanic population among all 50 states and the District of Columbia with 890,000.¹³ From the 2018 U.S. Census Bureau,¹³ Hispanic/Latino, Black/African Americans, Native Americans, Whites, and Asians represent 9.5%, 22.2%, 1.6%, 63.1% and 3.1%, respectively, of NC's population.

Given this growing demographic nationally and in NC, we provide Figures 7, 8, and 9 to explore insights on Hispanic CS graduation rates in the state. Figure 7 shows the largest spike in 2014 (22.22%) at Wake Forest University. For a more careful observation of this spike, Figure 10 also shows the raw data with UNCC graduating an increasing number of Hispanic students in CS. The raw data also indicates that 22 Hispanic completed CS degrees from UNCC and two graduated from Wake Forest. The University of North Carolina-Charlotte (UNCC) shows some upward trend in the latter years of the dataset.

Our findings are based on repre-

Figure 6. Percentage of White females completing CS bachelor's degrees between 2007 to 2017.

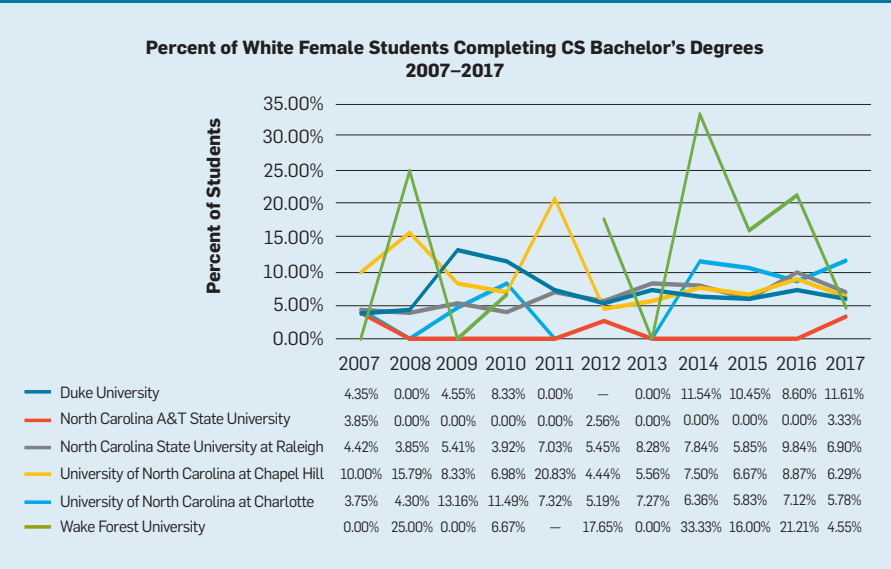
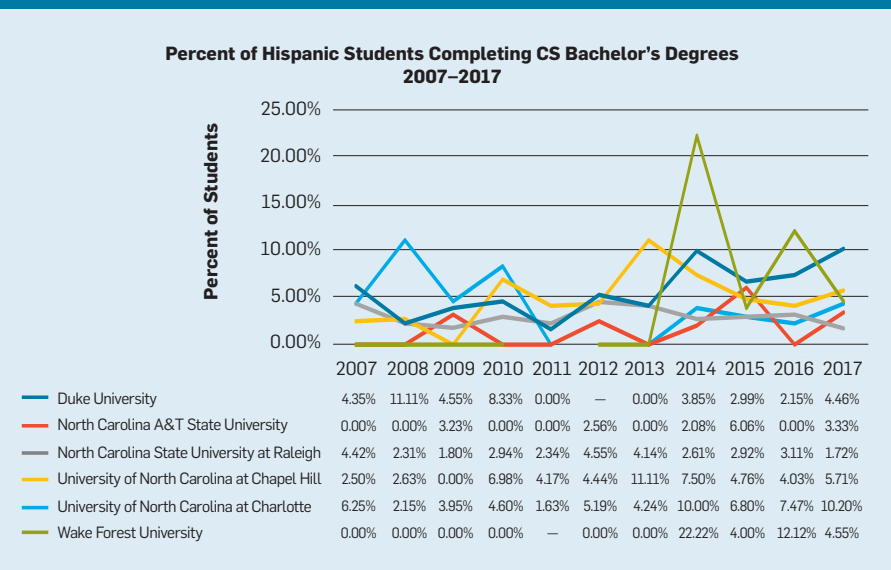


Figure 7. Percentage of Hispanic students completing CS bachelor's degrees between 2007 to 2017.



sensation. Notably, the composition of a program (demographics) can only change very slowly. Recruitment and retention of students is a slow-moving goal with enrollees from underserved groups with the hopes of translating to significant percentage growth and enrollment. This, however, in the disaggregate creates a false sense of success relative to broadening participation in CS. Hence, students benefitting from preparatory privilege would be more naturally attracted to these careers while other mechanisms should consider alternative approaches to broadening participation.

HBCUs, Hispanic-serving and other minority-serving institutions do seem to be doing well with regards to answering the call to broadening participation. This, however, is not new for these institutions—as institutional culture and mission have driven these efforts to attract and retain underserved and marginalized groups²⁰ as well as prepare them for graduate education and workforce alternatives. This brings to bear if there should be varied types of BPC strategies based on institutional types. This will require thoughtful intention, context, and culturally attuned climates beyond the pure numbers, and the awareness that representation is not inclusion. Our study does not account for critical factors, such as student enrollment/majors, and student/faculty diversity data, which could offer a clearer comparison among the institutions listed in this manuscript.

Conclusion

We assert that institutions should examine trends in public educational data as leadership and other stakeholders formulate strategies and make decisions around broadening participation. In this study, we used IPEDS data for a 10-year horizon. As noted on the National Center for Education Statistics website, “IPEDS annually gathers information from about 7,000 colleges, universities, and technical and vocational institutions that participate in the federal student aid programs.” This captures a variety of college and university types, allows for institutional comparisons and incorporates the Classification of Instructional Programs (CIP) system tax-

onomy which categorizes a discipline by groupings.

While our work involved analyses from the IPEDS data which has a broader definition of computing based on degree program codes in computer and information sciences, the CIP taxonomy classifications often do not align with the precise names of majors. In the ever-changing field of CS, this can create significant variability in how institutions analyze the data and the decisions that they make. Further, IPEDS is based on self-report, and institutional time burden and resources influence IPEDS data collection which can create additional barriers

to MSIs and community colleges.¹⁴ Alternative datasets, such as ACM’s Survey of Non-Doctoral Granting Departments in Computing, CRA’s Taulbee Survey, and the National Center for Education Statistics, can also provide some comparison of our results. Withstanding the dataset, the role of MSIs and community colleges expertise (both CS domain and inclusive BP acumen) should not be ignored. We are cognizant of the impacts of institutions’ admissions policies, role of academic preparation, curricula access and availability at the K-12 level, broader discipline career fit and biases that can impact this discourse.

Figure 8. Percentage of Hispanic males completing CS bachelor’s degrees between 2007 to 2017.

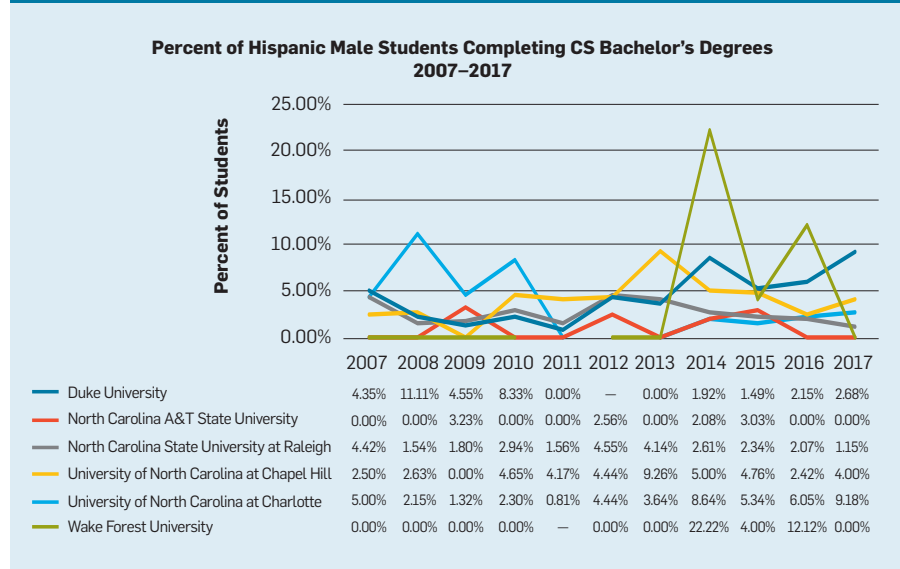
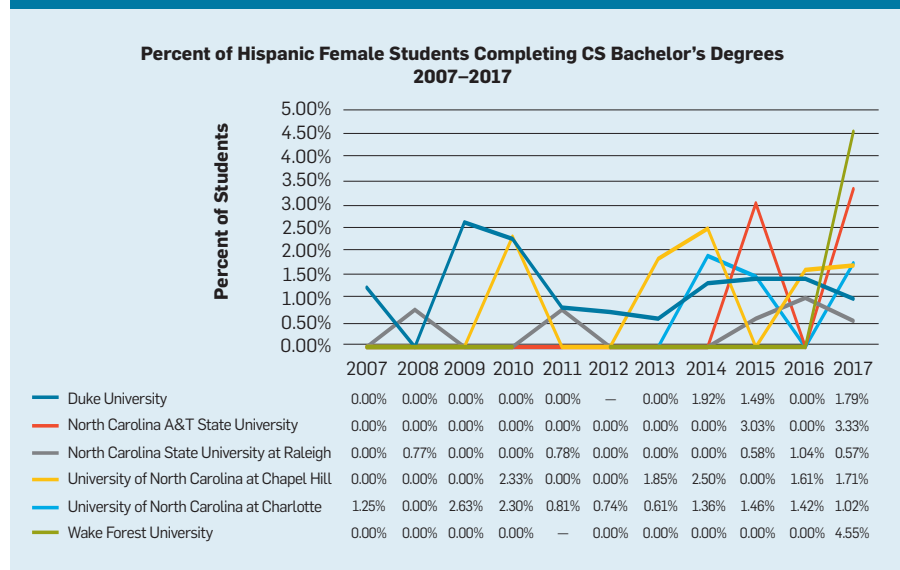


Figure 9. Percentage of Hispanic females completing CS bachelor’s degrees between 2007 to 2017.



We also contend these decisions should be anchored in an effective, contextualize broadening participation strategy. Engaging students in CS at the K-12 level will not solve the shortage problem if CS programs at the university level cannot scale.^{8,9} To address this question, we observed the following: Of these six institutions, four have participated in at least one BPC project in the last 10 years. Our results indicate that some progress has been made in the last decade across large R1 institutions in our sample. While North Carolina A&T and UNC-Charlotte have shown growth the raw data, these institutions can serve as models for effective strategies for targeting Black and Hispanic students, respectively. The efforts at both institutions can be viewed as part of the organizational culture with some critical mass of faculty committed to integrating broadening participation into research initiatives.

Thirdly, there is a need to disaggregate the numbers and employ *intersectional* (race/ethnicity and gender via those underrepresented in the field) interpretations to get a true picture of who is participating and graduating in CS. Larger predominately white institutions (PWIs), agencies and corporate foundations can stand to learn from HBCUs (in this case North Carolina A&T), minority-serving institutions (MSIs) and other colleges (in this case, UNC-Charlotte) about inclusive excellence to enhance broadening

participation practices, strategies and culture—and provide equitable funding where there are notable results.

Given our results, an examination of North Carolina’s MSIs, community colleges and smaller PWIs is worth exploring. In addition, the definition of computing is broader than CS and can draw from information sciences/technology which can show offer addition insights regarding participation in the field. When aggregated, the numbers appear promising, and the number of students completing CS bachelor’s degrees is increasing. However, it appears to be increasing at much higher rates for White men than for any other group. Some groups have not participated any more or less in the last 10 years. In fact, we found *zero Native American women* to have completed a CS degree in the last decade at any of the six institutions. Though North Carolina is not home to a Hispanic-serving institution despite the state’s growing Hispanic population, has a significant Black demographic and is home to a significant Native American population, IPEDS, and other data sources can be used to (re)formulate decisions associated with CS participation and (re)develop more inclusive programs.

Acknowledgments. This research was funded by NSF grant number, CNS 1740141. C

References

1. Abu-El-Haija, L. and Payton, F.C. Computer Science Enrollment in Magnet High Schools: Issues of

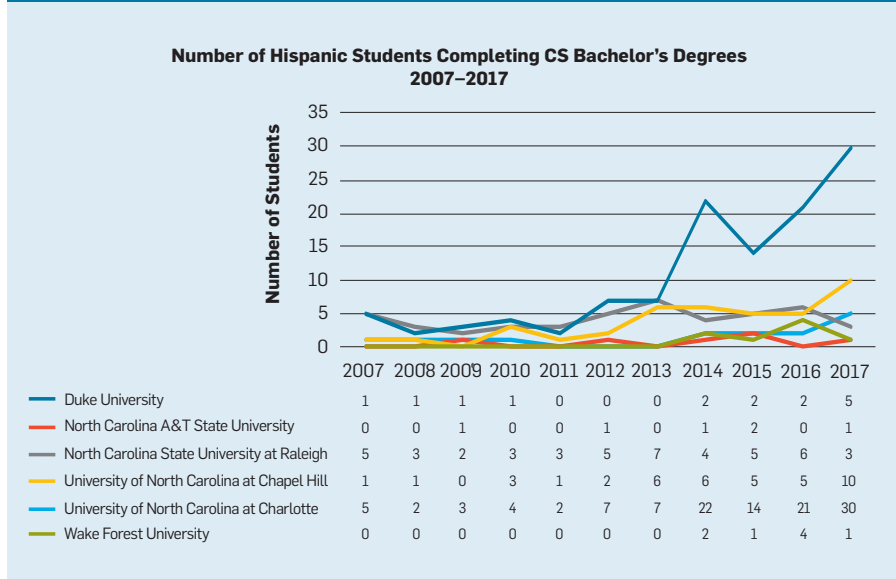
- Curricula Clusters, Equity and Pathways. RESPECT. Minneapolis, MN, 2019.
2. Bobb, K. Broadening Participation in Computing: A Critical Perspective. *ACM Inroads* 7, 4 (Dec. 2016), 49–51.
3. Chubin, D.E. and Johnson, R.Y. *Telling the Stories of the BPC Alliances: How one NSF program is changing the face of computing*. American Association for the Advancement of Science. (2010).
4. Gates, A.Q., Thiry, H. and Hug, S. Reflections: The Computing Alliance of Hispanic-Serving Institutions. *ACM Inroads* 7, 4 (Dec. 2016), 69–73.
5. Kaczmarczyk, L. and Doplick, R. and EP Committee. *Rebooting the Pathway to Success: Preparing students for computing workforce needs in the United States*. ACM, New York, NY, 2014.
6. Lehman, K.J., Sax, L.J. and Zimmerman, H.B. Women planning to major in computer science: Who are they and what makes them unique? *Computer Science Education* 26, 4 (2017), 277–298.
7. Margolis, J., Estrella, R., Goode, J., Holmes, J.J. and Nao, K. *Stuck in the Shallow End: Education, Race, and Computing*. MIT Press, Cambridge, MA, 2010.
8. Nager, A. and Atkinson, R. The case for improving U.S. computer science education. (2016).
9. National Academy of Engineering, and Institute of Medicine. *Rising above the gathering storm: Energizing and employing America for a brighter economic future*. The National Academies Press, Washington, D.C. 2007, 1–591.
10. National Science and Technology Council Committee on Technology. *Preparing for the Future of Artificial Intelligence*. Dec. 2016.
11. The National Science Foundation. Survey of federal science and engineering support to universities, colleges, and nonprofit institutions fiscal year 2015; <https://ncesdata.nsf.gov/fedsupport/2015/>
12. Payton, F.C. and Berki, E. Countering the negative image of women in computing. *Commun. ACM* 62, 5 (2019), 56–63.
13. Pew Research Center—Hispanic Trends. Demographic Profile of Hispanics in North Carolina, 2014; <https://www.pewhispanic.org/states/state/nc/>
14. Powers, K. and Henderson, A.E., eds. *Burden or Benefit: External Data Reporting: New Directions for Institutional Research*. Jossey-Bass Publishing, San Francisco, CA, 2016.
15. Ryoo, J., Goode, J. and Margolis, J. It takes a village: Supporting inquiry- and equity-oriented computer science pedagogy through a professional learning community. *Computer Science Education* 24, 4 (2015), 351–370.
16. Scott, A., Martin, A., McAlear, F., and Koshy, S. Broadening participation in computing: Examining experiences of girls of color. *ACM Inroads* 8, 4 (Dec. 2017), 48–52.
17. Smith, M. Computer Science for All. Blog, Jan. 30, 2016; <http://bit.ly/38SYAj6>
18. Southern Regional Education Board. *Bridging the Computer Science Education Gap: Five actions states can take*. Report of the SREB Commission on Computer Science and Information Technology. Nov. 2016.
19. U.S. Census Bureau Quick Facts—North Carolina, 2018; <https://www.census.gov/quickfacts/fact/table/NC/PST045218>.
20. Varma, R. Why so few women enroll in computing? Gender and ethnic differences in students’ perception. *Computer Science Education* 20, 4 (2010), 301–316.
21. Wilson, C., Sudol, L.A., Stephenson, C. and Stehlick, M. *Running on Empty: The failure to teach K-12 computer science in the digital age*. ACM and CSTA, 2010.
22. World Atlas. US States with the Largest Native American Populations, 2017; <http://bit.ly/32huH9R>.

The appendix for this article can be found at <http://dl.acm.org/citation.cfm?doi=3372122&picked=formats>

Fay Cobb Payton (fcpayton@ncsu.edu) is a professor and University Scholar of Information Technology/Analytics at North Carolina State University, Raleigh, NC, USA.

Alexa Busch (agbusch@ncsu.edu) received her master’s degree in Operations Research at North Carolina State University, Raleigh, NC, USA.

Figure 10. Hispanic students completing CS bachelor’s degrees between 2007 to 2017.



volume
01

number
01

FIRST
ISSUE
PUBLISHED

ACM Transactions on Internet of Things
is now available in
the ACM Digital Library



ACM Transactions on Internet of Things (TIOT) publishes novel research contributions and experience reports in several research domains whose synergy and interrelations enable the IoT vision. TIOT focuses on system designs, end-to-end architectures, and enabling technologies, and on publishing results and insights corroborated by a strong experimental component.

Research replication only works if there is confidence built into the results.

BY ANDY COCKBURN, PIERRE DRAGICEVIC,
LONNI BESANÇON, AND CARL GUTWIN

Threats of a Replication Crisis in Empirical Computer Science

“If we do not live up to the traditional standards of science, there will come a time when no one takes us seriously.”

—Peter J. Denning, 1980.¹³

FORTY YEARS AGO, Denning argued that computer science research could be strengthened by increased adoption of the scientific experimental method. Through the intervening decades, Denning’s call has been answered. Few computer science graduate students would now complete their studies without some introduction to experimental hypothesis testing, and computer science research papers routinely use p -values to formally assess the evidential strength of experiments. Our analysis of the 10 most-downloaded

articles from 41 ACM *Transactions* journals showed that statistical significance was used as an evidentiary criterion in 61 articles (15%) across 21 different journals (51%), and in varied domains: from the evaluation of classification algorithms, to comparing the performance of cloud computing platforms, to assessing a new video-delivery technique in terms of quality of experience.

While computer science research has increased its use of experimental methods, the scientific community’s faith in these methods has been eroded in several areas, leading to a ‘replication crisis’^{27,32} in which experimental results cannot be reproduced and published findings are mistrusted. Consequently, many disciplines have taken steps to understand and try to address these problems. In particular, misuse of statistical significance as the standard of evidence for experimental success has been identified as a key contributor in the replication crisis. But there has been relatively little debate within computer science about this problem or how to address it. If computer science fails to adapt while others move on to new standards then Denning’s concern will return—other disciplines will stop taking us seriously.

» key insights

- Many areas of computer science research (performance analysis, software engineering, AI, and human-computer interaction) validate research claims by using statistical significance as the standard of evidence.
- A loss of confidence in statistically significant findings is plaguing other empirical disciplines, yet there has been relatively little debate of this issue and its associated ‘replication crisis’ in CS.
- We review factors that have contributed to the crisis in other disciplines, with a focus on problems stemming from an over-reliance on—and misuse of—null hypothesis significance testing.
- Our analysis of papers published in a cross section of CS journals suggests a large proportion of CS research faces the same threats to replication as those encountered in other areas.

