

COMMUNICATIONS

CACM.ACM.ORG OF THE ACM 01/2019 VOL.62 NO.01



Face2Face: Real-Time Face Capture and Reenactment of RGB Videos

- Quantum Leap
- Illegal Pricing Algorithms
- Intelligent Systems for Geosciences
- Open Collaboration in an Age of Distrust

2019

June 25 – June 30, San Diego, USA

Ai · Blockchain · Cloud · bigData

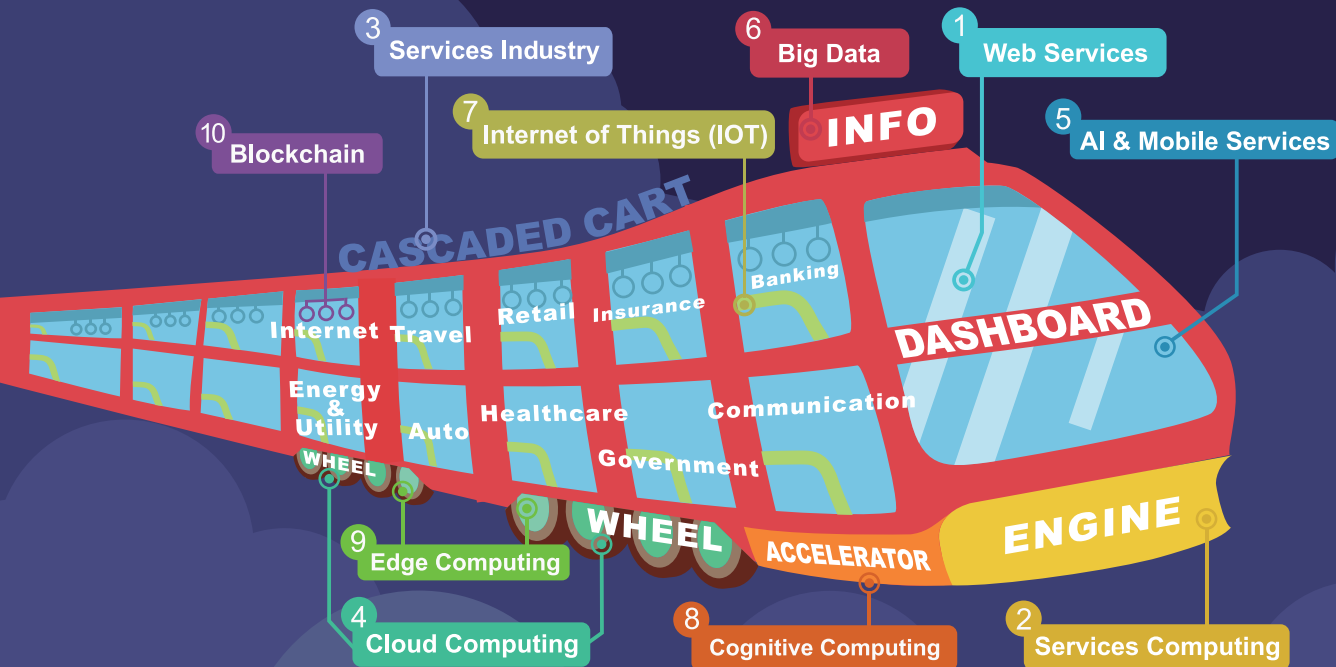
Everything is connected

THEME:
NEW ABCDE

CELEBRATING THE 17th BIRTHDAY OF SCF



- 1 2019 International Conference on Web Services (**ICWS 2019**)
- 2 2019 International Conference on Services Computing (**SCC 2019**)
- 3 2019 World Congress on Services (**SERVICES 2019**)
- 4 2019 International Conference on Cloud Computing (**CLOUD 2019**)
- 5 2019 International Conference on AI & Mobile Services (**AIMS 2019**)
- 6 2019 International Congress on Big Data (**BigData 2019**)
- 7 2019 International Conference on Internet of Things (**ICIOT 2019**)
- 8 2019 International Conference on Cognitive Computing (**ICCC 2019**)
- 9 2019 International Conference on Edge Computing (**EDGE 2019**)
- 10 2019 International Conference on Blockchain (**ICBC 2019**)



Submission Deadlines

1/6/2019: ICWS 2019 (<http://icws.org>)
 1/21/2019: SCC 2019 (<http://theSCC.org>)
 1/25/2019: SERVICES 2019 (<http://ServicesCongress.org>)
 1/6/2019: CLOUD 2019 (<http://theCloudComputing.org>)
 1/21/2019: AIMS 2019 (<http://ai1000.org>)

1/28/2019: BigData 2019 (<