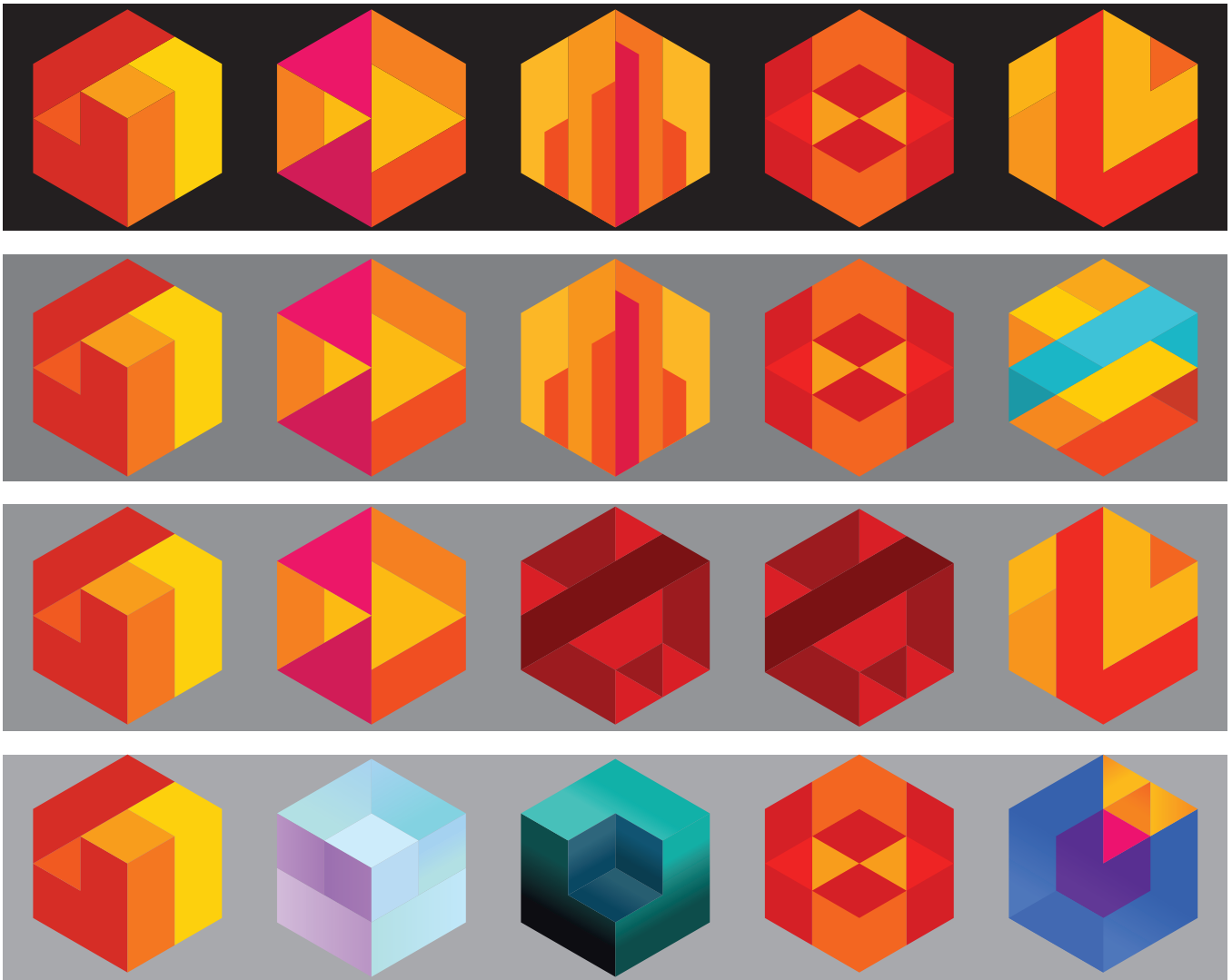


IMAGE BY ANDRZJ BORYS ASSOCIATES, USING SHUTTERSTOCK

Beyond issues of statistical significance, computer science research raises some distinct challenges and opportunities for experimental replication. Computer science research often relies on complex artifacts such as source code and datasets, and with appropriate packaging, replication of some computer experiments can be substantially automated. The replicability problems associated with access to research artifacts have been broadly discussed in computer systems research (for example, Krishnamurthi²⁵ and Collberg⁹), and the ACM now awards badges to recognize work that is *repeatable* (the original team of researchers can reliably produce the same result using the same experimental setup), *replicable* (a different team can produce the

same result using the original setup), and *reproducible* (a different team can produce the same result using a different experimental setup).⁵ However, these definitions are primarily directed at experiments that analyze the results of computations (such as new computer algorithms, systems, or methods), and uptake of the badges has been slow in fields involving experiments with human participants. Furthermore, the main issues contributing to the replication crisis in other experimental disciplines do not stem from access to artifacts; rather, they largely stem from a misuse of evidentiary criteria used to determine whether an experiment was successful or not.

Here, we review the extent and causes of the replication crisis in other

areas of science, with a focus on issues relating to the use of null hypothesis significance (NHST) as an evidentiary criterion. We then report on our analysis of a cross section of computer science publications to identify how common NHST is in our discipline. Later, we review potential solutions, dealing first with alternative ways to analyze data and present evidence for hypothesized effects, and second arguing for improved openness and transparency in experimental research.

The Replication Crisis in Other Areas of Science

In assessing the scale of the crisis in their discipline, cancer researchers attempted to reproduce the findings of landmark research papers, finding they

Some Terminology

- ▶ **Publication bias:** Papers supporting their hypotheses are accepted for publication at a much higher rate than those that do not.
- ▶ **File drawer effect:** Null findings tend to be unpublished and therefore hidden from the scientific community.
- ▶ **p-hacking:** Manipulation of experimental and analysis methods to produce statistically significant results. Used as a collective term in this paper for a variety of undesirable research practices.
- ▶ **p-fishing:** seeking statistically significant effects beyond the original hypothesis.
- ▶ **HARKing:** Hypothesising After the Results are Known: Post-hoc reframing of experimental intentions to present a *p*-fished outcome as having been predicted from the start.

could not do so in 47 of 53 cases,³ and psychology researchers similarly failed to replicate 39 out of 100 studies.³¹ Results of a recent *Nature* survey of more than 1,500 researchers found that 90% agree there is a crisis, that more than 70% had tried and failed to reproduce another scientist's experiments, and that more than half had failed to replicate their own findings.²

Experimental process. A scientist's typical process for experimental work is summarized along the top row of Figure 1, with areas of concern and potential solutions shown in the lower rows. In this process, initial ideas and beliefs (item 1) are refined through formative explorations (2), leading to the development of specific hypotheses and associated predictions (3). An experiment is designed and conducted

(4, 5) to test the hypotheses, and the resultant data is analyzed and compared with the predictions (6). Finally, results are interpreted (7), possibly leading to adjustment of ideas and beliefs.

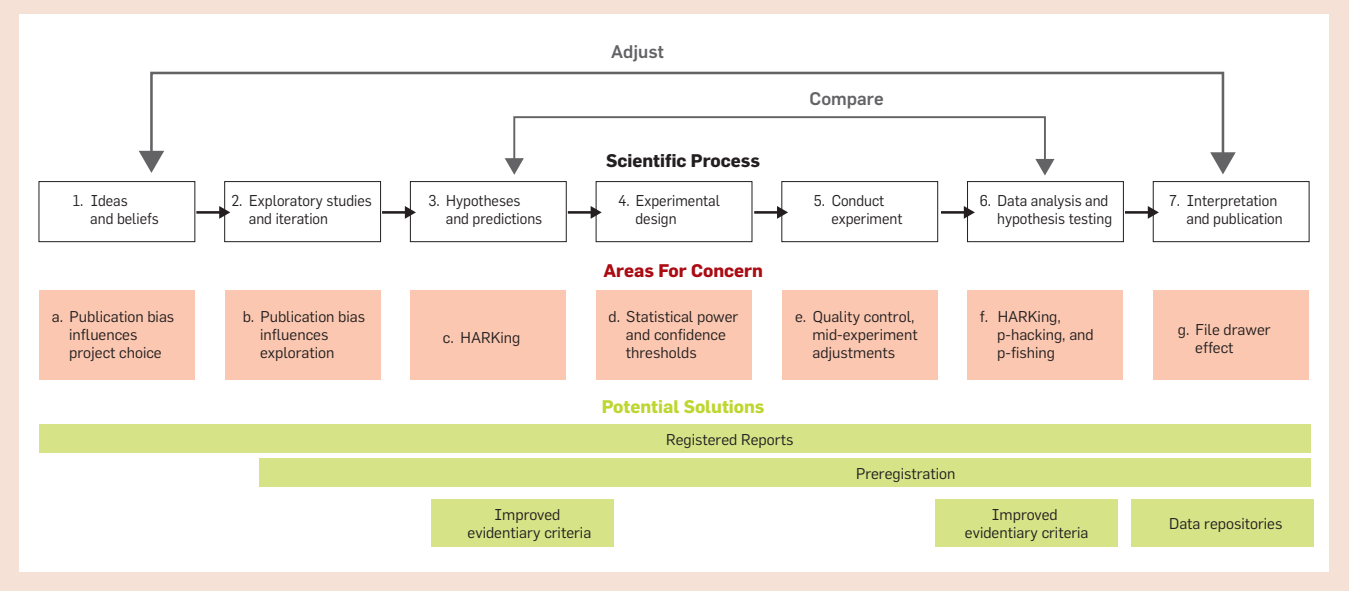
A critical part of this process concerns the evidentiary criteria used for determining whether experimental results (at 6) conform with hypotheses (at 3). Null hypothesis significance testing (NHST) is one of the main methods for providing this evidence. When using NHST, a *p*-value is calculated that represents the probability of encountering data at least as extreme as the observed data if a null hypothesis of no effect were true. If that probability is lower than a threshold value (the α level, normally .05, representing the Type I error rate of false positives) then the null hypothesis is deemed un-

tenable and the resultant finding is labelled 'statistically significant.' When the *p*-value exceeds the α level, results interpretation is not straightforward—perhaps there is no effect, or perhaps the experiment lacked sufficient power to expose a real effect (a Type II error or false negative, where β represents the probability of this type of error).

Publication bias. In theory, rejection of the null hypothesis should elevate confidence that observed effects are real and repeatable. But concerns about the dichotomous interpretation of NHST as 'significant' or not have been raised for almost 60 years. Many of these concerns stem from a troublesome *publication bias* in which papers that reject the null hypothesis are accepted for publication at a much higher rate than those that do not. Demonstrating this effect, Sterling⁴¹ analyzed 362 papers published in major psychology journals between 1955 and 1956, noting that 97.3% of papers that used NHST rejected the null hypothesis.

The high publication rates for papers that reject the null hypothesis contributes to a *file drawer effect*³⁵ in which papers that fail to reject the null go unpublished because they are not written up, written up but not submitted, or submitted and rejected.¹⁶ Publication bias and the file drawer effect combine to propagate the dissemination and maintenance of false knowledge: through the file drawer ef-

Figure 1. Stages of a typical experimental process (top, adapted from Gundersen¹⁹), prevalent concerns at each stage (middle), and potential solutions (bottom).



fect, correct findings of no effect are unpublished and hidden from view; and through publication bias, a single incorrect chance finding (a 1:20 chance at $\alpha = .05$, if the null hypothesis is true) can be published and become part of a discipline's *wrong* knowledge.

Ideally, scientists are objective and dispassionate throughout their investigations, but knowledge of the publication bias strongly opposes these ideals. Publication success shapes careers, so researchers need their experiments to succeed (rejecting the null in order to get published), creating many areas of concern (middle row of Figure 1), as follows.

Publication bias negatively influences project selection. There are risks that the direction of entire disciplines can be negatively affected by publication bias (Figure 1a and g). Consider a young faculty member or graduate student who has a choice between two research projects: one that is mundane, but likely to satisfy a perceived publication criterion of $p < .05$; the other is exciting but risky in that results cannot be anticipated and may end up in a file drawer. Publication bias is likely to draw researchers towards safer topics in which outcomes are more certain, potentially stifling researchers' interest in risky questions.

Publication bias also disincentivizes replication, which is a critical element of scientific validation. Researchers' low motivation to conduct replications is easy to understand—a successful replication is likely to be rejected because it merely confirms what is already 'known,' while a failure to replicate is likely to be rejected for failing to satisfy the $p < .05$ publication criterion.

Publication bias disincentivizes exploratory research. Exploratory studies and iteration play an important role in the scientific process (Figure 1b). This is particularly true in areas of computer science, such as human-computer interaction, where there may be a range of alternative solutions to a problem. Initial testing can quickly establish viability and provide directions for iterative refinement. Insights from explorations can be valuable for the research community, but if reviewers have been trained to expect standards of statistical evidence that only apply to confirmatory studies (such as the ubiquitous p -value) then publishing insights from

Publication bias disincentivizes replication, which is a critical element of scientific validation.

Figure 2. HARKing (Hypothesizing After the Results are Known) is an instance of the Texas sharpshooter fallacy. Illustration by Dirk-Jan Hoek, CC-BY.



exploratory studies and exploratory data analyses may be difficult. In addition, scientists' foreknowledge that exploratory studies may suffer from these problems can deter them from carrying out the exploratory step.

Publication bias encourages HARKing. Publication bias encourages researchers to explore hypotheses that are different to those that they originally set out to test (Figure 1c and f). This practice is called 'HARKing',²³ which stands for Hypothesizing After the Results are Known, also known as 'outcome switching'.

Diligent researchers will typically record a wide set of experimental data beyond that required to test their intended hypotheses—this is good practice, as doing so may help interpret and explain experimental observations. However, publication bias creates strong incentives for scientists to ensure that their experiments produce statistically significant results. Consciously or subconsciously, they may steer their studies to ensure that experimental data satisfies $p < .05$. If the researcher's initial hypothesis fails (concerning task time, say) but some other data satisfies $p < .05$ (error rate, for example), then authors may be tempted to reframe the study around the data that will increase the paper's chance of acceptance, presenting the paper as having predicted that outcome from the start. This reporting practice, which is an instance of the so-called "Texas sharpshooter fallacy" (see Figure 2), essentially invalidates the NHST procedure due to inflated Type I error rates. For example, if a researcher collects 15 dependent

variables and only reports statistically significant ones, and if we assume that in reality the experimental manipulation has no effect on any of the variables, then the probability of a Type I error is 54% instead of the advertised 5%.¹⁹

While many scientists might agree that *other* scientists are susceptible to questionable reporting practices such as HARKing, evidence suggests they are troublesomely widespread.^{20,21} For example, over 63% of respondents to a survey of 2,000 psychology researchers admitted failing to report all dependent measures, which is often associated with the selective reporting of favorable findings.²⁰

Even without any intention to misrepresent data, scientists are susceptible to cognitive biases that may promote misrepresentations: for example, *apophenia* is the tendency to see patterns in data where none exists, and it has been raised as a particular concern for big-data analyses;⁶ *confirmation bias* is the tendency to favor evidence that aligns with prior beliefs or hypotheses;³⁰ and *hindsight bias* is the tendency to see an outcome as having been predictable

from the start,³⁶ which may falsely assuage researchers' concerns when reframing their study around a hypothesis that differs from the original.

Publication bias encourages mid-experiment adjustments. In addition to the modification of hypotheses, other aspects of an experiment may be modified during its execution (Figure 1e), and the modifications may go unreported in the final paper. For example, the number of samples in the study may be increased mid-experiment in response to a failure to obtain statistical significance (56% of psychologists self-admitted to this questionable practice²⁰). This, again, inflates Type I error rates, which impairs the validity of NHST.

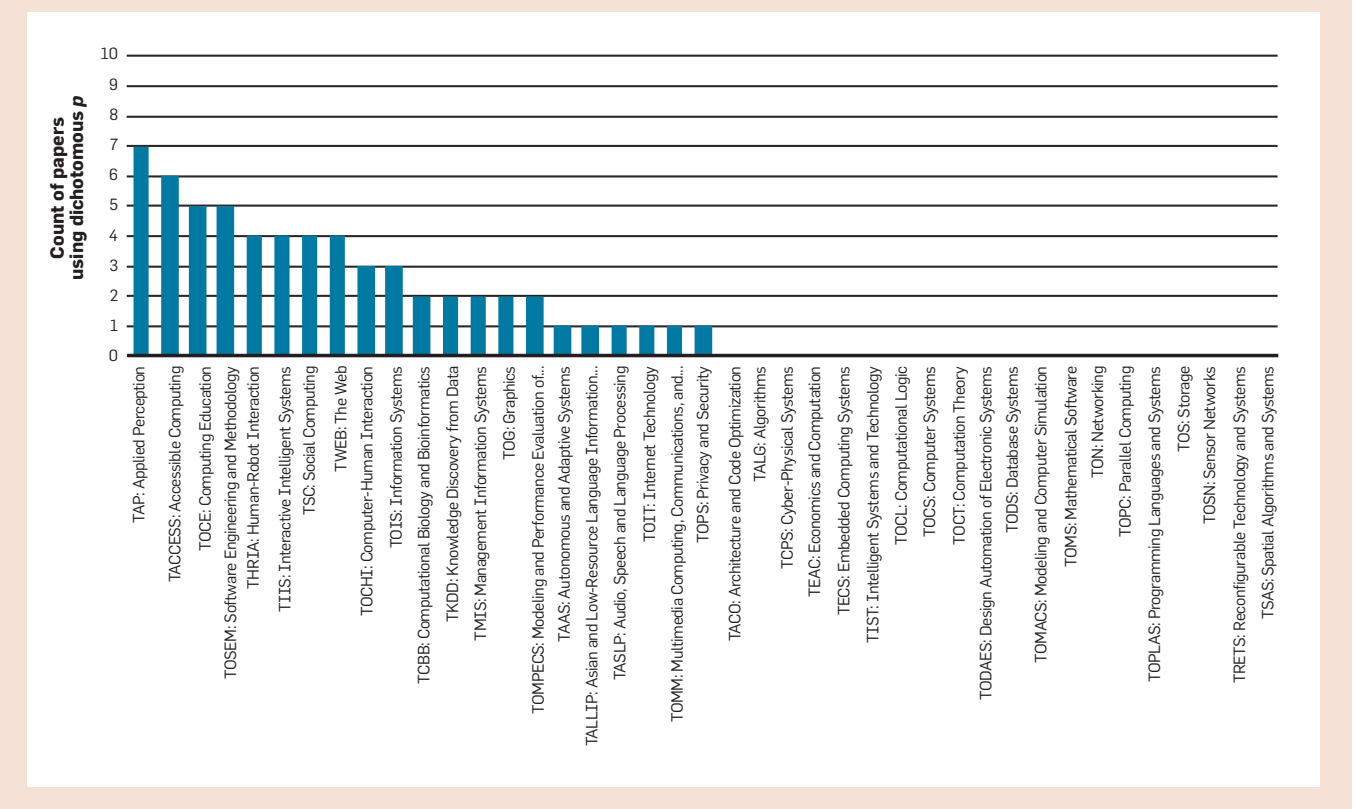
Publication bias encourages questionable data analysis practices. Dichotomous interpretation of NHST can also lead to problems in analysis: once experimental data has been collected, researchers may be tempted to explore a variety of post-hoc data analyses to make their findings look stronger or to reach statistical significance (Figure 1f). For example, they might consciously or unconsciously manipulate various

techniques such as excluding certain data points (for example, removing outliers, excluding participants, or narrowing the set of conditions under test), applying various transformations to the data, or applying statistical tests only to particular data subsets. While such analyses can be entirely appropriate if planned and reported in full, engaging in a data 'fishing' exercise to satisfy $p < .05$ is not, especially if the results are then selectively reported. Flexible data analysis and selective reporting can dramatically increase Type I error rates, and these are major culprits in the replication crisis.³⁸

Is Computer Science Research at Risk? (Spoiler: Yes)

Given that much of computer science research either does not involve experiments, or involves deterministic or large-sample computational experiments that are reproducible as long as data and code are made accessible, one could argue that the field is largely immune to replication issues that have plagued other empirical disciplines. To find out whether this argument is tenable, we analyzed the ten most down-

Figure 3. Count of articles from among the '10 most downloaded' (5/24/19) that use dichotomous interpretations of p from among ACM journals titled 'Transactions on...'



loaded articles for 41 ACM journals beginning with the name ‘*Transactions on.*’ We inspected all 410 articles to determine whether or not they used $p < \alpha$ (with α normally 0.05) as a criterion for establishing evidence of a difference between conditions. The presence of p -values is an indication of statistical uncertainty, and therefore of the use of nondeterministic small-sample experiments (for example involving human subjects). Furthermore, as we have previously discussed, the use of a dichotomous interpretation of p -values as ‘significant’ or ‘not significant’ is thought to promote publication bias and questionable data analysis practices, both of which heavily contributed to the replication crisis in other disciplines.

A total of 61 of the 410 computer science articles (15%) included at least one dichotomous interpretation of a p -value.^a All but two of the papers that used dichotomous interpretations (97%) identified at least one finding as satisfying the $p < .05$ criterion, suggesting that publication bias (long observed in other disciplines⁴¹) is likely to also exist in empirical computer science. Furthermore, 21 different journals (51%) included at least one article using a dichotomous interpretation of p within the set of 10 papers inspected. The count of articles across journals is summarized in Figure 3, with fields such as applied perception, education, software engineering, information systems, bioinformatics, performance modeling, and security all showing positive counts.

Our survey showed four main ways in which experimental techniques are used in computer science research, spanning work in graphics, software engineering, artificial intelligence, and performance analysis, as well as the expected use in human-computer interaction. First, empirical methods are used to assess the quality of an artifact produced by a technique, using humans as judges (for example, the photorealism of an image or the quality of streaming video). Second, empirical methods are used to evaluate classification or prediction algorithms on real-world data (for example, a power

scheduler for electric vehicles, using real data from smart meters). Third, they are used to carry out performance analysis of hardware or software, using actual data from running systems (for example, a comparison of real cloud computing platforms). Fourth, they are used to assess human performance with interfaces or interaction techniques (for example, which of two menu designs is faster).

Given the high proportion of computer science journals that accept papers using dichotomous interpretations of p , it seems unreasonable to believe that computer science research is immune to the problems that have contributed to a replication crisis in other disciplines. Next, we review proposals from other disciplines on how to ease the replication crisis, focusing first on changes to the way in which experimental data is analyzed, and second on proposals for improving openness and transparency.

Proposals for Easing the Crisis: Better Data Analysis

Redefine statistical significance. Many researchers attribute some of the replication crisis to the dominant use of NHST. Among the noted problems with NHST is the ease with which experiments can produce false-positive findings, even without scientists contributing to the problem through questionable research practices. To address this problem, a group of 75 senior scientists from diverse fields (including computer science) proposed that the accepted norm for determining ‘significance’ in NHST tests be reduced from $\alpha = .05$ to $\alpha = .005$.⁴ Their proposal was based on two analyses—the relationship between Bayes factors and p -values, and the influence of statistical power on false positive rates—both of which indicated disturbingly high false positive rates at $\alpha = .05$. The authors also recommended the word ‘suggestive’ be used to describe results in the range $.005 \leq p < .05$.

Despite the impressive list of authors, this proposal attracted heavy criticism (see Perezgonzalez³³ for a review). Some have argued the reasoning behind the .005 threshold is flawed, and that adopting it could actually make the replication crisis worse (by causing a drop in the statistical power of studies

without reducing incentives for p -hacking, and by diverting resources away from replications). Another argument is the threshold value remains arbitrary, and that focusing instead on effect sizes and their interval estimates (confidence intervals or credible intervals) can better characterize results. There is also a pragmatic problem that until publication venues firmly announce their standards, authors will be free to choose terminology (‘statistically significant’ at $p < .05$ or ‘statistically significant’ at $p < .005$) and reviewers/readers may differ in their expectations. Furthermore, the proposal does nothing to discourage or prevent problems associated with inappropriate modification of experimental methods and objectives after they begin.

Abandon statistical significance. Many researchers argue the replication crisis does not stem from the choice of the .05 cutoff, but from the general idea of using an arbitrary cutoff to classify results, in a dichotomous manner, as statistically significant or not. Some of these researchers have called for reporting exact p -values and abandoning the use of statistical significance thresholds.¹ Recently, a comment published in *Nature* with more than 800 signatories called for abandoning binary statistical significance.²⁸ Cumming¹² argued for the banning of p -values altogether and recommended the use of estimation statistics where strength of evidence is assessed in a non-dichotomous manner, by examining confidence intervals. Similar recommendations have been made in computer science.¹⁴ The editorial board of the *Basic and Applied Social Psychology* journal went further by announcing it would not publish papers containing any statistics that could be used to derive dichotomous interpretations, including p -values and confidence intervals.⁴² Overall there is no consensus on what should replace NHST, but many methodologists are in favor of banning dichotomous statistical significance language.

Despite the forceful language opposing NHST (for example, “very few defenses of NHST have been attempted,”¹²), some researchers believe NHST and the notion of dichotomous hypothesis testing still have their place.⁴ Others have suggested the calls

^a Data for this analysis is available at osf.io/hkqyt/, including a quote extracted from each counted paper showing its use of a dichotomous interpretation.

to abandon NHST are a red herring in the replicability crisis,³⁷ not least due to the lack of evidence that doing so will aid replicability.

Adopt Bayesian statistics. Several researchers propose replacing NHST with Bayesian statistical methods. One of the key motivators for doing so concerns a common misunderstanding of the p -value in NHST. Researchers wish to understand the probability the null hypothesis is true, given the data observed ($P(H_0|D)$), and p is often misunderstood to represent this value. However, the p -value actually represents the probability of observing data at least as extreme as the sample if the null hypothesis were true: ($P(D|H_0)$). In contrast to NHST, Bayesian statistics can enable the desired computation of ($P(H_0|D)$).

Bayesian statistics are perfectly suited for doing estimation statistics, and have several advantages over confidence intervals.^{22,26} Nevertheless, they can also be used to carry out dichotomous tests, possibly leading to the same issues as NHST. Furthermore, Bayesian analysis is not immune to the problems of p -hacking—researchers can still ‘b-hack’ to manipulate experimental evidence.^{37,39} In particular, the choice of priors adds an important additional experimenter degree of freedom in Bayesian analysis.³⁹

Help the reader form their own conclusion. Given the contention over the relative merits of different statistical methods and thresholds, researchers have proposed that when reporting results, authors should focus on assisting the reader in reaching their own conclusions by describing the data and the evidence as clearly as possible. This can be achieved through the use of carefully crafted charts that focus on effect sizes and their interval estimates, and the use of cautionary language in the author’s interpretations and conclusions.^{11,14}

While improved explanation and characterization of underlying experimental data is naturally desirable, authors are likely to encounter problems if relying only on the persuasiveness of their data. First, the impact of using more cautious language on the persuasiveness of arguments when compared to categorical arguments is still uncertain.¹⁵ Sec-

ond, many reviewers of empirical papers are familiar and comfortable with NHST procedures and its associated styles of results reporting, and they may criticize its absence; in particular, reviewers may suspect that the absence of reported dichotomous outcomes is a consequence of their failure to attain $p < .05$. Both of these concerns suggest that a paper’s acceptance prospects could be harmed if lacking simple and clear statements of results outcome, such as those provided by NHST, despite the simplistic and often misleading nature of such dichotomous statements.

Quantify p -hacking in published work. None of the proposals discussed here address problems connected with researchers consciously or subconsciously revising experimental methods, objectives, and analyses after their study has begun. Statistical analysis methods exist that allow researchers to assess whether a set of already published studies are likely to have involved such practices. A common method is based on the p -curve, which is the distribution of statistically significant p -values in a set of studies.⁴⁰ Studies of true effects should produce a right-skewed p -curve, with many lower statistically significant p -values (for example, .01s) than high values (for example, .04s); but a set of p -hacked studies are likely to show a left-skewed p -curve, indicative of selecting variables that tipped analyses into statistical significance.

While use of p -curves appears promising, it has several limitations. First, it requires a set of study results to establish a meaningful curve, and its use as a diagnostic tool for evidence of p -hacking in any single article is discouraged. Second, its usefulness for testing the veracity of any particular finding in a field depends on the availability of a series of related or replicated studies; but replications in computer science are rare. Third, statisticians have questioned the effectiveness of p -curves for detecting questionable research practices, demonstrating through simulations that p -curve methods cannot reliably distinguish between p -hacking of null effects and studies of true effects that suffer experimental omissions such as unknown confounds.⁷

Openness, Preregistration, and Registered Reports

While the debate continues over the merits of different methods for data analysis, there is a wide agreement on the need for improved openness and transparency in empirical science. This includes making materials, resources, and datasets available for future researchers who might wish to replicate the work.

Making materials and data available after a study’s completion is a substantial improvement, because it greatly facilitates peer scrutiny and replication. However, it does not prevent questionable research practices, since the history of a data analysis (including possible p -hacking) is not visible in the final analysis scripts. And if others fail to replicate a study’s findings, the original authors can easily explain away the inconsistencies by questioning the methodology of the new study or by claiming that an honest Type I error occurred.

Overcoming these limitations requires a clear statement of materials, methods, and hypotheses *before* the experiment is conducted, as provided by experimental preregistration and registered reports, discussed next.


Experimental preregistration. In response to concerns about questionable research practices, various authorities instituted registries in which researchers preregister their intentions, hypotheses, and methods (including sample sizes and precise plans for the data analyses) for upcoming experiments. Risks of p -hacking or outcome switching are dramatically reduced when a precise statement of method predates the experimental conduct. Furthermore, if the registry subsequently stores experimental data, then the file drawer is effectively opened on experimental outcomes that might otherwise have been hidden due to failure to attain statistical significance.

Although many think preregistration is only a recent idea, and therefore one that needs to be refined and tested before it can be fully adopted, it has in fact been in place for a long time in medical research. In 1997, the U.S. Food and Drug Administration Modernization Act (FDAMA) established the registry ClinicalTrials.gov, and over 96,000 experiments were registered in


its first 10 years, assisted by the decision of the International Committee of Medical Journal Editors to make preregistration a requirement for publication in their journals.³⁴ Results suggest that preregistration has had a substantial effect on scientific outcomes—for example, an analysis of studies funded by the National Heart, Lung, and Blood Institute between 1970 and 2012 showed the rate at which studies showed statistically significant findings plummeted from 57% before the introduction of mandatory preregistration (in 2000) to only 8% after.²¹ The success of ClinicalTrials.gov and the spread of the replication crisis to other disciplines has prompted many disciplines to introduce their own registries, including the American Economic Association (<https://www.socialscienceregistry.org/>) and the political science ‘dataverse.’²⁹ The Open Science Framework (OSF) also supports preregistration, ranging from simple and brief descriptions through to complete experimental specification (<http://osf.io>). Although originally focused on replications of psychological studies, it is now used in a range of disciplines, including by computer scientists.

Registered reports. While experimental preregistration should enhance confidence in published findings, it does not prevent reviewers from using statistical significance as a criterion for paper acceptance. Therefore, it does not solve the problem of publication bias and does not help prevent the file drawer effect. As a result, the scientific record can remain biased toward positive findings, and since achieving statistical significance is harder if *p*-hacking is not an option, researchers may be even more motivated to focus on unsurprising but safe hypotheses where the null is likely to be rejected. However, we do not want to simply take null results as equivalent to statistical significance, because null results are trivially easy to obtain; instead, the focus should be on the quality of the question being asked in the research.

Registered reports are a way to provide this focus. With registered reports, papers are submitted for review *prior* to conducting the experiment. Registered reports include the study motivation, related work, hypotheses, and detailed



With registered reports, papers are submitted for review *prior* to conducting the experiment.



method; everything that might be expected in a traditional paper *except* for the results and their interpretation. Submissions are therefore considered based on the study’s motivations (is this an interesting research question?) and method (is the way of answering the question sound and valid?). If accepted, a registered report is published *regardless* of the final results.

A recent analysis of 127 registered reports in the bio-medical and psychological sciences showed that 61% of studies did not support their hypothesis, compared to the estimated 5%–20% of null findings in the traditional literature.¹⁰ As of February 2019, the Center for Open Science (<https://cos.io/rr/>) lists 136 journals that accept registered reports and 27 journals that have accepted them as part of a special issue. No computer science journal is currently listed.

Recommendations for Computer Science

The use of NHST in relatively small-sample empirical studies is an important part of many areas of computer science, creating risks for our own reproducibility crisis.^{8,14,24} The following recommendations suggest activities and developments that computer scientists can work on to protect the credibility of the discipline’s empirical research.

Promote preregistration. The ACM has the opportunity and perhaps the obligation to lead and support changes that improve empirical computer science—its stated purpose includes ‘promotion of the highest standards’ and the ACM Publications Board has the goal of ‘aggressively developing the highest-quality content.’ These goals would be supported by propagating to journal editors and conference chairs an expectation that empirical studies should be preregistered, preferably using transdisciplinary registries such as the Open Science Framework (<http://osf.io>). Authors of papers describing empirical studies could be asked or required to include a standardized statement at the end of their papers’ abstract providing a link to the preregistration, or explicitly stating that the study was not preregistered (in other disciplines, preregistration is mandatory). Reviewers would also need to be educated on the value of


preregistration and the potential implications of its absence.

It is worth noting that experimental preregistration has potential benefits to authors even if they do not intend to test formal hypotheses. If the registry entry is accessible at the time of paper submission (perhaps through a key that is disclosed to reviewers), then an author who preregisters an exploratory experiment is protected against reviewer criticism that the stated exploratory intent is due to HARKing following a failure to reject the null hypothesis.⁸


Another important point regarding preregistration is that it does not constrain authors from reporting unexpected findings. Any analysis that might be used in an unregistered experiment could also be used in a preregistered one, but the language used to describe the analysis in the published paper must make the post-hoc discovery clear, such as ‘Contrary to expectations ...’ or ‘In addition to the preregistered analysis, we also ran ...’

Publish registered reports. The editorial boards of ACM journals that feature empirical studies could adapt their reviewing process to support the submission of registered reports and push for this publication format. This is perhaps the most promising of all interventions aimed at easing the replication crisis—it encourages researchers to address interesting questions, it eliminates the need to produce statistically significant results (and, thus, addresses the file drawer problem), and it encourages reviewers to focus on the work’s importance and potential validity.¹⁰ In addition, it eliminates hindsight bias among reviewers, that is, the sentiment that they could have predicted the outcomes of a study, and that the findings are therefore unsurprising.

The prospect of permitting the submission of registered reports to large-scale venues is daunting (for example, ACM 2019 Conference on Human-Computer Interaction received approximately 3,000 submissions to its papers track). However, the two-round submission and review process adopted by conferences within the *Proceedings of the ACM* (PACM) series could be adapted to embrace the submission of registered reports at round 1. We encour-



Experimental preregistration has potential benefits to authors even if they do not intend to test formal hypotheses.



age conference chairs to experiment with registered report submissions.

Encourage data and materials openness. The ACM Digital Library supports access to resources that could aid replication through links to auxiliary materials. However, more could be done to encourage or require authors to make data and resources available. Currently, authors decide whether or not to upload resources. Instead, uploading data could be compulsory for publication, with exceptions made only following special permission from an editor or program chair. While such requirements may seem draconian given the permissive nature of current practice in computer science, the requirement is common in other disciplines and outlets, such as *Nature’s* ‘Scientific Data’ (www.nature.com/sdata/).

A first step in this direction would be to follow *transparency and openness guidelines* (<https://cos.io/our-services/top-guidelines/>), which encourage authors to state in their submission whether or not they made their data, scripts, and preregistered analysis available online, and to provide links to them where available.

Promote clear reporting of results. While the debate over standards for data analysis and reporting continues, certain best-practice guidelines are emerging. First, authors should focus on two issues: conveying effect sizes (this includes simple effect sizes such as differences between means¹¹), and helping readers to understand the uncertainty around those effect sizes by reporting interval estimates^{14,26} or posterior distributions.²² A range of recommendations already exist for improving reporting clarity and transparency and must be followed more widely. For example, most effect sizes only capture central tendencies and thus provide an incomplete picture. Therefore, it can help to also convey population variability through well-known practices such as reporting standard deviations (and their interval estimates) and/or plotting data distributions. When reporting the outcomes of statistical tests, the name of the test and its associated key data (such as degrees of freedom) should be reported. And, if describing the outcomes of a NHST test, the exact *p*-value should be reported. Since the probability of a successful replication de-

depends on the order of magnitude of p ,¹⁷ we suggest avoiding excessive precision (one or two significant digits are enough), and using scientific notation (for example, $p = 2 \times 10^{-5}$) instead of inequalities (for example, $p < .001$) when reporting very small p -values.

Encourage replications. The introduction of preregistration and registered reports in other disciplines caused a rapid decrease in the proportion of studies finding statistically significant effects. Assuming the same was to occur in computer science, how would this influence accepted publications? It is likely that many more empirical studies would be published with statistically non-significant findings or with no statistical analysis (such as exploratory studies that rely on qualitative methods). It is also likely that this would encourage researchers to consider conducting experimental replications, regardless of previous outcomes. Replications of studies with statistically significant results help reduce Type I error rates, and replications of studies with null outcomes reduce Type II error rates and can test the boundaries of hypotheses. If better data repositories were available, computer science students around the world could contribute to the robustness of findings by uploading to registries the outcomes of replications conducted as part of their courses on experimental methods. Better data repositories with richer datasets would also facilitate meta-analyses, which elevate confidence in findings beyond that possible from a single study.

Educate reviewers (and authors). Many major publication venues in computer science are under stress due to a deluge of submissions that creates challenges in obtaining expert reviews. Authors can become frustrated when reviewers focus on equivocal results of a well-founded and potentially important study—but reviewers can also become frustrated when authors fail to provide definitive findings on which to establish a clear contribution. In the spirit of registered reports, our recommendation is to educate reviewers (and authors) on the research value of studying interesting and important effects, largely irrespective of the results generated. If reviewers focused on questions and method rather

than traditional evidentiary criteria such as $p < .05$, then researchers would be better motivated to identify interesting research questions, including potentially risky ones. One potential objection to risky studies is their typically low statistical power: testing null effects or very small effects with small samples can lead to vast overestimations of effect sizes.²⁷ However, this is mostly true in the presence of p -hacking or publication bias, two issues that are eliminated by moving beyond the statistical significance filter and adopting registered reports. **C**

References

1. Amrhein, V., Korner-Nievergelt, F., and Roth, T. The earth is flat ($p > 0.05$): significance thresholds and the crisis of unreplicable research. *PeerJ* 5, 7 (2017), e3544.
2. Baker, M. Is there a reproducibility crisis? *Nature* 533, 7604 (2016), 452–454.
3. Begley, C. G., and Ellis, L. M. Raise standards for preclinical cancer research. *Nature* 483, 7391 (2012), 531.
4. Benjamin, D. et al. Redefine statistical significance. *PsyArXiv* (July 22, 2017).
5. Boisvert, R.F. Incentivizing reproducibility. *Commun. ACM* 59, 10 (Oct. 2016), 5–5.
6. Boyd, D., and Crawford, K. Critical questions for big data. *Information, Communication & Society* 15, 5 (2012), 662–679.
7. Bruns, S.B., and Ioannidis, J.P.A. P -curve and p -hacking in observational research. *PLOS One* 11, 2 (Feb. 2016), 1–13.
8. Cockburn, A., Gutwin, C., and Dix, A. HARK no more: On the preregistration of CHI experiments. In *Proceedings of the 2018 ACM CHI Conference on Human Factors in Computing Systems* (Montreal, Canada, Apr. 2018), 141:1–141:12.
9. Collberg, C., and Proebsting, T. A. Repeatability in computer systems research. *Commun. ACM* 59, 3 (Mar. 2016), 62–69.
10. Cristea, I.A., and Ioannidis, J.P.A. P -values in display items are ubiquitous and almost invariably significant: A survey of top science journals. *PLOS One* 13, 5 (2018), e0197440.
11. Cumming, G. *Understanding the New Statistics: Effect Sizes, Confidence Intervals, and Meta-analysis*. Multivariate applications series. Routledge, 2012.
12. Cumming, G. The new statistics: Why and how. *Psychological Science* 25, 1 (2014), 7–29.
13. Denning, P. J. ACM President's letter: What is experimental computer science? *Commun. ACM* 23, 10 (Oct. 1980), 543–544.
14. Dragicevic, P. Fair statistical communication in HCI. *Modern Statistical Methods for HCI*, J. Robertson and M. Kaptein, eds. Springer International Publishing, 2016, 291–330.
15. Durik, A.M., Britt, M.A., Reynolds, R., and Storey, J. The effects of hedges in persuasive arguments: A nuanced analysis of language. *J. Language and Social Psychology* 27, 3 (2008), 217–234.
16. Franco, A., Malhotra, N., and Simonovits, G. Publication bias in the social sciences: Unlocking the file drawer. *Science* 345, 6203 (2014), 1502–1505.
17. Goodman, S.N. A comment on replication, p -values and evidence. *Statistics in Medicine* 11, 7 (1992), 875–879.
18. Gundersen, O.E., and Kjensmo, S. State of the art: Reproducibility in artificial intelligence. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence, the 30th Innovative Applications of Artificial Intelligence, and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence*. New Orleans, LA, USA, Feb. 2–7, 2018), 1644–1651.
19. Ioannidis, J.P.A. Why most published research findings are false. *PLOS Medicine* 2, 8 (Aug. 2005).
20. John, L.K., Loewenstein, G., and Prelec, D. Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science* 23, 5 (2012), 524–532. PMID: 22508865.
21. Kaplan, R.M., and Irvin, V.L. Likelihood of null effects

of large NHLBI clinical trials has increased over time. *PLOS One* 10, 8 (Aug. 2015), 1–12.

22. Kay, M., Nelson, G.L., and Hekler, E.B. Researcher-centered design of statistics: Why Bayesian statistics better fit the culture and incentives of HCI. In *Proceedings of the 2016 ACM CHI Conference on Human Factors in Computing Systems*, 4521–4532.
23. Kerr, N.L. Harking: Hypothesizing after the results are known. *Personality & Social Psychology Rev.* 2, 3 (1998), 196. Lawrence Erlbaum Assoc.
24. Kosara, R., and Haroz, S. Skipping the replication crisis in visualization: Threats to study validity and how to address them. *Evaluation and Beyond—Methodological Approaches for Visualization* (Berlin, Germany, Oct. 2018).
25. Krishnamurthi, S., and Vitek, J. The real software crisis: Repeatability as a core value. *Commun. ACM* 58, 3 (Mar. 2015), 34–36.
26. Kruschke, J.K., and Liddell, T.M. The Bayesian new statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a Bayesian perspective. *Psychonomic Bulletin & Rev.* 25, 1 (2018), 178–206.
27. Loken, E., and Gelman, A. Measurement error and the replication crisis. *Science* 355, 6325 (2017), 584–585.
28. McShane, B., Gal, D., Gelman, A., Robert, C., and Tackett, J.L. Abandon statistical significance. *The American Statistician* 73, sup1 (2019), 235–245.
29. Monogan, III, J.E. A case for registering studies of political outcomes: An application in the 2010 house elections. *Political Analysis* 21, 1 (2013), 21.
30. Nickerson, R.S. Confirmation bias: A ubiquitous phenomenon in many guises. *Rev. General Psychology* 2, 2 (1998), 175–220.
31. Open Science Collaboration and others. Estimating the reproducibility of psychological science. *Science* 349, 6251 (2015), aac4716.
32. Pashler, H., and Wagenmakers, E.-J. Editors' introduction to the special section on replicability in psychological science: A crisis of confidence? *Perspectives on Psychological Science* 7, 6 (2012), 528–530.
33. Perezgonzalez, J.D., and Frias-Navarro, D. Retract 0.005 and propose using JASP, instead. Preprint, 2017; <https://psyarxiv.com/t2fn8>.
34. Rennie, D. Trial registration: A great idea switches from ignored to irresistible. *JAMA* 292, 11 (2004), 1359–1362.
35. Rosenthal, R. The file drawer problem and tolerance for null results. *Psychological Bulletin* 86, 3 (1979), 638–641.
36. Sanbonmatsu, D.M., Posavac, S.S., Kardes, F.R. and Mantel, S.P. Selective hypothesis testing. *Psychonomic Bulletin & Rev.* 5, 2 (June 1998), 197–220.
37. Savalei, V., and Dunn, E. Is the call to abandon p -values the red herring of the replicability crisis? *Frontiers in Psychology* 6 (2015), 245.
38. Simmons, J.P., Nelson, L.D., and Simonsohn, U. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science* 22, 11 (2011), 1359–1366.
39. Simonsohn, U. Posterior-hacking: Selective reporting invalidates Bayesian results also. *SSRN* (2014).
40. Simonsohn, U., Nelson, L.D., and Simmons, J.P. P -curve: A key to the file-drawer. *J. Experimental Psychology: General* 143, 2 (2014), 534–547.
41. Sterling, T.D. Publication decisions and their possible effects on inferences drawn from tests of significance—or vice versa. *J. American Statistical Assoc.* 54, 285 (1959), 30–34.
42. Trafimow, D., and Marks, M. Editorial. *Basic and Applied Social Psychology* 37, 1 (2015), 1–2.

Andy Cockburn (andy.cockburn@canterbury.ac.nz) is a professor at the University of Canterbury, Christchurch, New Zealand, where he is head of the HCI and Multimedia Lab.

Pierre Dragicevic is a research scientist at Inria, Orsay, France.

Lonni Besançon is a postdoc student at Linköping University, Linköping, Sweden

Carl Gutwin is a professor in the Department of Computer Science and director of the HCI Lab at the University of Saskatchewan, Canada.

The Handbook of Multimodal-Multisensor Interfaces, Volume 3

*Language Processing, Software,
Commercialization, and Emerging Directions*

This third volume of **The Handbook of Multimodal-Multisensor Interfaces** focuses on state-of-the-art multimodal language and dialogue processing, including semantic integration of modalities. The development of increasingly expressive embodied agents and robots has become an active test-bed for coordinating multimodal dialogue input and output, including processing of language and nonverbal communication. In addition, major application areas are featured for commercializing multimodal-multisensor systems, including automotive, robotic, manufacturing, machine translation, banking, communications, and others. These systems rely heavily on software tools, data resources, and international standards to facilitate their development. For insights into the future, emerging multimodal-multisensor technology trends are highlighted for medicine, robotics, interaction with smart spaces, and similar topics. Finally, this volume discusses the societal impact of more widespread adoption of these systems, such as privacy risks and how to mitigate them. The handbook chapters provide a number of walk-through examples of system design and processing, information on practical resources for developing and evaluating new systems, and terminology and tutorial support for mastering this emerging field. In the final section of this volume, experts exchange views on a timely and controversial challenge topic, and how they believe multimodal-multisensor interfaces need to be equipped to most effectively advance human performance during the next decade.

Edited by Sharon Oviatt et al

ISBN: 978-1-970001-72-3

DOI: 10.1145/3233795

<http://books.acm.org>

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



ACM BOOKS
Collection I

P. 82

**Technical
Perspective
Entity Matching
with Magellan**

By Wang-Chiew Tan

P. 83

**Magellan: Toward
Building Ecosystems of
Entity Matching Solutions**

By AnHai Doan, Pradap Konda, Paul Suganthan G.C.,
Yash Govind, Derek Paulsen, Kaushik Chandrasekhar,
Philip Martinkus, and Matthew Christie

P. 92

**Technical
Perspective
Supporting Linear
Algebra Operations
in SQL**

By Yannis Papakonstantinou

P. 93

**Scalable Linear Algebra on
a Relational Database System**

By Shangyu Luo, Zekai J. Gao, Michael Gubanov,
Luis L. Perez, Dimitrije Jankov, and Christopher Jermaine

Technical Perspective

Entity Matching with Magellan

By Wang-Chiew Tan

FERDINAND MAGELLAN WAS a Portuguese explorer who launched a Spanish expedition that completed the first circumnavigation of the Earth. It is in this spirit that Magellan was used as the name of the end-to-end entity matching system that is developed at the University of Wisconsin.

Entity matching (also known as *entity resolution* or *reference reconciliation* or *deduplication*) is a major task in the larger problem of data integration, a problem that is pervasive in many organizations. Despite being a subject of extensive research for many years, the entity matching problem is surprisingly simple to describe and understand. It is to determine whether two different representations refer to the same real-world entity. For example, whether the two tuples—(*J. Doe, UWisc*) and (*John Doe, Univ. of Wisconsin*)—refer to the same person.

Perhaps more surprisingly, most prior systems for entity matching are stand-alone systems, sometimes built for specific applications, and are difficult to interoperate in the larger data integration setting, which often involves a composition of various other tasks such as data acquisition, preparation, transformation, cleaning, and schema matching, in addition to entity matching. For example, the two tuples above may be the result of data extracted from acquired pdfs or text files and transformed into the format above before they are matched. Different tasks need different libraries and techniques and they must interoperate before an end-to-end entity matching or data integration pipeline can be successfully executed. Magellan is able to provide all of the above.


Magellan's key insight is that a successful entity matching system must offer a versatile system building paradigm for entity matching that can be easily adapted for different application needs. Furthermore, it must also be easy to “plug-and-play” entity

matching into data integration pipelines or other systems. There already exist vibrant ecosystems of data science libraries and tools (for example, those in Python and R), which are heavily used by data scientists to solve many data integration tasks. By developing entity matching tools within such ecosystems, Magellan makes it easy for data scientists to exploit the tools (including Magellan) in the ecosystems and in turn, make such ecosystems better at solving various data integration problems. In sum, Magellan distinguishes itself by making it easy to develop entity matching tools that incorporates advanced entity matching techniques. In addition, it allows researchers to “connect” and exploit the vast ecosystems of data science tools and build entity matching tools directly into those ecosystems.

In Magellan, there are two entity matching tools developed for two widely used execution environments: (1) PyMatcher is an entity matching tool that is developed as part of the PyData ecosystem. This allows users to leverage the rich set of Python libraries to carry out the entire entity matching pipeline, which may involve subtasks such as data cleaning, visualization, in addition to blocking and matching. (2) CloudMatcher is a cloud-based entity matching tool that is part of the Amazon Web Ser-

vices ecosystem. PyMatcher is intended for a “power user” who possess knowledge about entity matching, programming, and basic machine learning while CloudMatcher is targeted for “lay users” who may not know how to program or possess machine learning knowledge.

PyMatcher provides how-to guides that describe how to approach the development of entity matching workflows. These guides describe how to develop a solution for a small sample of data (by *downsampling*, *blocking*, and *training a matcher*) and how to scale the solution to work with production data. The entity matching workflow for CloudMatcher is similar to that of PyMatcher except that CloudMatcher actively learns from the user how to block tuples. Afterwards, it executes the blocking rules that are learnt to obtain a set of candidate pairs of tuples and again actively learns from the users what are the (non-)matching candidate pairs of tuples before deriving a model that can be applied to match tuples across two tables.

In short, Magellan makes it easy to develop an entity matching solution and easy to interoperate with other tools to form a bigger data integration pipeline that solves larger problems. It is a showcase for practical software development tools that originate from data management research. It has been successfully applied to multiple entity matching problems in the real world, is used in production at many data science groups and companies, and is recently being commercialized, demonstrating that using data science ideas to build entity matching systems is highly promising. For more details, check out Magellan's website at <https://sites.google.com/site/anhaidgroup/projects/magellan>. 

**Magellan
makes it easy
to develop
an entity
matching solution.**

Wang-Chiew Tan is Director of Research at Megagon Labs, Mountain View, CA, USA.

Copyright held by author.

Magellan: Toward Building Ecosystems of Entity Matching Solutions

By AnHai Doan, Pradap Konda, Paul Suganthan G.C., Yash Govind, Derek Paulsen, Kaushik Chandrasekhar, Philip Martinkus, and Matthew Christie*

Abstract

Entity matching (EM) finds data instances that refer to the same real-world entity. In 2015, we started the Magellan project at UW-Madison, jointly with industrial partners, to build EM systems. Most current EM systems are stand-alone monoliths. In contrast, Magellan borrows ideas from the field of data science (DS), to build a new kind of EM systems, which is ecosystems of interoperable tools for multiple execution environments, such as on-premise, cloud, and mobile. This paper describes Magellan, focusing on the system aspects. We argue why EM can be viewed as a special class of DS problems and thus can benefit from system building ideas in DS. We discuss how these ideas have been adapted to build PyMatcher and CloudMatcher, sophisticated on-premise tools for power users and self-service cloud tools for lay users. These tools exploit techniques from the fields of machine learning, big data scaling, efficient user interaction, databases, and cloud systems. They have been successfully used in 13 companies and domain science groups, and have been pushed into production for many customers, and are being commercialized. We discuss the lessons learned and explore applying the Magellan template to other tasks in data exploration, cleaning, and integration.

1. INTRODUCTION

Entity matching (EM) finds data instances that refer to the same real-world entity, such as tuples (David Smith, UW-Madison) and (D. Smith, UWM). This problem, also known as entity resolution, record linkage, deduplication, data matching, et cetera, has been a long-standing challenge in the database, AI, KDD, and Web communities.^{2,6}

As data-driven applications proliferate, EM will become even more important. For example, to analyze raw data for insights, we often integrate multiple raw data sets into a single unified one, before performing the analysis, and such integration often requires EM. To build a knowledge

graph, we often start with a small graph and then expand it with new data sets, and such expansion requires EM. When managing a data lake, we often use EM to establish semantic linkages among the disparate data sets in the lake.

Given the growing importance of EM, in the summer of 2015, together with industrial partners, we started the Magellan project at the University of Wisconsin-Madison, to develop EM solutions.⁹ Numerous works have studied EM, but most of them develop *EM algorithms* for isolated steps in the EM workflow. In contrast, we seek to build *EM systems*, as we believe such systems are critical for advancing the EM field. Among others, they help evaluate EM algorithms, integrate R&D efforts, and make practical impacts, the same way systems such as System R, Ingres, Apache Hadoop, and Apache Spark have helped advance the fields of relational database management systems (RDBMSs) and Big Data.

Of course, Magellan is not the first project to build EM systems. Many such systems have been developed.^{9,2} However, as far as we can tell, virtually all of them have been built as *stand-alone monolithic EM systems*, or parts of larger monolithic systems that perform data cleaning and integration.^{2,6,9} These systems often employ the RDBMS building template. That is, given an EM workflow composed of logical operators (specified declaratively or via a GUI by a user), they compile this workflow into one consisting of physical operators and then optimize and execute the compiled workflow.

In contrast, Magellan develops a radically different system building template for EM, by leveraging ideas from the field of data science (DS). Although DS is still “young,” several common themes have emerged.

- For many DS tasks, there is a general consensus that it is not possible to fully automate the two stages of developing and productionizing DS workflows. So users must “be in the loop,” and many step-by-step guides that tell users how to execute the above two stages have been developed.
- Many “pain points” in these guides, that is, steps that are time-consuming for users, have been identified, and (semi)-automated tools have been developed to reduce user effort.

* Additional authors are Sanjib Das (Google), Erik Paulson (Johnson Control), Palaniappan Nagarajan (Amazon), Han Li (UW-Madison), Sidharth Mudgal (Amazon), Aravind Soundararajan (Amazon), Jeffrey R. Ballard (UW-Madison), Haojun Zhang (UW-Madison), Adel Ardalan (Columbia Univ.), Amanpreet Saini (UW-Madison), Mohammed Danish Shaikh (UW-Madison), Youngchoon Park (Johnson Control), Marshall Carter (American Family Ins.), Mingju Sun (American Family Ins.), Glenn M. Fung (American Family Ins.), Ganesh Krishnan (WalmartLabs), Rohit Deep (WalmartLabs), Vijay Raghavendra (WalmartLabs), Jeffrey F. Naughton (Google), Shishir Prasad (Instacart), and Fatemah Panahi (Google).

The original version of this paper is entitled “Entity Matching Meets Data Science: A Progress Report from the Magellan Project” and was published in *Proceedings of the 2019 SIGMOD Conference*.

- Users often use multiple execution environments (EE), such as on-premise, cloud, and mobile, switching among them. So tools have been developed for all of these EEs.
- Finally, within each EE, tools have been designed to be atomic and interoperable, forming a growing ecosystem of DS tools. Examples include PyData, the ecosystem of 184,000+ interoperable Python packages (as of June 2019), R, tidyverse, and many others.⁴

We observed that EM bears strong similarities to many DS tasks.⁹ As a result, we leveraged the above ideas to build a new kind of EM systems. Specifically, we develop guides that tell users how to perform EM step by step, identify the “pain points” in the guides, and then develop tools to address these pain points. We develop tools for multiple execution environments (EEs), such that within each EE, tools interoperate and build upon existing DS tools in that EE.

Thus, the notion of “system” in Magellan has changed. It is no longer a stand-alone monolithic system such as RDBMSs or most current EM systems. Instead, this new “system” spans multiple EEs. Within each EE, it provides a growing ecosystem of interoperable EM tools, situated in a larger ecosystem of DS tools. Finally, it provides detailed guides that tell users how to use these tools to perform EM.

Since the summer of 2015, we have pursued the above EM agenda and developed small ecosystems of EM tools for on-premise and cloud EEs. These tools exploit techniques from the fields of machine learning, big data scaling, efficient user interaction, databases, and cloud systems. They have been successfully used in 13 companies and domain science groups, have been pushed into production for many customers, and are being commercialized. Developing them has also raised many research challenges.⁴

In this paper, we describe the above progress, focusing on the system aspects. The next section discusses the EM problem and related work. Section 3 discusses the main system building themes of data science and the Magellan agenda. Sections 4–5 discuss PyMatcher and CloudMatcher, two current thrusts of Magellan. Section 6 discusses the application of Magellan tools to real-world EM problems. Section 7 discusses lessons learned and ongoing work. Section 8 concludes by exploring how to apply the Magellan template to other tasks in data exploration, cleaning, and integration. More information about Magellan can be found at sites.google.com/site/anhaidgroup/projects/magellan.

2. THE ENTITY MATCHING PROBLEM

Entity matching, also known as entity resolution, record linkage, data matching, et cetera., has received enormous attention.^{2, 6, 5, 13} A common EM scenario finds all tuple pairs that match, that is, refer to the same real-world entity, between two tables *A* and *B* (see Figure 1). Other EM scenarios include matching tuples within a single table, matching into a knowledge graph, matching XML data, et cetera.²

When matching two tables *A* and *B*, considering all pairs in $A \times B$ often takes very long. So users often execute a blocking step followed by a matching step.² *The blocking step* employs heuristics to quickly remove obviously nonmatched

Figure 1. An example of matching two tables.

Table A			Table B			Matches		
Name	City	State	Name	City	State			
a_1	Dave Smith	Madison	WI	b_1	David D. Smith	Madison	WI	(a_1, b_1)
a_2	Joe Wilson	San Jose	CA	b_2	Daniel W. Smith	Middleton	WI	(a_3, b_2)
a_3	Dan Smith	Middleton	WI					

tuple pairs (e.g., persons residing in different states). *The matching step* applies a matcher to the remaining pairs to predict matches.

The vast body of work in EM falls roughly into three groups: algorithmic, human-centric, and system. Most EM works develop *algorithmic solutions* for blocking and matching, exploiting rules, learning, clustering, crowdsourcing, external data, et cetera.^{2, 6, 5} The focus is on improving accuracy, minimizing runtime, and minimizing cost (e.g., crowdsourcing fee), among others.^{13, 6}

A smaller but growing body of EM work (e.g., HILDA¹) studies *human-centric* challenges, such as crowdsourcing, effective user interaction, and user behavior during the EM process.

The third group of EM work develops *EM systems*. In 2016, we surveyed 18 noncommercial systems (e.g., D-Dupe, Febrl, Dedoop, and Nadeef) and 15 commercial ones (e.g., Tamr, Informatica, and IBM InfoSphere).^{9, 12} Most of these systems are *stand-alone monoliths, built using the RDBMS template*. Specifically, such a system has a set of logical operations (e.g., blocking and matching) with multiple physical implementations. Given an EM workflow (composing of these operations) specified by the user using a GUI or a declarative language, the system translates the workflow into an execution plan and then optimizes and executes this plan.

3. THE MAGELLAN AGENDA

We now discuss system building ideas in the field of data science (DS). Then we argue that EM is very similar in nature to DS and thus can benefit from these ideas. Finally, we suggest a system building agenda for Magellan.

System Building Ideas of Data Science: Although the DS field has been growing rapidly, we are not aware of any explicit description of its “system template.” But our examination reveals the following important ideas.

First, many DS tasks distinguish between two stages, development and production, as these stages raise different challenges. The development stage finds an accurate DS workflow, often using data samples. This raises challenges in data exploration, profiling, understanding, cleaning, model fitting and evaluation, et cetera. The production (a.k.a. deployment) stage executes the discovered DS workflow on the entirety of data, raising challenges in scaling, logging, crash recovery, monitoring, et cetera.

Second, DS developers do not assume that the above two stages can be automated. Users often must be “in the loop” and often do not know what to do, how to start, et cetera. As a result, developers provide detailed guides that tell users how to solve a DS problem, step by step. Numerous guides have

been developed, described in books, papers, Jupyter notebooks, training camps, blogs, tutorials, et cetera.

It is important to note that such a guide is *not* a user manual on how to use a tool. Rather, it is a step-by-step instruction to the user on how to start, when to use which tools, and when to do what manually, in order to solve the DS task end to end. Put differently, it is an (often complex) algorithm for the user to follow. (See Section 4 for an example.)

Third, even without tools, users should be able to follow a guide and manually execute all the steps to solve a DS task. But some of the steps can be very time-consuming. DS developers have identified such “pain point” steps and developed (semi-)automatic tools to reduce the human effort.

Fourth, these tools target not just power users, but also lay users (as such users increasingly also need to work on the data), and use a variety of techniques, for example, machine learning (ML), RDBMS, visualization, effective user interaction, Big Data scaling, and cloud technologies.

Fifth, it is generally agreed that users will often use multiple execution environments (EEs), such as on-premise, cloud, and mobile, switching among these EEs as appropriate, to execute a DS task. As a result, tools have been developed for all of these EEs.

Finally, within each EE, tools have been designed to be atomic (i.e., each tool does just one thing) and interoperable, forming a growing ecosystem of DS tools. Popular examples of such ecosystems include PyData, R, tidyverse, and many others.⁴

The Similarities between EM and DS: We argue that EM bears strong similarities to many DS tasks. EM often shares the same two stages: development, where users find an accurate EM workflow using data samples, and production, where users execute the workflow on the entirety of data (see Section 4 for an example).

The above two EM stages raise challenges that are remarkably similar to those of DS tasks, for example, data understanding, model fitting, scaling, et cetera. Moreover, there is also an emerging consensus that it is not possible to fully automate the above two stages for EM. Similar to DS, this also raises the need for step-by-step guides that tell users how to be “in the loop,” as well as the need for identifying “pain points” in the guide and developing tools for these pain points (to reduce user effort). Finally, these tools also have to target both power and lay users, and use a variety of techniques, for example, ML, RDBMS, visualization, scaling, et cetera.

Thus, we believe EM can be viewed as a special class of DS problems, which focuses on finding the semantic matches, for example, “(David Smith, UWM) = (D. Smith, UW-Madison).” As such, we believe EM can benefit from the system building ideas in DS.

Our Agenda: Using the above “system template” of DS, we developed the following agenda for Magellan. First, we identify common EM scenarios. Next, we develop how-to guides to solve these scenarios end to end, paying special attention to telling the user exactly what to do. Then we identify the pain points in the guides and develop (semi-)automatic tools to reduce user effort. We design tools to be atomic and interoperable, as a part of a growing ecosystem of DS tools.

Developing these tools raises research challenges, which we address. Finally, we work with users (e.g., domain scientists, companies, and students) to evaluate our EM tools.

In the past few years, we have been developing EM tools for two popular execution environments: on-premise and cloud. Specifically, PyMatcher is a small ecosystem of on-premise EM tools for power users, built as a part of the PyData ecosystem of DS tools, and CloudMatcher is a small ecosystem of cloud EM tools for lay users, built as a part of the AWS ecosystem of DS tools. The next two sections briefly describe these ecosystems.

4. PYMATCHER

We now describe PyMatcher, an EM system developed for power users in the on-premise execution environment.

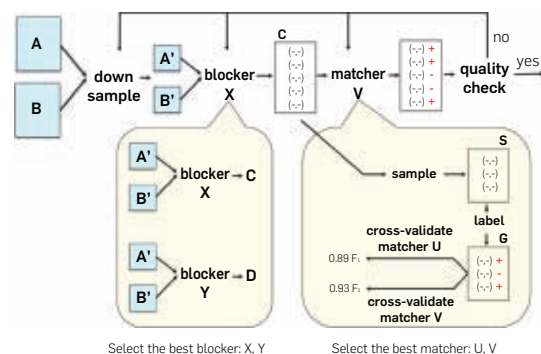
Problem Scenarios: In this first thrust of Magellan, we consider an EM scenario that commonly occurs in practice, where a user U wants to match two tables (e.g., see Figure 1), with as high matching accuracy as possible, or with accuracy exceeding a threshold. U is a “power user” who knows programming, EM, and ML.

Developing How-to Guide: We developed an initial guide based on our experience and then kept refining it based on user feedback and on watching how real users do EM. As of Nov 2018, we have developed a guide for the above EM scenario, which consists of two smaller guides for the development and production stages, respectively. Here, we focus on the guide for the development stage (briefly discussing the guide for the production stage at the end of this section).

This guide (which is illustrated in Figure 2) heavily uses ML. To explain it, suppose user U wants to match two tables A and B , each having 1 million tuples. Trying to find an accurate workflow using these two tables would be too time-consuming, because they are too big. Hence, U will first “down sample” the two tables to obtain two smaller tables A' and B' , each having 100K tuples, say (see the figure).

Next, suppose the EM system provides two blockers X and Y . Then, U experiments with these blockers (e.g., executing both on Tables A' and B' and examining their output) to select the blocker judged the best (according to some criterion). Suppose U selects blocker X . Then next, U executes X on Tables A' and B' to obtain a set of candidate tuple pairs C .

Figure 2. The steps of the guide for the development stage of PyMatcher.



Next, U takes a sample S from C and labels the pairs in S as “match”/“no-match” (see the figure). Let the labeled set be G , and suppose the EM system provides two learning-based matchers U and V (e.g., decision trees and logistic regression). Then, U uses the labeled set G to perform cross validation for U and V . Suppose V produces higher matching accuracy (such as F_1 score of 0.93, see the figure). Then, U selects V as the matcher and applies V to the set C to predict “match”/“no-match,” shown as “+” or “-” in the figure. Finally, U may perform quality check (by examining a sample of the predictions and computing the resulting accuracy) and then go back and debug and modify the previous steps as appropriate. This continues until U is satisfied with the accuracy of the EM workflow.

Developing Tools for the Steps of the Guide: Over the past 3.5 years, 13 developers have developed tools for the steps of the above guide (see Govind et al.⁸). As of September 2019, PyMatcher consists of 6 Python packages with 37K lines of code and 231 commands (and is open sourced⁹). It is built on top of 16 different packages in the PyData ecosystem (e.g., pandas and scikit-learn). As far as we can tell, PyMatcher is the most comprehensive open-source EM system today, in terms of the number of features it supports.

Principles for Developing Tools & Packages: In PyMatcher, each tool is roughly equivalent to a Python command, and tools are organized into Python packages. We adopted five principles for developing tools and packages:

1. They should *interoperate* with one another, and with existing PyData packages.
2. They should be *atomic*, that is, each does only one thing.
3. They should be *self-contained*, that is, they can be used by themselves, not relying on anything outside.
4. They should be *customizable*.
5. They should be *efficient* for both humans and machines.

We now illustrate these principles. As an example of facilitating *interoperability* among the commands of different packages, we use only generic well-known data structures such as Pandas DataFrame to hold tables (e.g., the two tables A and B to match and the output table after blocking).

Designing each command, that is, tool, to be “atomic” is somewhat straightforward. Designing each package to be so is more difficult. Initially, we designed just one package for all tools of all steps of the guide. Then, as soon as it was obvious that a set of tools form a coherent stand-alone group, we extracted it as a new package. However, this extraction is not always easy to do, as we will discuss soon.

Ignoring self-containment for now, to make tools and packages highly *customizable*, we expose all possible “knobs” for the user to tweak and provide easy ways for him/her to do so. For example, given two tables A and B to match, PyMatcher can automatically define a set of features (e.g., $jaccard(3gram(A.name), 3gram(B.name))$). We store this set of features in a global variable F . We give users ways to delete features from F and to declaratively define more features and then add them to F .

As an example of making a tool, that is, a command, X *efficient for a user*, we can make X easy to remember and specify (i.e., it does not require the user to enter many arguments). Often, this also means that we provide multiple variations for X , because each user may best remember a particular variation.

Command X is *efficient for machine* if it minimizes run-time and space. For instance, let A and B be two tables with schema (id,name,age). Suppose X is a blocker command that when applied to A and B produces a set of tuple pairs C . Then, to save space, X should not use (A.id, A.name, A.age, B.id, B.name, B.age), but only (A.id, B.id) as the schema of C .

If so, we need to store the “metadata information” that there is a key-foreign key (FK) relationship between tables A , B , and C . Storing this metadata in the tables themselves is not an option if we have already elected to store the tables using Pandas DataFrame (which cannot store such metadata, unless we redefine the DataFrame class). So we can use a stand-alone catalog Q to store such metadata for the tables.

But this raises a problem. If we use a command Y of some other package to remove a tuple from table A , Y is not even aware of catalog Q and so will not modify the metadata stored in Q . As a result, the metadata is now incorrect: Q still claims that an FK relationship exists between tables A and C . But this is no longer true.

To address this problem, we can design the tools to be *self-contained*. For example, if a tool Z is about to operate on table C and needs the metadata “there is an FK constraint between A and C ” to be true, it will first check that constraint. If the constraint is still true, then Z will proceed normally. Otherwise, Z outputs a warning that the FK constraint is no longer correct and then stops or proceeds (depending on the nature of the command). Thus, Z is self-contained in that it does not rely on anything outside to ensure the correctness of the metadata that it needs.

Trade-Offs Among the Principles: It should be clear by now that the above principles often interact and conflict with one another. For example, as discussed, to make commands interoperate, we may use Pandas DataFrames to hold the tables, and to make commands efficient, we may need to store metadata such as FK constraints. But this means the constraints should be stored in a global catalog. This makes extracting a set of commands to create a new package difficult, because the commands need access to this global catalog.

There are many examples such as this, which together suggest that designing an “ecosystem” of tools and packages that follow the above principles requires making trade-offs. We have made several such trade-offs in designing PyMatcher. But obtaining a clear understanding of these trade-offs and using it to design a better ecosystem is still ongoing work.

The Production Stage: So far, we have focused on the development stage for PyMatcher and have developed only a basic solution for the production stage. Specifically, we assume that after the development stage, the user has obtained an accurate EM workflow W , which is captured as a Python script (of a sequence of commands). We have developed tools that can execute these commands on a multicore single machine, using customized code or Dask (which is a

Python package developed by Anaconda that can be used to quickly modify a Python command to run on multiple cores, among others). We have also developed a how-to guide that tells the user how to scale using these tools.

5. CLOUDMATCHER

We now describe CloudMatcher, an EM system developed for lay users in the cloud environment.

Problem Scenarios: We use the term “lay user” to refer to a user who does not know programming, ML, or EM, but understands what it means to be match (and thus can label tuple pairs as match/no-match). Our goal is to build a system that such lay users can use to match two tables A and B . We call such systems self-service EM systems.

Developing an EM System for a Single User: In a recent work,³ we have developed Falcon, a self-service EM system that can serve a single user. As CloudMatcher builds on Falcon, we begin by briefly describing Falcon.

To match two tables A and B , like most current EM solutions, Falcon performs blocking and matching, but it makes both stages self-service (see Figure 3). In the blocking stage (Figure 3a), it takes a sample S of tuple pairs (Step ①) and then performs active learning with the lay user on S (in which the user labels tuple pairs as match/no-match) to learn a random forest F (Step ②), which is a set of n decision trees. The forest F declares a tuple pair p a match if at least αn trees in F declare p a match (where α is prespecified).

In Step ③, Falcon extracts all tree branches from the root of a tree (in random forest F) to a “No” leaf as candidate blocking rules. For example, the tree in Figure 4a predicts that two book tuples match only if their ISBNs match and the number of pages match. Figure 4b shows two blocking rules extracted from this tree. Falcon enlists the lay user to evaluate the extracted blocking rules and retains only the precise rules. In Step ④, Falcon executes these rules on tables A and B to obtain a set of candidate tuple pairs C . This completes the blocking stage (Figure 3a). In the matching stage (Figure 3b), Falcon performs active learning with the lay user on C to obtain another random forest G and then applies G to C to predict matches (Steps ⑤ and ⑥).

As described, Falcon is well suited for lay users, who only have to label tuple pairs as match/no-match. We implemented Falcon as CloudMatcher 0.1 and deployed as shown in Figure 5, with the goal of providing self-service EM to domain scientists at UW. Any scientist wanting to match two tables A and B can go to the homepage of CloudMatcher, upload the tables, and then label a set of tuple pairs (or ask crowd workers say on Mechanical Turk to do so). CloudMatcher uses the labeled pairs to block and match, as described earlier, and then returns the set of matches between A and B .

Developing an EM System for Multiple Users: We soon recognized, however, that CloudMatcher 0.1 does not scale, because it can execute only one EM workflow at a time. So we designed CloudMatcher 1.0, which can efficiently execute multiple concurrent EM workflows (e.g., submitted by multiple scientists at the same time). Developing CloudMatcher 1.0 was highly challenging.⁷ Our solution was to break each submitted EM workflow into multiple DAG fragments, where each fragment performs only one kind of task, for example, interaction with the user, batch processing of data, crowd-sourcing, et cetera. Next, we execute each fragment on an appropriate execution engines. We developed three execution engines: user interaction engine, crowd engine, and batch engine. To scale, we interleave the execution of DAG fragments coming from different EM workflows and coordinate all of the activities using a “metamanager.” See Govind et al.⁷ for more details.

Providing Multiple Basic Services: CloudMatcher 1.0 implemented only the above rigid Falcon EM workflow. As we interacted with real users, however, we observed that many users want to flexibly customize and experiment with different EM workflows. For example, a user may already know a blocking rule, so he or she wants to skip the step of learning such rules. Yet another user may want to use CloudMatcher just to label tuple pairs (e.g., to be used in PyMatcher).

So we developed CloudMatcher 2.0, which extracts a set of basic services from the Falcon EM workflow and makes them available on CloudMatcher, and then allows users to flexibly combine them to form different EM workflows (such as the original

Figure 3. The workflow of Falcon, where a lay user labels tuple pairs as match/no-match in Steps ②, ③, and ⑤.

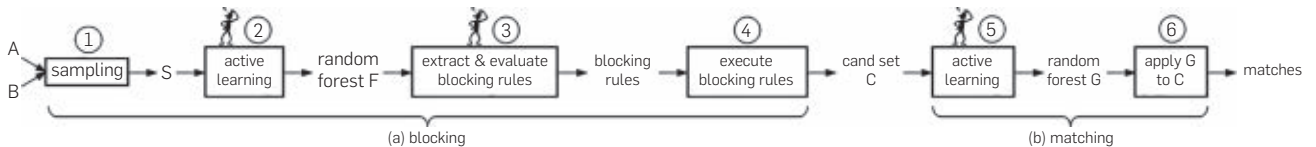


Figure 4. (a) A decision tree learned by Falcon and (b) blocking rules extracted from the tree.

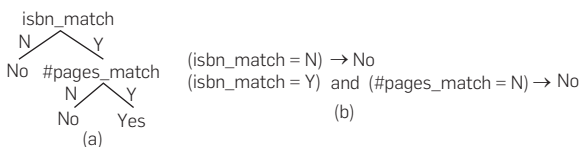


Figure 5. Self-service EM with CloudMatcher.



Falcon one). Appendix C of Govind et al.⁸ shows the list of services that we currently provide. *Basic services* include uploading a data set, profiling a data set, editing the metadata of a data set, sampling, generating features, training a classifier, et cetera. We have combined these basic services to provide *composite services*, such as active learning, obtaining blocking rules, and Falcon. For example, the user can invoke the “Get blocking rules” service to ask CloudMatcher to suggest a set of blocking rules that he/she can use. As another example, the user can invoke the “Falcon” service to execute the end-to-end Falcon EM workflow.

6. REAL-WORLD APPLICATIONS

We now discuss real-world applications of PyMatcher and CloudMatcher, as well as their typical usage patterns. In the discussion here, we measure EM accuracy using *precision*, the fraction of predicted matches that are correct, and *recall*, the fraction of true matches that are returned in the set of predicted matches.

Applications of PyMatcher: PyMatcher has been successfully applied to multiple real-world EM applications in both industry and domain sciences. It has been pushed into production in most of these applications and has attracted significant funding (e.g., \$950K from UW-Madison, \$1.1M from NSF, and \$480K from industry). It has also been used by 400+ students in 5 data science classes at UW-Madison. Finally, it has resulted in multiple publications, both in the database field and in domain sciences.⁴

Table 1 summarizes the real-world applications. The first column shows that PyMatcher has been used in a variety of companies and domain sciences. The second column shows that PyMatcher has been used for three purposes: debugging an EM pipeline in production (Walmart), building a better EM pipeline than an existing one (economics and land use), and integrating disparate data sets (e.g., Recruit, Marshfield Clinic, and limnology).

The third column shows the main results. This column shows that PyMatcher found EM workflows that were significantly better than the EM workflows in production in three cases: Walmart, Economics (UW), and Land Use (UW). The fourth column indicates that, based on those results, PyMatcher has been put into production in 6 out of 8 applications. This is defined as either (a) PyMatcher is used in a part of an EM pipeline in production or (b) the data resulted from using PyMatcher has been pushed into production, that is, being sent to and consumed by real-world customers.

The fifth column shows that in all cases that we know of, PyMatcher does not require a large team to work on it (and the teams are only part-time). The final column lists additional notable results. (Note that funding from UW came from highly selective internal competitions.) More details about these applications can be found in Govind et al.⁸ and Konda et al.¹⁰

CloudMatcher: CloudMatcher has been successfully applied to multiple EM applications and has attracted commercial interest. It has been in production at American Family Insurance since the summer of 2018 and is being considered for production at two other major companies.

Table 2 summarizes CloudMatcher’s performance on 13 real-world EM tasks. The first two columns show that CloudMatcher has been used in 5 companies, 1 nonprofit, and 1 domain science group, for a variety of EM tasks. The next two columns show that CloudMatcher was used to match tables of varying sizes, from 300 to 4.9M tuples.

Ignoring the next two columns on accuracy, let us zoom in on the three columns under “Cost” in Table 2. The first column (“Questions”) lists the number of questions CloudMatcher had to ask, that is, the number of tuple pairs to be labeled. This number ranges from 160 to 1200 (the upper limit for the current CloudMatcher).

Table 1. Real-world deployment of PyMatcher.

Problem owner	Problem type	Notable result	In production?	Team	Other
Walmart	Debug a system in production that matches products	Improved recall by 34%, reduced precision by 0.65%	Yes	1 student, 1 employee	Funding
Recruit holdings	Matching names of stores, companies, properties	Reported 98.9% accuracy on matching 10K store names	Yes	Multiple employees	Press release
Johnson controls	Matching suppliers	Precision and recall in 96–100%	Unknown	1 student	Funding
Marshfield clinic	Matching drugs	99.2% precision and 95.3% recall	Yes	1 student, 1 employee	Paper
Economics (UW)	Matching grants. Build a better EM pipeline	Precision in [96.7%, 98.8%], recall in [94.2%, 97.1%] A system in production achieves 100% precision, recall in [65.1%, 71.8%]	Not yet	2 students	Paper published, funding from UW
Land use (UW)	Matching cattle ranches. Build a better EM pipeline	Precision in [89.7%, 99.0%], recall in [79.2%, 92.2%] A system in production achieves precision in [94.9%, 100%], recall in [29.4%, 46.6%]	Yes	1 student, 1 programmer, 2 staff persons	Paper planned, funding from UW
Biomedicine (UW)	Matching ontology terms	metasra.biostat.wisc.edu	Yes	1 student	Paper published
Limnology (UW)	Matching table attributes	High-value data sets created from multiple data sets	Yes	2 students	Funding from UW

Table 2. Real-world deployment of CloudMatcher.

Problem owner	Problem type	Table A	Table B	Precision		Cost			Time		
				(%)	Recall (%)	Questions	Crowd	Compute	User/crowd	Machine	Total
Fortune 500 Company	Phoenix customers	300	300	96.4	99.03	160	–	\$2.33	9m	5m	14m
	Commercial insurance policy holders	1049	17,572	96.15	97.22	321	–	\$2.33	18m	25m	43m
	Commercial farm/ranch policy members	109,974	4,922,505	99.5	95	780	–	\$13.96	50m	4h 58m	5h 48m
Johnson Controls International	Vehicles Drivers	18,938	72,898	66.02–80.02	81.65–93.15	851	–	\$7.00	2h	46m	2h 46m
	Addresses	790	634	99.86	94.89	250	–	\$2.33	10m	8m	18m
	Vendors	90,673	231,081	93.22–95.72	76.93–81.01	1200	\$72	–	36h 48m	38m	37h 26m
	Vendors (no Brazil)	50,295	50,292	29.95–38.04	91.89–98.10	1160	\$69.60	–	30h 31m	58m	31h 29m
	Vendors (no Brazil)	28,152	28,149	95.44–97.75	88.82–92.41	1200	\$72	–	22h 19m	22m	22h 41m
UW Health	Doctors & staff	1786	1786	99.66	98.18	1200	–	\$4.66	50m	15m	1h 5m
Large Data Integration Company	Persons	48,119	48,119	100–100	98.42–100	462	–	\$7.00	36m	1h 35m	2h 11m
Marshfield Clinic Nonprofit Org	Drugs	446,048	440,048	99.14–99.63	98.45–99.14	1162	–	–	1h 10m	8h 40m	9h 50m
Domain Science	Elected officials	9751	706,878	93.75–96.32	95.50–97.76	960	\$57.60	–	23h 14m	23m	23h 37m
	UMetrics economics	2616	21,530	94.5–96.5	98.12–99.21	680	\$61.20	–	23h 12m	12m	23h 24m

In the next column (“Crowd”), a cell such as “\$72” indicates that for the corresponding EM task, CloudMatcher used crowd workers on Mechanical Turk to label tuple pairs, and it cost \$72. A cell “–” indicates that the task did not use crowd-sourcing. It used a single user instead, typically the person who submitted the EM task, to label, and thus incurred no monetary cost.

In the third column (“Compute”), a cell such as “\$2.33” indicates that the corresponding EM task used AWS, which charged \$2.33. A cell such as “–” indicates that the EM task used a local machine owned by us, and thus incurred no monetary cost.

Turning our attention to the last three columns under “Time,” the first column (“User/Crowd”) lists the total labeling time, either by a single user or by the Mechanical Turk crowd. We can see that when a single user labeled, it was typically quite fast, with time from 9m to 2h. When a crowd labeled, time was from 22h to 36h (this does not mean crowd workers labeled nonstop and took that long; it just meant Mechanical Turk took that long to finish the labeling task). These results suggest that CloudMatcher can execute a broad range of EM tasks with very reasonable labeling time from both users and crowd workers. The next two columns under “Time” show the machine time and the total time.

We now zoom in on the accuracy. The columns “Precision” and “Recall” show that in all cases except three, CloudMatcher

achieves high accuracy, often in the 90 percentage. The three cases of limited accuracy are “Vehicles,” “Addresses,” and “Vendors.” A domain expert at American Family Insurance (AmFam) labeled tuple pairs for “Vehicles.” But the data was so incomplete that even he was uncertain in many cases on whether the tuple pair matches. At some point, he realized that he had incorrectly labeled a set of tuple pairs, but CloudMatcher provided no way for him to “undo” the labeling, hence the low accuracy. This EM task is currently being re-executed at AmFam.

For “Vendors,” it turned out that the portion of data that consists of Brazilian vendors is simply incorrect: the vendors entered some generic addresses instead of their real addresses. As a result, even users cannot match such vendors. Once we removed such vendors from the data, the accuracy significantly improved (see the row for “Vendors (no Brazil)”). It turned out that “Addresses” had similar dirty data problems, which explained the low recall of 76–81%.

Typical Usage Patterns: We observed the following patterns of using PyMatcher and CloudMatcher. When working with enterprise customers, a common scenario is that the EM team, which typically consists of only a few developers, is overwhelmed with numerous EM tasks sent in by many business teams across the enterprise.

To address this problem, the EM team asks business teams to use CloudMatcher to solve their EM tasks

(in a self-service fashion), contacting the EM team only if CloudMatcher does not reach the desired EM accuracy. In those cases, the EM team builds on the results of CloudMatcher but uses PyMatcher to debug and improve the accuracy further.

We found that the EM team also often uses CloudMatcher to solve their own EM tasks, because it can be used to quickly solve a large majority of EM tasks, which tend to be “easy,” allowing the EM team to focus on solving the small number of more difficult EM tasks using PyMatcher.

For domain sciences at UW, some teams used only CloudMatcher, either because they do not have EM and ML expertise or they found the accuracy of CloudMatcher acceptable. Some other teams preferred PyMatcher, as it gave them more customization options and higher EM accuracies.

Finally, some customers used both and switched between them. For example, a customer may use PyMatcher to experiment and create a set of blocking rules and then use CloudMatcher to execute these rules on large tables.

7. DISCUSSION

We now discuss lessons learned and ongoing work.

The Need for How-to Guides: Our work makes clear that it is very difficult to fully automate the EM process. The fundamental reason is because at the start, the user often does not fully understand the data, the match definition, and even what he or she wants. For example, in a recent case study with PyMatcher,¹⁰ we found that the users repeatedly revised their match definition *during* the EM process, as they gained a better understanding of the data.

This implies that the user must “be in the loop” and that a guide is critical for telling the user what to do, step by step. In addition, we found that these guides provide assurance to our customers that we can help them do EM *end to end*. The guides provide a common vocabulary and roadmap for everyone on the team to follow, regardless of their background. Even for the EM steps where we currently do not have tools, the guide still helps enormously, because it tells the customers what to do, and they can do it manually or find some external tools to help with it. Such guides, however, are completely missing from most current EM solutions and systems.

Difficulties in Developing How-to Guides: Surprisingly, we found that developing clear how-to guides is quite challenging. For example, the current guide for PyMatcher is still quite preliminary. It does not provide detailed guidance for many steps such as how to help users converge to a match definition, how to collaboratively label effectively, and how to debug learning-based matchers, among others. Developing detailed guidance for such steps is ongoing work.

Focusing on Reducing User Effort: Many existing EM works focus on automating the EM process. In Magellan, our focus switched to developing a step-by-step guide that tells users how to execute the EM process, identifying “pain points” of the guide and then *developing tools to reduce the user effort in the pain points*. We found this new perspective to be much more practical. It allows us to quickly develop end-to-end EM solutions that we can deploy with real users on Day 1 and then work with them closely to gradually improve these solutions and reduce their effort.

Many New Pain Points: Existing EM work has largely focused on blocking and matching. Our work makes clear that there are many pain points that current work has ignored or not been aware of. Examples include how to quickly converge to a match definition, how to label collaboratively, how to debug blockers and matchers, and how to update an EM workflow if something (e.g., data and match definition) has changed. We believe that more effort should be devoted to addressing these real pain points in practice.

Monolithic Systems vs. Ecosystems of Tools: We found that EM is so much messier than we thought. Fundamentally, it was a “trial and error” process, where users kept experimenting until they find a satisfactory EM workflow. As a result, users tried all kinds of workflows, customization, data processing, et cetera. (e.g., see Konda et al.¹⁰).

Because EM is so messy and users want to try so many different things, we found that an ecosystem of tools is ideal. For every new scenario that users want to try, we can quickly put together a set of tools and a mini how-to guide that they can use. This gives us a lot of flexibility.

Many “trial” scenarios require only a part of the entire EM ecosystem. Having an ecosystem allows us to very quickly pull out the needed part, and popular parts end up being used everywhere. For example, several string matching packages in PyMatcher are so useful in many projects (not just in EM) that they ended up being installed on Kaggle, a popular data science platform.

Extensibility is also much easier with an ecosystem. For example, recently, we have developed a new matcher that uses deep learning to match textual data.¹¹ We used PyTorch, a new Python library, to develop it, released it as a new Python package in the PyMatcher ecosystem, and then extended our guide to show how to use it. This smoothly extended PyMatcher with relatively little effort.

Clearly, we can try to achieve the above three desirable traits (flexibility/customizability, partial reuse, and extensibility) with monolithic stand-alone systems for EM, but our experience suggests it would be significantly harder to do so. Finally, we found that it is easier for academic researchers to develop and maintain (relatively small) tools in an ecosystem, than large monolithic systems.

Using Multiple Execution Environments (EEs): We found that users often want to use multiple EEs for EM. For example, a user may want to work on-premise using his or her desktop to experiment and find a good EM workflow and then upload and execute the workflow on a large amount of data on the cloud. Whereas working on-premise, if the user has to perform a computation-intensive task, such as executing a blocker, he or she may opt to move that task to the cloud and execute it there. Similarly, collaborative tasks such as labeling and data cleaning are typically executed on the cloud, using Web interfaces, or on mobile devices, although the user is taking the bus, say.

This raises two challenges. First, we need to develop an ecosystem of EM tools for each EE, for example, Python packages for the on-premise EE, containerized apps for the cloud, and mobile apps for smart phones. Second, we need to develop ways to quickly move data, models, and workflows

across the EEs, to allow users to seamlessly switch among the EEs. In Magellan, we have taken some initial steps to address these two challenges. But clearly a lot more remains to be done.

Serving Both Lay Users and Power Users: In Magellan, we have developed PyMatcher as a solution for power users and CloudMatcher as a self-service solution for lay users. Serving both kinds of users is important, as suggested by our experience with EM teams and business teams at enterprises, as well as with domain scientists at UW (see Section 6).

Support for Easy Collaboration: We found that in many EM settings there is actually a team of people wanting to work on the problem. Most often, they collaborate to label a data set, debug, clean the data, et cetera. However, most current EM tools are rudimentary in helping users collaborate easily and effectively. As users often sit in different locations, it is important that such tools are cloud-based, to enable easy collaboration.

Managing Machine Learning “in the Wild”: Our work makes clear that ML can be very beneficial to EM, mainly because it provides an effective way to capture complex matching patterns in the data and to capture domain expert’s knowledge about such patterns. ML is clearly at the heart of EM workflows supported by PyMatcher and CloudMatcher. In many real-world applications we have worked with, ML helps significantly improve recall although retaining high precision, compared to rule-based EM solutions.


Yet to our surprise, deploying even traditional ML techniques to solve EM problems already raises many challenges, such as labeling, debugging, coping with new data, et cetera. Our experience using PyMatcher also suggests that the most accurate EM workflows are likely to involve a combination of ML and rules. More generally, we believe ML must be used effectively in conjunction with hand-crafted rules, visualization, good user interaction, and Big Data scaling, in order to realize its full potential.

Cannot Work on EM in Isolation: It turned out that when working on EM, users often perform a wide variety of non-EM tasks, such as exploring the data (to be matched), understanding it, cleaning, extracting structures from the data, et cetera. User also often perform many so-called DS tasks, such as visualization, analysis, et cetera., by invoking DS tools (e.g., calling Matplotlib or running a clustering algorithm in scikit-learn). Worse, users often interleave these non-EM and DS tasks with the steps of the EM process. For example, if the accuracy of the current EM workflow is low, users may want to clean the data, then retrain the EM matcher again, then clean the data some more, et cetera.

As described, building different ecosystems of tools for different tasks (e.g., EM, schema matching, cleaning, exploration, and extraction) is suboptimal, because constant switching among them creates a lot of overhead. Rather, we believe it is important to build unified ecosystems of tools. That is, for the on-premise EE, build one (or several) ecosystem that provides tools not just for EM, but also for exploration, understanding, cleaning, et cetera. Then, repeat for the cloud and mobile EEs. Further, these ecosystems should “blend in” seamlessly with DS ecosystems of tools, by being built on top of those.

Going forward, we are continuing to develop both the on-premise and cloud-hosted ecosystems of EM tools. In particular, we are paying special attention to the cloud-hosted ecosystem, where in addition to CloudMatcher, we are developing many other cloud tools to label, clean, and explore the data. We are also working on ways for users to seamlessly move data, workflows, and models across these two ecosystems. Finally, we are looking for more real-world applications to “test drive” Magellan.

8. CONCLUSION

We have described Magellan, a project to build EM systems. The key distinguishing aspect of Magellan is that unlike current EM systems, which use an RDBMS monolithic stand-alone system template, Magellan borrows ideas from the data science field to build ecosystems of interoperable EM tools. Our experience with Magellan in the past few years suggests that this new “system template” is highly promising for EM. Moreover, we believe that it can also be highly promising for other non-EM tasks in data integration, such as data cleaning, data extraction, and schema matching, among others. 

References

1. Workshop on Human-In-the-Loop Data Analytics, <http://hilda.io/>.
2. Christen P. *Data Matching*. Springer, 2012.
3. Das, S., P.S.G.C., Doan, A., Naughton, J.F., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V., Park, Y. Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD '17* (New York, NY, USA, 2017), ACM, 1431–1446.
4. Doan, A., et al. Toward a system building agenda for Data Integration (and Data Science). *IEEE Data Eng. Bull.* 41, 2 (2018), 35–46.
5. Doan, A., Halevy, A.Y., Ives, Z.G. *Principles of Data Integration*. Morgan Kaufmann, 2012.
6. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S. Duplicate record detection: A survey. *IEEE TKDE* 19, 1 (2007), 1–16.
7. Govind, Y., et al. Cloudmatcher: A cloud/crowd service for entity matching. In *BIGDAS* (2017).
8. Govind, Y., et al. Entity matching meets data science: A progress report from the magellan project. In *SIGMOD* (2019).
9. Konda, Y., et al. Magellan: Toward building entity matching management systems. *PVLDB* 9, 12 (2016), 1197–1208.
10. Konda, P., et al. Performing entity matching end to end: A case study. In *EDBT* (2019).
11. Mudgal, S., et al. Deep learning for entity matching: A design space exploration. In *IGMOD* (2018).
12. Papadakis, G., et al. The return of JedAI: End-to-End entity resolution for structured and semi-structured data. *PVLDB* 11, 12 (2018), 1950–1953.
13. Papadakis, G., et al. Web-scale, Schema-Agnostic, End-to-End Entity Resolution. In *The Web Conference (WWW)*, (Lyon, France, April), 2018.

AnHai Doan, Pradap Konda, Paul Sunganthan GC, Yash Govind, Derek Paulsen, Kaushik Chandrasekhar, Philip Martinkus, and Matthew Christie.
University of Wisconsin-Madison.

Association for
Computing Machinery

ACM Transactions on Quantum Computing (TQC)

A New Journal from ACM

ACM Transactions on Quantum Computing (TQC) publishes high-impact, original research papers and select surveys on topics in quantum computing and quantum information science. The journal targets the quantum computer science community with a focus on the theory and practice of quantum computing.



For further information
and to submit your
manuscript,
visit tqc.acm.org

Technical Perspective Supporting Linear Algebra Operations in SQL

By Yannis Papakonstantinou

LINEAR ALGEBRA OPERATIONS are at the core of machine learning. Multiple specialized systems have emerged for the scalable, distributed execution of matrix and vector operations. The relationship of such computations to data management and databases however brings frictions. It is well known that a great deal of human time and machine time is being spent nowadays on fetching data out of the database and performing a computation on a specialized system. One answer to the issue is that we truly need a new kind of non-SQL database that is tuned to these computations.

The creators of SimSQL opted for the decidedly incremental approach. Can we make a very small set of changes to the relational model and RDBMS software to render them suitable for executing linear algebra in the database?


We have come across the “brand new system” versus “incremental to relational” question many times in the database field. For example, do we need brand new query languages and query processors for data cubes? Or do we need to have our query processors pay attention to specific cases that are especially common in data analytics queries over stars and snowflakes? Do semi-structured query languages need to depart from SQL or it is enough to be incremental to SQL? Same for query processors. Repeat the questions to graph data and RDF data. In many cases, new custom systems emerged only to figure out later that we could/should have tackled the problem incrementally. That is the trap the authors of the following paper avoid.

This is not to say that radical changes and extensions should be forbidden. Rather it says that we should closely scrutinize the necessity of the changes, do them when needed and keep them minimal. The authors

Do we need a completely new database system to support machine learning?

identify the right opportunities. Here is a non-exhaustive list:

- Writing matrix and vector operations as a join over the index can be syntactically tedious. They solve the problem by introducing special syntactic features.
 - ▶ They notice a connection between signatures and size estimation and exploit it.
 - ▶ They allow their query user to move across different denormalizations to find the one that makes sense from expressiveness and performance point of view. The point where types relate to performance is whether the right level of granularity for distribution in a shared-nothing architecture is specified.

Overall, the extensions of this paper follows a thoughtful and minimal approach that is worth studying in the particular field of linear algebra operations, as well as generally in the design of systems for analytics. 

Yannis Papakonstantinou is a professor of computer science and engineering at the University of California, San Diego, CA, USA.

Scalable Linear Algebra on a Relational Database System

By Shangyu Luo, Zekai J. Gao, Michael Gubanov, Luis L. Perez, Dimitrije Jankov, and Christopher Jermaine

Abstract

As data analytics has become an important application for modern data management systems, a new category of data management system has appeared recently: the scalable linear algebra system. We argue that a parallel or distributed database system is actually an excellent platform upon which to build such functionality. Most relational systems already have support for cost-based optimization—which is vital to scaling linear algebra computations—and it is well known how to make relational systems scalable.

We show that by making just a few changes to a parallel/distributed relational database system, such a system can become a competitive platform for scalable linear algebra. Taken together, our results should at least raise the possibility that brand new systems designed from the ground up to support scalable linear algebra are not absolutely necessary, and that such systems could instead be built on top of existing relational technology.

1. INTRODUCTION

Data analytics, such as machine learning and large-scale statistical processing, is an important application domain, and such computations often require linear algebra. As such, a lot of recent efforts have been targeted at building distributed linear algebra systems, with the goal of supporting large-scale data analytics. Unlike classical efforts in high-performance computing such as ScaLAPACK⁶, such systems may include support for storage/retrieval of data to/from disk, buffering/caching of data, and automatic logical/physical optimizations of computations (automatic rewriting of queries, pipelining, etc.). Such systems also typically offer some form of recovery, as well as a domain-specific language.

One example of such a system is SystemML, developed at IBM.¹² Given deep learning's reliance on arrays and array-based operations such as matrix multiply, systems facilitating distributed deep learning, such as TensorFlow,³ can also be included among such efforts. In the database area, there has long been of interest in building array database systems.^{17, 5} A motivating use case for these systems is distributed linear algebra. Moreover, there have also been significant efforts targeted at using dataflow systems such as Apache Spark²⁰ to build distributed linear algebra dataflow APIs (such as Spark's `mllib.linalg`¹).

Is a new type of system actually necessary? The hypothesis underlying this paper is that building a new system from scratch for distributed linear algebra may not be necessary. Instead, we believe that with just a few changes, a classical, parallel relational database is actually an excellent platform for building a scalable linear algebra system. In practice, there is a close correspondence between distributed linear

algebra and distributed relational algebra, the foundation of modern database systems, meaning that it is easy to use a database for scalable linear algebra. Relational database systems are highly performant, reaping the benefits of decades of research and engineering efforts targeted at building efficient systems. Further, relational systems already have software components such as a cost-based query optimizer to aid in performing efficient computations. In fact, much of the work that goes into developing a scalable linear algebra system from the ground up⁷ requires implementing functionality that looks a lot like a database query optimizer.¹⁰

Given that much of the world's data currently sits in relational databases, and that dataflow systems increasingly provide at least some support for relational processing^{4, 19}, building linear algebra facility into relational systems would mean that much of the world's data would be sitting in systems capable of performing scalable linear algebra. This would have several obvious benefits:

1. It would eliminate the “extract-transform-reload nightmare”, particularly if the goal is performing analytics on data already stored in a relational system. It is difficult and expensive (in terms of computing/network costs and engineering dollars) to remove data from one system and put it in another, and if a database came off-the-shelf with the necessary functionality, there would be no reason to undertake such an often arduous task.
2. It would obviate the need for practitioners to adopt yet another type of data processing system in order to perform mathematical computations.
3. The design and implementation of high-performance distributed and parallel relational systems is well-understood. If it is possible to adapt such a system to the task of scalable linear algebra, most or all of the science performed over decades, aimed at determining how to build a distributed relational system, is directly applicable.

Along those lines, in this paper, we ask the question:

can we make a very small set of changes to the relational model and an RDBMS software to render them suitable for in-database linear algebra?

The approach we examine is simple: we consider adding new VECTOR, MATRIX, and LABELED_SCALAR data types to relational database systems. Technically, this seems to be a rather minor change. After all, array has been available as a data type

The original version of this paper was published in the *Proceedings of the IEEE 33rd International Conference on Data Engineering*, 2017, 523–534.

in most modern DBMSs—arrays can clearly be used to encode vectors and matrices—and some database systems (such as Oracle Database) offer a form of integration between arrays and linear algebra libraries such as BLAS and LAPACK. However, these previous ad-hoc approaches do not offer complete integration with the database system. The query optimizer, for example, does not understand the semantics of the linear algebra, and this results in losing opportunities for optimization.

In this paper, we evaluate our ideas, and we believe that our results call into question the need to build yet another special-purpose data management system for linear-algebra-based analytics.

2. LA ON TOP OF RA

In this section of the paper, we discuss why a relational database system might make an excellent platform for high-performance, distributed linear algebra. We then discuss the challenges in using a database system for linear algebra, as well as our basic approach.

2.1. Linear and relational algebra

Development of distributed algorithms for linear algebra has been an active area of scientific investigation for decades. Figure 1(a) shows the example of performing a distributed multiplication of two large, dense matrices, $O \leftarrow L \times R$.

For efficiency and storage considerations, matrices in a distributed system are typically “blocked” or “chunked”; that is, they are divided into smaller matrices, which can then be moved around in bulk to specific processors where high-performance local computations are performed. Imagine that the six blocks making up each of the two input matrices L and R are distributed among three nodes as shown at the left of Figure 1(b). The blocks from L are hash-partitioned randomly, whereas the blocks from R are round-robin-partitioned, based upon each block’s row identifier.

As a first step, we would shuffle the blocks from L so that all of the blocks from L , column i , are co-located with all of the blocks from R , row i . Then, at each node, a local join

(in this case, a cross product) is performed to iterate through all $(L_{j,i}, R_{i,k})$ pairs that can be formed at the node. For each pair, a matrix multiply is performed, so that $I_{i,j,k} \leftarrow L_{j,i} \times R_{i,k}$. Finally, all of the $I_{i,j,k}$ blocks are again shuffled so that all $I_{i,j,k}$ blocks are co-located based upon their (j, k) values—these blocks are then summed, so that the output block is computed as $O_{j,k} \leftarrow \sum_i I_{i,j,k}$.

The key observation is that *this is really just a relational algebra computation* over the blocks making up L and R . The first two steps of the computation are a distributed join that computes all $(L_{j,i}, R_{i,k})$ pairs, followed by a projection that performs the matrix multiply. The next two steps—the shuffle and summation—are nothing more than a distributed grouping with aggregation.

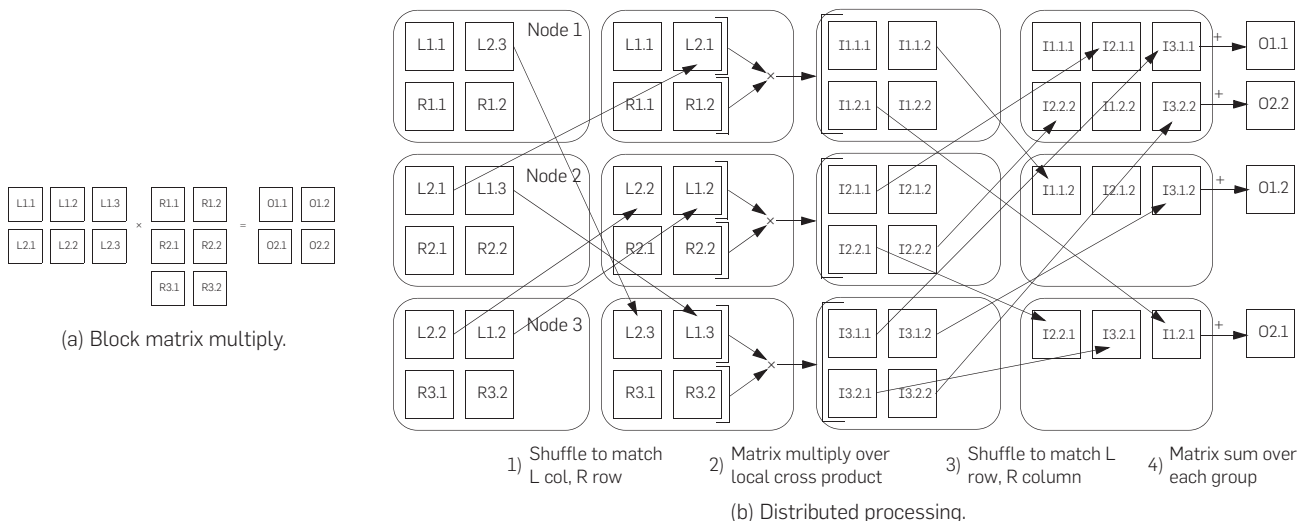
The matrix multiplication example shows that distributed linear algebra computations are often nothing more than distributed relational algebra computations. This fact underlies our assertion that a relational database system makes an excellent platform for distributed linear algebra. Database researchers have spent decades studying efficient algorithms for distributed joins and aggregations, and many relational systems are mature and highly performant; there is no need to reinvent the wheel.

A further benefit of using a distributed database system as a linear algebra engine is that decades of work in query optimization are directly applicable. In our example, we decided to shuffle L because R was already partitioned on the join key. Had L been pre-partitioned and not R , it would have been better to shuffle R . This is *exactly* the sort of decision that a modern query optimizer makes with total transparency. Using a database as the basis for a linear algebra engine gives us the benefit of query optimization for free.

2.2. The challenges

However, there are two main concerns associated with implementing linear algebra directly on top of an existing relational system, without modification. First is the

Figure 1. Distributed processing of a block matrix multiply.



complexity of writing linear algebra computations on top of SQL. Consider a data set consisting of the vectors $\{x_1, x_2, \dots, x_n\}$, and imagine that our goal is to compute the distance

$$d_A^2(x_i, x') = (x_i - x')^T A (x_i - x')$$

for a Riemannian metric¹⁶ encoded by the matrix **A**. We might wish to compute this distance between a particular data point x_i and every other point x' in the database. This would be required in a k NN-based classification in the metric space defined by **A**.

This distance computation can be implemented in SQL as follows. Assume the set of vectors is encoded as a table:

```
data (pointID, dimID, value)
```

with the matrix **A** encoded as another table:

```
matrixA (rowID, colID, value)
```

Then, the desired computation is expressed in SQL as:

```
CREATE VIEW xDiff (pointID, dimID, value) AS
  SELECT x2.pointID, x2.dimID, x1.value - x2.value
  FROM data AS x1, data AS x2
  WHERE x1.pointID = i AND x1.dimID = x2.dimID

SELECT x.pointID, SUM (firstPart.value * x.value)
FROM (SELECT x.pointID AS pointID, a.colID AS
      colID, SUM (a.value * x.value) AS value
      FROM xDiff AS x, matrixA AS a
      WHERE x.dimID = a.rowID
      GROUP BY x.pointID, a.colID)
      AS firstPart, xDiff AS x
WHERE firstPart.colID = x.dimID
      AND firstPart.pointID = x.pointID
GROUP BY x.pointID
```

Although it is clearly possible to write such a code, it is not necessarily a good idea. The first obvious problem is that this is a very intricate specification, requiring a nested subquery and a view—without the view it is even more intricate—and it bears little resemblance to the original, simple mathematics.

The second problem is perhaps less obvious from looking at the code, but just as severe: performance. This code is likely to be inefficient to execute, requiring three or four joins and two groupings. Even more concerning in practice is the fact that if the data is dense and the number of data dimensions is large (that is, there are a lot of `dimID` values for each `pointID`), then the execution of this query will move a huge number of small tuples through the system, because a million, thousand-dimensional vectors are encoded as a billion tuples. In the classical, iterator-based execution model, there is a fixed cost incurred per tuple, which will translate to a very high execution cost. Vector-based processing can alleviate this somewhat, but the fact remains that satisfactory performance is unlikely. This fixed-cost-per-tuple problem was often cited as the impetus for designing new systems, specifically for vector- and matrix-based processing, or for processing of more general-purpose arrays.

2.3. The solution

As a solution, we propose a very small set of changes to a typical relational database system that includes adding new `LABELED_SCALAR`, `VECTOR`, and `MATRIX` data types to the relational model. Because these nonnormalized data types cause the contents of vectors and matrices to be manipulated as a single unit during query processing, the simple act of adding these new types brings significant performance improvements.

Further, we propose a very small number of SQL language extensions for manipulating these data types and moving between them. This alleviates the complicated-code problem. In our Riemannian metric example, the two input tables `data` and `matrixA` become `data (pointID, val)` and `matrixA (val)`, respectively, where `data.val` is a vector, and `matrixA.val` is a matrix. The SQL code to compute the pairwise distances becomes dramatically simpler:

```
SELECT x2.pointID,
       inner_product (
         matrix_vector_multiply (
           a.val, x1.val - x2.val),
           x1.val - x2.val) AS value
FROM data AS x1, data AS x2, matrixA AS a
WHERE x1.pointID = i
```

In the next full section of the paper, we describe our proposed extensions in detail.

3. OVERVIEW OF EXTENSIONS

3.1. New types

We propose adding `VECTOR`, `MATRIX`, and `LABELED_SCALAR` column types to SQL and the relational model, as well as implementing a useful set of operations over those types (`diag` to extract the diagonal of a matrix, `matrix_vector_multiply` to multiply a matrix and a vector, `matrix_matrix_multiply` to multiply two matrices, etc.). Overall, 22 various built-in functions over `LABELED_SCALAR`, `VECTOR`, and `MATRIX` types are present in our implementation. Each element of a `VECTOR` or a `MATRIX` is a `DOUBLE`.

In this particular subsection, we focus on introducing the `VECTOR` and `MATRIX` types; `LABELED_SCALAR` will be considered in detail in a subsequent subsection.

For a simple example of the use of `VECTOR` and `MATRIX` types, consider the following table:

```
CREATE TABLE m (mat MATRIX[10][10],
                 vec VECTOR[100]);
```

This code specifies a relational table, where each tuple in the table has two attributes, `mat` and `vec`, of types `MATRIX` and `VECTOR`, respectively. In our language extensions, `VECTORS` and `MATRIXES` (as above) can have specified sizes, in which case operations such as `matrix_vector_multiply` are automatically type-checked for size mismatches. For example, the following query:

```
SELECT matrix_vector_multiply (m.mat, m.vec)
      AS res
FROM m
```

will not compile because the number of columns in `m.mat` does not match the number of entries in `m.vec`. However, if the original table declaration had been:

```
CREATE TABLE m (mat MATRIX[10][10],
                 vec VECTOR[10]);
```

then the aforementioned SQL query would compile and execute, and the output would be a database table with a single attribute (called `res`) of type `VECTOR[10]`.

Note that in our extensions, there is no distinction between row and column vectors; whether or not a vector is a row or a column vector is up to the interpretation of each individual operation. `matrix_vector_multiply` interprets a vector as a column vector, for example. To perform a matrix-vector multiplication treating the vector as a row vector, a programmer would first transform the vector into a one-row matrix (this transformation is described in the subsequent subsection), and then call `matrix_matrix_multiply`. Or, a programmer could transform the matrix first, and then apply the `matrix_vector_multiply` function.

It is possible to create `MATRIX` and `VECTOR` types where the sizes are unspecified:

```
CREATE TABLE m (mat MATRIX[10][10],
                 vec VECTOR[]);
```

In this case, the aforementioned `matrix_vector_multiply` SQL query would compile, but there could possibly be a runtime error if one or more of the tuples in `m` contained a `vec` attribute that did not have 10 entries.

It is possible to have a `MATRIX` declaration where only one of the dimensionalities is given; for example, `MATRIX[10][]`. However, it is generally a good idea for a programmer to specify the sizes in the table declaration. If a dimensionality *is* given, then the system ensures that there can be no runtime failures due to size mismatches. During the loading time, data is checked to ensure the correct dimensionality, and queries are type-checked to ensure that proper dimensionalities are used and satisfied. Further, if dimensions are known, it can help the optimization process; a plan that uses a linear algebra operation that greatly reduces the amount of data early on (a multiplication of two “skinny” matrices, for example, which results in a small output matrix) may be chosen as being optimal.

3.2. Built-in operations

In addition to a long list of standard linear algebra operations, the standard arithmetic operations `+`, `-`, `*` and `/` (element-wise) are also defined over `MATRIX` and `VECTOR` types. For example,

```
CREATE TABLE m (mat MATRIX[100][10]);

SELECT mat * mat
FROM m
```

returns a database table which stores the Hadamard product of each matrix in `m` with itself.

As the standard arithmetic operations are all overloaded to work with `MATRIX` and `VECTOR` types, it means that the standard SQL aggregate operations all work as expected automatically. The `SUM` aggregate over `VECTOR` type attribute, for example, performs a `+` (entry-by-entry addition) over each `VECTOR` in a relation. This can be very convenient for implementing mathematical computations. For example, imagine that we have a matrix stored as a relational table of vectors, and we wish to perform a standard Gram matrix computation (if the matrix `X` is stored as a set of columns $X = \{x_1, x_2, \dots, x_n\}$, then the Gram matrix of `X` is $\sum_{i=1}^n x_i x_i^T$). This computation can be implemented using our extensions as:

```
CREATE TABLE v (vec VECTOR[]);

SELECT SUM (outer_product (vec, vec))
FROM v
```

Arithmetic between a scalar value and a `MATRIX` or `VECTOR` type performs the arithmetic operation between the scalar and *every* entry in the `MATRIX` or `VECTOR`. In this way, it becomes very easy to specify linear algebra computations of significant complexity using just a few lines of code. For example, consider the problem of learning a linear regression model. Given a matrix $X = \{x_1, x_2, \dots, x_n\}$ and a set of outcomes $\{y_1, y_2, \dots, y_n\}$, the goal is to estimate a vector $\hat{\beta}$ where for each i , $x_i \hat{\beta} \approx y_i$. In practice, $\hat{\beta}$ is typically computed so as to minimize the squared loss $\sum_i (x_i \hat{\beta} - y_i)^2$. In this case, the formula for $\hat{\beta}$ is given as:

$$\hat{\beta} = \left(\sum_i x_i x_i^T \right)^{-1} \left(\sum_i x_i y_i \right)$$

This can be coded as follows. If we have:

```
CREATE TABLE X (i INTEGER, x_i VECTOR []);
CREATE TABLE Y (i INTEGER, y_i DOUBLE);
```

then the SQL code to compute $\hat{\beta}$ is:

```
SELECT matrix_vector_multiply (
      matrix_inverse (
        SUM (outer_product (X.x_i, X.x_i)),
        SUM (X.x_i * y_i))
FROM X, Y
WHERE X.i = Y.i
```

Note the multiplication `X.x_i * y_i` between the vector `X.x_i` and the scalar `y_i`, which multiplies `y_i` by each entry in `X.x_i`.

3.3. Moving between types

By introducing `MATRIX` and `VECTOR` types, we then have new, de-normalized alternatives for storing data. For example, a matrix can be stored as a traditional relation:

```
mat (row INTEGER, col INTEGER, value DOUBLE)
```

or as a relation containing a set of row vectors, or as a set of column vectors using


```
row_mat (row INTEGER, vec_value VECTOR[])
or
col_mat (col INTEGER, vec_value VECTOR[])
```

Or, the matrix can be stored as a relation with a single tuple having the whole matrix:

```
mat (value MATRIX [][])
```

It is of fundamental importance to be able to move around between these various representations, for several reasons. Most importantly, each representation has its own performance characteristics and ease-of-use for various tasks; depending upon a particular computation, one may be preferred over another.

Reconsider the linear regression example. Had we stored the data as:

```
CREATE TABLE X (mat MATRIX [][]);
CREATE TABLE y (vec VECTOR []);
```

then the SQL code to compute $\hat{\beta}$ would have been:

```
SELECT matrix_vector_multiply (
  matrix_inverse (
    matrix_matrix_multiply(trans_matrix(mat), mat) ),
  matrix_vector_multiply (
    trans_matrix (mat), vec) )
FROM X, y
```

Arguably, this is a more straightforward translation of the mathematics compared to the code that stores X as a set of vectors. However, it may not perform as well because it may be more difficult to parallelize on a shared-nothing cluster of machines. In comparison to the vector-based implementation, the matrix multiply $X^T X$ is implicit in the relational algebra.

As different representations are going to have their own merits, it may be necessary to construct (or deconstruct) `MATRIX` and `VECTOR` types using SQL. To facilitate this, we introduce the notion of a *label*. In our extension, each `VECTOR` attribute implicitly or explicitly has an integer label value attached to it (if the label is never explicitly set for a particular vector, then its value is -1 by default). In addition, we introduce a new type called `LABELED_SCALAR`, which is essentially a `DOUBLE` with a label. Using those labels along with three special aggregate functions (`ROWMATRIX`, `COLMATRIX`, and `VECTORIZE`), it is possible to write SQL code that creates `MATRIX` types and `VECTOR` types, respectively, from normalized data.

For example, reconsider the table:

```
CREATE TABLE y (i INTEGER, y_i DOUBLE);
```

Imagine that we want to create a table with a single vector tuple from the table y . To do this, we simply write:

```
SELECT VECTORIZE (label_scalar (y_i, i))
FROM y
```

Here, the `label_scalar` function creates an attribute of type `LABELED_SCALAR`, attaching the label i to the

`DOUBLE y_i`. Then the `VECTORIZE` operation aggregates the resulting values into a vector, adding each `LABELED_SCALAR` value to the vector at the position indicated by the label. Any “holes” (or entries in the vector for which no `LABELED_SCALAR` were found) in the resulting vector are set to zero.

As stated above, `VECTOR` attributes implicitly have labels, but they can be set explicitly, and those labels can be used to construct matrices. For example, imagine that we want to create a single tuple as a single matrix from the table:

```
mat (row INTEGER, col INTEGER, value DOUBLE)
```

We can do this with the following SQL code:

```
CREATE VIEW vecs (vec, row) AS
SELECT VECTORIZE (label_scalar (val, col) )
  AS vec, row
FROM mat
GROUP BY row
```

followed by:

```
SELECT ROWMATRIX (label_vector (vec, row) )
FROM vecs
```

The first bit of code creates one vector for each row and the second bit of code aggregates those vectors into a matrix, using each vector as a row. It would have been possible to create a column matrix by first using a `GROUP BY col` and then `SELECT COLMATRIX`.

So far, we have discussed how to de-normalize relations into vectors and matrices. It is equally easy to normalize `MATRIX` and `VECTOR` types. Assuming the existence of a table `label (id)` which simply lists the values $1, 2, 3$, etc., one can move from the vectorized representation (found in the `vecs` view defined above) to a purely-relational representation using a `join` of the form:

```
SELECT label.id, get_scalar (vecs.vec, label.id)
FROM vecs, label
```

Code to normalize a matrix is written similarly.

4. IMPLEMENTATION

4.1. Underlying database

We have implemented all of these ideas on top of the SimSQL distributed database system.⁹ SimSQL is a prototype database system designed to perform scalable numerical and statistical computations over large data sets, written mostly in Java, with a C/C++ foreign function interface.

In this section, we describe some details regarding our implementation. In building linear algebra capabilities into SimSQL, our mantra was “incremental, not revolutionary”. Our goal was to see whether, with a small set of changes, a relational database system could be a reasonable platform for distributed linear algebra.

4.2. Distributed matrices?

One of the very first questions that we had to ask ourselves when architecting the changes to SimSQL to support vectors

and matrices was: should we allow individual matrices stored in an RDBMS to be large enough to exceed the size of RAM available on one machine?

After a lot of debate, we decided that, in keeping with a traditional RDBMS design, SimSQL would enforce a requirement that all vectors and matrices should be small enough to fit into the RAM of an individual machine, and that individual vectors and matrices would *not* be distributed across multiple machines. As our mantra was “incremental, not revolutionary,” we did not want to replace database tables with new linear algebra types—which would effectively give us an array database system. Thus, vectors/matrices are stored as attributes in tuples. And as distributing individual tuples or attributes across machines (or having individual tuples larger than the RAM available on a machine) is generally not supported by modern database systems, it seemed reasonable not to support this in our system.

Of course, one might ask, *what if one has a matrix that is too large to fit into the RAM of an individual machine?* This might be a reasonably common use case, and it would be desirable to support very large matrices. Fortunately, it turns out that one can still handle efficient operations over very large matrices using an RDBMS with our extensions. For example, a large, dense matrix with 100,000 rows and 100,000 columns that require nearly a terabyte to store in all can be stored as one hundred tuples in the table:

```
bigMatrix (tileRow INTEGER, tileCol INTEGER,
           mat MATRIX[10000][10000])
```

Efficient, distributed matrix operations are then easily possible via SQL. For example, to multiply `bigMatrix` with `anotherLargeMat`:

```
anotherLargeMat (tileRow INTEGER,
                 tileCol INTEGER, mat MATRIX[10000][10000])
```

We would use:

```
SELECT lhs.tileRow, rhs.tileCol,
       SUM (matrix_matrix_multiply (lhs.mat, rhs.mat))
FROM bigMatrix AS lhs, anotherLargeMat AS rhs
WHERE lhs.tileCol = rhs.tileRow
GROUP BY lhs.tileRow, rhs.tileCol
```

The resulting, very efficient computation is identical to what one would expect from a distributed matrix engine.

```
SELECT *
FROM matrix_matrix_multiply (bigMatrix, anotherLargeMat)
```

4.3. Storage

Given such considerations, storage for vectors and matrices is quite simple. Vectors are stored in dense fashion, as lists of double-precision values, along with an integer label (because, as described in the previous section, all vectors are labeled with a row or a column number so that they can be used to construct matrices). This may sometimes represent a waste if vectors are indeed sparse, but if necessary, vectors can easily be compressed before being written to secondary storage.

Matrices, on the other hand, are stored as sparse lists of vectors, using a run-length encoding scheme (missing vectors are treated as consisting entirely of zeros). As described previously, matrices can be stored as lists of column vectors or lists of row vectors; the exact storage format is specified during matrix construction (via either the `ROWMATRIX` or `COLMATRIX` aggregate function).

4.4. Algebraic operations

SimSQL is written mostly in Java, which presented something of a problem for us when implementing linear algebra operations: some readers of this paper will no doubt disagree, but after much examination, we felt that Java linear algebra packages still lag behind their C/FORTRAN contemporaries in terms of raw performance. Although a high-performance C implementation is (in theory) available to a Java system via JNI, passing through the Java/C barrier typically requires a relatively expensive data copy.

The solution that we implemented is, in the end, a compromise. We decided not to use any Java linear algebra package. The majority of SimSQL’s built-in linear algebra operations (indeed, the majority of *any* linear algebra system’s built-in operations), are simple and easy to implement efficiently: extracting/setting the diagonal of a matrix, computing the outer product of two vectors (which is of linear cost in the size of the output matrix), scalar/matrix and scalar/vector multiplication, etc. All such “simple” operations are implemented in Java, directly on top of our in-memory representation.

There is, however, another set of operations (matrix inverse, matrix-matrix multiply, etc.) that are much more challenging to implement in terms of achieving good performance and dealing with numerical instabilities. For those operations, we use SimSQL’s *foreign function* interface to transform vector- and matrix-valued inputs into C++ objects, where we then use BLAS implementations.

4.5. Aggregation

The extensions proposed in this paper require two new types of aggregation. First, we must be able to perform standard aggregate computations (`SUM`, `AVERAGE`, `STD_DEV`, etc.) over vectors and matrices. As, in SimSQL, these standard aggregate computations are all written in terms of basic arithmetic operations (`+`, `-`, `*`, etc.), the standard aggregate computations over vectors and matrices all happen “for free” without any additional modifications.

Second, our extensions need a few new aggregate functions with special semantics: `VECTORIZE`, `ROWMATRIX`, and `COLMATRIX`. The first constructs a vector out of a set of `LABELLED_SCALAR` objects. The latter two construct a matrix out of a set of vectors. All are implemented within the system via hashing. For example, in the case of `VECTORIZE`, all of the `LABELLED_SCALAR` objects used to build the vector are collected in a hash table (in the case of a `GROUP BY` clause, there would be many such hash tables). As aggregation is performed in a distributed manner, hash tables from different machines that are being used to create the same vector will need to be merged into a single hash table on a single machine. Merging may also need to happen if there

are enough groups during aggregation so that memory is exhausted; in this case, a partially-complete hash table may need to be flushed to disk.

Once all of the `LABELED_SCALAR` objects for a vector have been collected into a single hash table, the objects are sorted based on the position labels, and are then converted into a vector. Any missing entries are treated as zero, and the length of the resulting vector is equal to the largest label used to construct the vector.

Matrices are constructed similarly, with one change being that the objects hashed to construct the matrix are `VECTOR` objects, rather than `LABELED_SCALAR` objects. Note that by definition, all `VECTOR` objects are labeled, and it is those labels that are used to perform the aggregation.

5. EXPERIMENTS

In this section, we experimentally test whether these extensions can, in fact, result in a performant distributed linear algebra system. In the first set of experiments, we compare the efficiency of our SimSQL linear algebra implementation with several alternative platforms, on a set of relatively straightforward compilations. In the second set of experiments, we evaluate the utility of our extensions for implementing very large-scale deep learning.¹

We stress that this is not a “which system is faster?” comparison. SimSQL is implemented in Java and runs on top of Hadoop MapReduce, with the high latency that implies. A commercial system would be much faster. Rather, our goal is simply to ask: is an RDBMS a viable platform for running distributed linear algebra?

Platforms tested. The platforms we evaluated are:

- (1) SimSQL. We tested several different SimSQL implementations: Without vector/matrix support (the original SimSQL implementation without our extensions), with data stored as vectors, and with data stored as vectors, then converted into blocks.
- (2) SystemML. This is SystemML V1.2.0, which runs on *Spark-Batch* mode. All computations are written in SystemML’s DML programming language.
- (3) SciDB. This is SciDB V18.1. All computations are written in SciDB’s AQL language which is similar to SQL.
- (4) `Spark mllib.linalg`. This is run on Spark V2.4 in standalone mode. All computations are written in Scala.
- (5) TensorFlow. This is TensorFlow V0.12.0. All computations are written in Python.

Computations performed. In our first set of experiments, we performed three different representative computations.

- (1) Gram matrix computation. A Gram matrix is the inner products of a set of vectors. It is a common computational pattern in machine learning, and is often used to compute the kernel functions and

covariance matrices. If we use a matrix \mathbf{X} to store the input vectors, then the Gram matrix \mathbf{G} can be calculated as $\mathbf{G} = \mathbf{X}^T \mathbf{X}$.

- (2) Least squares linear regression. Given a paired data set $\{y_i, \mathbf{x}_i\}$, $i = 1, \dots, n$, we wish to model each y_i as a linear combination of the values in \mathbf{x}_i . Let $y_i \approx \mathbf{x}_i^T \beta + \epsilon_i$, where β is the vector of regression coefficients. The most common estimator for β is the least squares estimator: $\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$.
- (3) Distance computation. We first compute the distance between each data point pair \mathbf{x}_i and \mathbf{x}' : $d_A^2(\mathbf{x}_i, \mathbf{x}') = \mathbf{x}_i^T \mathbf{A} \mathbf{x}'$. Then, for each data point \mathbf{x}_i , we compute the minimum $d_A^2(\mathbf{x}_i, \mathbf{x}')$ value over all $\mathbf{x}' \neq \mathbf{x}_i$. Lastly, we select the data points which have the max value among those minimums.

In our second set of experiments, we use a Wikipedia dump of 4.86 million documents to learn how to predict the year of the last edit to a Wikipedia article. There are 17 possible labels in total. We pre-process the Wikipedia dump, representing each document as a 60,000-dimensional feature vector, where each feature corresponds to the number of times a particular unigram or bigram appears in the document. This is input into a two-layer feed-forward neural network (FFNN). In most of our experiments, we use 10,000 as the batch size, as recent results indicate that a relatively large batch of this size is a reasonable choice for large-scale learning.¹³

Implementation details. A SimSQL programmer uses queries and built-in functions to implement computations. For the first set of experiments for SimSQL, we implemented each model using three different SQL codes. First, we wrote a pure-tuple-based code (as on an existing, standard SQL-based platform). Second, we wrote an SQL code where each data point is stored as an individual vector. Third, we wrote an SQL code where data points are grouped together in blocks, and are stored as matrices so that they can be manipulated as a group. For FFNN learning, we used only blocked matrices.

In SystemML, data is stored and processed as blocks, which are square matrices. All code is written using SystemML’s Python-like programming language. In `Spark mllib.linalg`, we carefully tuned our implementation to answer questions such as: should the input data be stored/processed as vectors, or as matrices? And, if a matrix is used, should it be a local matrix, or a distributed one? For example, for the Gram matrix computation and linear regression, the vector-based implementation is the fastest. Data in SciDB is partitioned as chunks. We use 1000 as the chunk size for all arrays.

Experiment setup. We ran the first set of experiments on 10 Amazon EC2 `r5d.2xlarge` machines, each having eight CPU cores, 64 GB of RAM, and a 300GB SSD drive. For Gram matrix computation and linear regression, the number of data points per machine was 10^5 . For the distance computation, the number of data points per machine was 10^4 . All data sets were dense, and all the data was synthetic—as we are only interested in running time; there is likely no practical difference between synthetic and real data. For each computational task, we considered three data dimensionalities: 10, 100, and 1000. We ran the FFNN experiments

¹ Using RDBMS-based linear algebra for deep learning is considered in detail in Jankov et al.¹⁵; the experimental results given here are taken from that paper.

on 5, 10, and 20 Amazon EC2 r5d.2xlarge machines, and tested the neural network with different number of neurons in the hidden layer.

Experiment results and discussion. The results of the first set of experiments are shown in Figures 2–4, and the results of the FFNN experiments are shown in Figure 5.

In the first set of experiments, we see that vector- and block-based SimSQL clearly dominate the tuple-based implementation for each of the three computations. The results show that it is simply not possible to move enough tuples through a database system to fulfill large-scale linear algebra operations using only tuples.

For linear regression and Gram matrix, we see that the vector-based computation was faster than block-based for 10- and 100-dimensional computations. This is because our experiments counted the time of grouping vectors into blocked matrices. This additional computation was not worthwhile for less computationally expensive problems. But for the 1000-dimensional computations, additional time savings could be realized via blocking.

For the higher-dimensional problems, there was no clear winner among block-based SystemML and SimSQL (the former being a tiny bit faster for linear regression and Gram

Figure 2. Gram matrix results. Format is MM:SS.

Gram Matrix Computation			
Platform	10 dims	100 dims	1000 dims
Tuple SimSQL	00:48	02:25	Fail
Vector SimSQL	00:18	00:23	02:48
Block SimSQL	00:39	00:41	01:13
SystemML	00:01	00:02	01:03
Spark <code>mllib</code>	00:15	00:44	15:00
SciDB	00:02	00:08	03:46

Figure 3. Linear regression results. Format is MM:SS.

Linear Regression			
Platform	10 dims	100 dims	1000 dims
Tuple SimSQL	02:11	03:48	Fail
Vector SimSQL	00:28	00:33	02:55
Block SimSQL	00:41	00:44	01:06
SystemML	00:01	00:02	01:04
Spark <code>mllib</code>	00:22	00:47	15:10
SciDB	00:06	00:16	04:41

Figure 4. Distance computation results. Format is MM:SS.

Distance Computation			
Platform	10 dims	100 dims	1000 dims
Tuple SimSQL	Fail	Fail	Fail
Vector SimSQL	03:19	03:56	11:31
Block SimSQL	01:09	01:09	01:21
SystemML	01:01	01:05	03:39
Spark <code>mllib</code>	01:43	02:00	05:51
SciDB	19:20	19:34	23:13

Figure 5. Average iteration time for FFNN learning, using various CPU cluster and hidden layer sizes.

FFNN		
Hidden Layer Neurons	RDBMS	TensorFlow
Cluster with 5 workers		
10000	05:39	01:36
20000	05:46	03:38
40000	08:30	09:02
80000	24:52	Fail
160000	Fail	Fail
Cluster with 10 workers		
10000	04:53	00:54
20000	05:32	02:00
40000	07:41	04:59
80000	17:46	Fail
160000	44:21	Fail
Cluster with 20 workers		
10000	04:08	00:32
20000	05:40	01:12
40000	06:13	02:56
80000	12:55	Fail
160000	25:00	Fail

matrix, the latter being considerably faster for the distance computation). SimSQL was slower for the lower-dimensional problems because as a prototype system, it is not engineered for high throughput. Spark `mllib` and SciDB were not competitive on the higher-dimensional data.

For FFNN learning (Figure 5), SimSQL was slower than TensorFlow in most cases, but it scaled well, whereas TensorFlow crashed (due to memory problems) on a problem size of larger than 40,000 hidden neurons. In TensorFlow, there is no automatic way to distribute matrices across machines, and for the bigger problem sizes, the weight matrices are very large (the problem with 160,000 hidden neurons uses 102 GB weight matrices). Although a distributed database can easily handle data of this size by distributing it across machines or using the local disk to buffer data, TensorFlow lacks such capability.

Micro-benchmarks showed that for the 40,000-hidden-neuron problem, all of the matrix operations required for an iteration of FFNN learning took 6 min, 17 s (6:17) on a single machine. Assuming a perfect speedup, the learning should take just 1:15 per iteration on a five-machine cluster. However, SimSQL took 8:30 and TensorFlow took 9:02. This shows that both systems incur significant overhead, at least at such a large model size. SimSQL, in particular, requires a total of 61 s per FFNN iteration just starting up and tearing down Hadoop jobs. Also in Hadoop, each intermediate result that cannot be pipelined must be written to disk, and it causes a significant amount of I/O. A faster database could likely lower this overhead significantly.

One may wonder: how would TensorFlow have worked were GPUs were used instead? Using a similar dollars-per-hour budget, we ran TensorFlow on several AWS GPU clusters (using a combination of p3.2xlarge and r5.4xlarge machines). At the same cost-per-hour as the five-worker CPU cluster, TensorFlow ran an iteration in 24 s for 10,000

neurons, and failed at all other sizes. At the same cost as the 10-worker cluster, it ran an iteration in 15 s for 10,000 neurons, again failing at all other sizes. And at the same cost as the 20-worker cluster, the time was 12 s, failing for all other sizes. The reason for TensorFlow's failure to run at more than 10,000 neurons is the limited memory available on a modern GPU. Again, TensorFlow does not page data on and off of a GPU, and so it cannot easily be used to learn larger models.

6. RELATED WORK

There has been recent interest in the construction of special purpose data management systems for scalable linear algebra. SystemML¹² was evaluated in this paper. Another good example is the Cumulon system¹⁴, which has the notable capability of optimizing its own hardware settings in the cloud. MadLINQ¹⁸, built on top of Microsoft's LINQ framework, can also be seen as an example of this. Other work aims at scaling statistical/numerical programming languages such as R. Ricardo¹¹ aims to support R programming on top of Hadoop. Riot²¹ attempts to plug an I/O efficient backend into R to bring scalability.

The idea of moving past relations onto arrays as a database data model, particularly for scientific and/or numerical applications, has been around for a long time. One of the most notable efforts is Baumann and his colleague's work on Rasdaman.⁵ In this paper, we have compared with SciDB⁸, an array database for which linear algebra is a primary use case.

There is some support for linear algebra in modern, commercial relational database systems (such as Oracle Database). But that support is not well-integrated into the declarative (SELECT-FROM-WHERE) interface of SQL, and is generally challenging to use. For example, Oracle provides the UTL_NLA² package to support BLAS and LAPACK operations. To multiply two matrices using this package, and assuming two input matrices m1 and m2 declared as type utl_nla_array_dbl (and an output matrix res defined similarly), a programmer would write:

```
utl_nla.blas_gemm(
  transa => 'N', transb => 'N', m => 3, n => 3,
  k => 3, alpha => 1.0, a => m1, lda => 3,
  b => m2, ldb => 2, beta => 0.0, c => res,
  ldc => 3, pack => R);
```


This code specifies details about the input matrices, as well as details about the invocation of the BLAS library.

7. CONCLUSION

We conclude the paper by asking the question: have we affirmed the hypothesis at the core of the paper, that a relational engine can be used with little modification to support efficient linear algebra processing? We feel that our experimental evaluation did in fact confirm the hypothesis. SimSQL was not exactly fast, but it was competitive compared to all of the evaluated systems, at least for larger and more complicated problems, even compared with TensorFlow. And given the baked-in efficiencies associated with SimSQL—it is, after all, a Hadoop-based system,

written mostly in Java—the fact that SimSQL did reasonably well argues that a high-performance RDBMS could be a very effective engine for distributed linear algebra processing.

Acknowledgments

Material in this paper has been supported by the NSF under grant nos. 1355998 and 1409543 and by the DARPA MUSE program. 

References

1. Apache spark mllib: <http://spark.apache.org/docs/latest/mllib-datatypes.html>.
2. Oracle corporation: https://docs.oracle.com/cd/B1930-6_01/index.htm.
3. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th [USENIX] Symposium on Operating Systems Design and Implementation ([OSDI] 16, 2016)*, 265–283.
4. Armbrust, M., Xin, R.S., Lian, C., Huai, Y., Liu, D., Bradley, J.K., Meng, X., Kaftan, T., Franklin, M.J., Ghodsi, A., et al. Spark sql: Relational data processing in spark. In *SIGMOD (2015)*, ACM, 1383–1394.
5. Baumann, P., Dehmel, A., Furtado, P., Ritsch, R., Widmann, N. The multidimensional database system rasdaman. In *SIGMOD Record (Volume 27, 1998)*, ACM, 575–577.
6. Blackford, L.S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., et al. *ScaLAPACK Users' Guide, Volume 4*. SIAM, 1997.
7. Boehm, M., Burdick, D.R., Evfimievski, A.V., Reinwald, B., Reiss, F.R., Sen, P., Tatikonda, S., Tian, Y. Systemml's optimizer: Plan generation for large-scale machine learning programs. *IEEE Data Eng. Bull.* 3, 37 (2014), 52–62.
8. Brown, P.G. Overview of SciDB: Large scale array storage, processing and analysis. In *SIGMOD, 2010*, 963–968.
9. Cai, Z., Vagena, Z., Perez, L.L., Arumugam, S., Haas, P.J., Jermaine, C. Simulation of database-valued Markov chains using SimSQL. In *SIGMOD, 2013*, 637–648.
10. Chaudhuri, S. An overview of query optimization in relational systems. In *PODS (1998)*, ACM, 34–43.
11. Das, S., Sismanis, Y., Beyer, K.S., Gemulla, R., Haas, P.J., McPherson, J., Ricardo: integrating R and Hadoop. In *SIGMOD, 2010*, 987–998.
12. Ghoting, A., Krishnamurthy, R., Pednault, E., Reinwald, B., Sindhvani, V., Tatikonda, S., Tian, Y., Vaithyanathan, S. SystemML: Declarative machine learning on mapreduce. In *ICDE, 2011*, 231–242.
13. Goyal, P., Dollár, P., Girshick, R.B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *CoRR, 2017*, abs/1706.02677.
14. Huang, B., Babu, S., Yang, J. Cumulon: Optimizing statistical data analysis in the cloud. In *SIGMOD, 2013*, 1–12.
15. Jankov, D., Luo, S., Yuan, B., Cai, Z., Zou, J., Jermaine, C., Gao, Z.J. Declarative recursive computation on an rdbms, or, why you should use a database for distributed machine learning. *PVLDB, 2019*, 12.
16. Lebanon, G. Metric learning for text documents. *IEEE PAMI* 4, 28 (2006), 497–508.
17. Libkin, L., Machlin, R., Wong, L. A query language for multidimensional arrays: Design, implementation, and optimization techniques. In *SIGMOD (1996)*, 228–239.
18. Qian, Z., Chen, X., Kang, N., Chen, M., Yu, Y., Moscibroda, T., Zhang, Z. Madlinq: large-scale distributed matrix computation for the cloud. In *EuroSys (2012)*, ACM, 197–210.
19. Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P., Murthy, R. Hive: A warehousing solution over a map-reduce framework. *Vldb 2, 2* (2009), 1626–1629.
20. Zaharia, M., Chowdhury, M., Franklin M.J., Shenker, S., Stoica, I. Spark: Cluster computing with working sets. In *USENIX HotCloud, 2010*, 1–10.
21. Zhang, Y., Zhang, W., Yang, J. I/o-efficient statistical computing with riot. In *ICDE, 2010*, 1157–1160.

Shangyu Luo, Zekai J. Gao, Luis L. Perez, Dimitrije Jankov, and Christopher Jermaine (sl45@rice.edu, [jacobgao, lperetzp, dimitrijejankov]@gmail.com, cmj4@rice.edu)
Rice University, Houston, TX, USA.

Michael Gubanov [gubanov@cs.fsu.edu]
Florida State University, Tallahassee, FL, USA.

Attention: Undergraduate and Graduate Computing Students

There's an **ACM Student Research Competition (SRC)**
at a SIG Conference of interest to you!



Association for Computing Machinery
Advancing Computing as a Science & Profession



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



"Attending ACM SRC was a transformative experience for me. It was an opportunity to take my research to a new level, beyond the network of my home university. Most important, it was a chance to make new connections and encounter new ideas that had a lasting impact on my academic life. I can't recommend ACM SRC enough to any student who is looking to expand the horizons of their research endeavors."

David Mueller
North Carolina State University | SIGDOC 2018



"Participating in the ACM SRC was a unique opportunity for practicing my presentation skills, getting feedback on my work, and networking with both leading researchers and fellow SRC participants. Winning the competition was a great honor, a motivation to continue working in research, and a useful boost for my career. I highly recommend any aspiring student researcher to participate in the SRC."

Manuel Rigger
Johannes Kepler University Linz, Austria | Programming 2018



"The SRC was a great chance to present early results of my work to an international audience. Especially the feedback during the poster session helped me to steer my work in the right direction and gave me a huge motivation boost. Together with the connections and friendships I made, I found the SRC to be a positive experience."

Matthias Springer
Tokyo Institute of Technology | SPLASH 2018



"I have been a part of many conferences before both as an author and as a volunteer but I found SRC to be an incredible conference experience. It gave me the opportunity to have the most immersive experience, improving my skills as a presenter, researcher, and scientist. Over the several phases of ACM SRC, I had the opportunity to present my work both formally (as a research talk and research paper) and informally (in poster or demonstration session). Having talked to a diverse range of researchers, I believe my work has much broader visibility now and I was able to get deep insights and feedback on my future projects. ACM SRC played a critical role in facilitating my research, giving me the most productive conference experience."

Muhammad Ali Gulzar
University of California, Los Angeles | ICSE 2018



"At the ACM SRC, I got to learn about the work done in a variety of different research areas and experience the energy and enthusiasm of everyone involved. I was extremely inspired by my fellow competitors and was happy to discover better ways of explaining my own work to others. I would like to specifically encourage undergraduate students to not hesitate and apply! Thank you to all those who make this competition possible for students like me."

Elizaveta Tremsina
UC Berkeley | TAPIA 2018



"The ACM SRC was an incredible opportunity for me to present my research to a wide audience of experts. I received invaluable, supportive feedback about my research and presentation style, and I am sure that the lessons I learned from the experience will stay with me for the rest of my career as a researcher. Participating in the SRC has also made me feel much more comfortable speaking to other researchers in my field, both about my work as well as projects I am not involved in. I would strongly recommend students interested in research to apply to an ACM SRC—there's really no reason not to!"

Justin Lubin
University of Chicago | SPLASH 2018



"Joining the Student Research Competition of ACM gave me the opportunity to measure my skills as a researcher and to carry out a preliminary study by myself. Moreover, I believe that "healthy competition" is always challenging in order to improve yourself. I suggest that every Ph.D. student try this experience."

Gemma Catolino
University of Salerno | MobileSoft 2018

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

- ◆ Participants receive: \$500 (USD) travel expenses
- ◆ All Winners receive a medal and monetary award. First place winners advance to the SRC Grand Finals
- ◆ Grand Finals Winners receive a handsome certificate and monetary award at the ACM Awards Banquet

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



[CONTINUED FROM P. 104] are complex, it can be quite challenging to make sure there are no vulnerabilities.

Your research group has been working on some systematic approaches to identifying vulnerabilities in cellular networks based on formal methods.

We use a combination of formal methods and well-known tools like model checkers and cryptographic verifiers. But it requires a lot of domain knowledge, and I am lucky to have some good students who really understand cellular networks. We are also trying to come up with defenses for some of the vulnerabilities, which isn't always trivial—not because of the lack of techniques, but because the cellular network ecosystem is so complex. Then also there are a lot of technical constraints, like, for example, backwards compatibility. But because of that, it's also interesting.

Increasingly, people aren't just concerned with data security, but with data trustworthiness and accuracy, an area you began looking into more than a decade ago.

The problem of data quality has been around forever, because organizations that have a lot of data need to be able to ensure it is up to date, free of errors, consistent, clean, and so forth. In cybersecurity, we have techniques for digitally signing information, so that when you get the data, you can check whether or not someone has tampered with it. But the real problem is that somebody can feed you wrong data from the beginning.

We cast our work in the area of sensor networks. When you have a lot of sensors acquiring data, it may not be practical to verify that each piece of data is correct. But you can assign a trust score, that is, an indicator of trustworthiness, by cross-checking the values obtained from all different sources, and use that score to determine which piece of data you want to use. We did a lot of work along those lines with various technical approaches. Another area we've been working on is provenance, because understanding where data was acquired can help you evaluate its trustworthiness.

Finally, in the era of big data, there is a lot of redundancy in data. People

have worked for many years on the area of data fusion, where you combine different sources of data to cross-validate and detect errors. So I think that, in a way, we have the technical means to solve many of these problems, but of course that may not be enough. In the end, the companies that collect all that data and make it available must be willing to enforce data quality.

Are you still involved with IoT security research?

Yes, we do a lot of work in that area. Right now, we are focusing on the use of a machine learning technique known as reinforcement learning, which allows a device to learn through reward functions. So the device will take an action, and then it will evaluate a certain reward function to see if this action is beneficial, for example, in saving energy. It will learn by itself. This is a very interesting area, and a lot of people in machine learning and AI are working on it. On the other hand, some of these devices can also make changes to the physical world—for example, they can open a door or a window. And already, some studies have shown that when you combine multiple devices together, their combined actions may lead to some unsafe situations. So we are looking into that issue, and specifically, how to control the autonomous learning of the devices to make sure what they learn does not lead to unsafe situations.

We are also looking into some of the IoT communication protocols to assess their vulnerabilities, and then we'll apply our methodologies based on formal methods.

Has working in cybersecurity made you more pessimistic?

To be honest, I'm not pessimistic. Attacks can be very sophisticated, but a lot of data breaches are due to the lack of even basic security measures. If you're the manager of a very sensitive facility like a nuclear power plant, then you must be extra careful. But in most cases, if you follow best practices like access control, authentication, anomaly detection, and so on, you will have a reasonable level of protection.

Leah Hoffmann is a technology writer based in Piermont, NY, USA.

© 2020 ACM 0001-0782/20/8 \$15.00



ACM Transactions on Evolutionary Learning and Optimization (TELO)

ACM Transactions on Evolutionary Learning and Optimization (TELO) publishes high-quality, original papers in all areas of evolutionary computation and related areas such as population-based methods, Bayesian optimization, or swarm intelligence. We welcome papers that make solid contributions to theory, method and applications. Relevant domains include continuous, combinatorial or multi-objective optimization.



For further information and to submit your manuscript, visit telo.acm.org

Q&A

Seeing Light at the End Of the Cybersecurity Tunnel

After decades of cybersecurity research, Elisa Bertino remains optimistic.

ACM ATHENA AWARD recipient Elisa Bertino, a professor at Purdue University and research director of the Cyber Space Security Lab of Purdue's Department of Computer Science, has spent her career trying to ensure the security and integrity of the information that is stored in databases and transmitted over mobile, social, cloud, Internet of Things (IoT), and sensor networks. Here, she talks about how her research interests have evolved and why she's not pessimistic about the future of cybersecurity.

You began your research career in the field of databases, first at the Italian National Research Council, and later as a post-doc at IBM's San Jose Research Laboratory. What drew you to security?

My original interest in security began at IBM, where I was looking into how to protect the data stored in databases. From there, I moved from conventional databases to multilevel security databases and began to collaborate with people in cybersecurity. In a way, it was a continuous movement. What really changed was when I moved to Purdue, where there is a big cybersecurity center and a lot of faculty and students working in cybersecurity. That broadened my research perspective quite a lot.

How did you get interested in access control?

When I was at IBM, I was lucky to work in the group that prototyped a lot of fundamental ideas in the area of relational databases. One prototype—the first prototype of SQL—was called



System R. System R had an access control system to make sure that users could only access the data they were authorized to access, so I learned how these concepts work from inside an actual system.

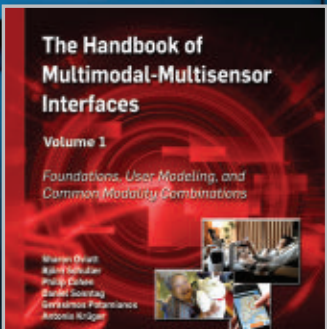
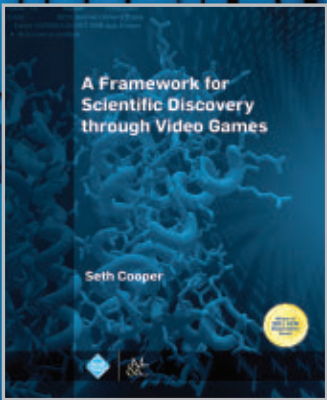
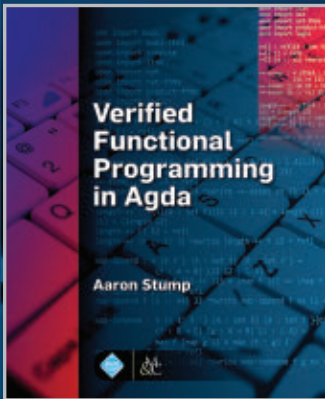
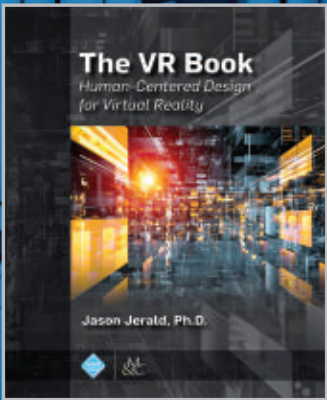
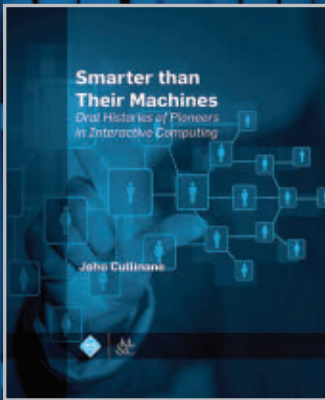
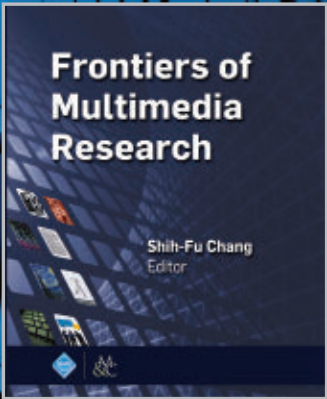
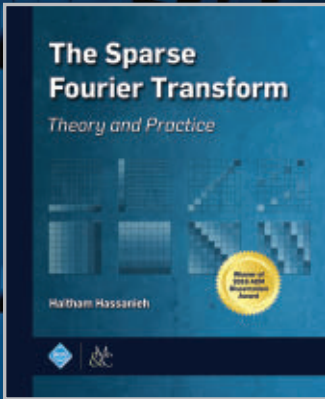
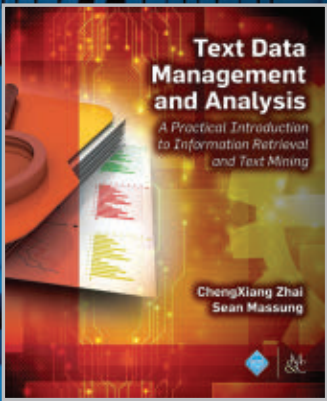
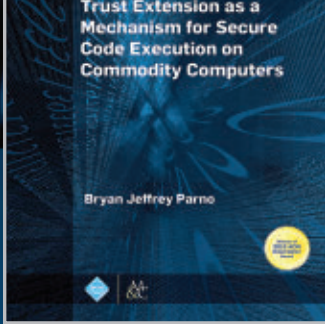
Later, you began to explore how to incorporate temporal and locational constraints into access control.

With the rise of the Internet and mobile systems, it occurred to me that whether you can access an item may also depend on your location or on the time of day. That motivated my work on time- and space-based access control systems. I knew it would be useful one day, and of course location-based access control is now increasingly important because everything is mobile.

Mobility brings both opportunities and challenges when it comes to cybersecurity.

Today's systems are much more open than they were in the past. Companies need to be able to collaborate and share data with other companies, and users expect to have direct access to these resources. Because of that, our systems are very complex. When you add mobile systems and IoT devices and robots into the mix, the complexity is even greater. This is a challenge for security, because you've got to deal with complex protocols involving multiple parties.

One very good example is represented by the protocol for cellular networks, where we have recently been doing a lot of work. The standards that are specified for cellular networks are very complex, because they have to deal with many different operations and situations and parties. Ensuring that protocols are correct is easy if the protocols are simple, but when they [CONTINUED ON P. 103]



In-depth. Innovative. Insightful.

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

**Full Collection | Title List
Now Available**

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



Association for Computing Machinery

1601 Broadway, 10th Floor, New York, NY 10019-7434, USA

Phone: +1-212-626-0658 Email: acmbooks-info@acm.org



SIGGRAPH ASIA 2020 DAEGU

The 13th ACM SIGGRAPH Conference and
Exhibition on Computer Graphics and Interactive
Techniques in Asia

Conference 17 – 20 November 2020

Exhibition 18 – 20 November 2020

EXCO, Daegu, South Korea

Driving Diversity

SA2020.SIGGRAPH.ORG

[#SIGGRAPHAsia](https://twitter.com/SIGGRAPHAsia) | [#SIGGRAPHAsia2020](https://twitter.com/SIGGRAPHAsia2020)



Sponsored by



Organized by

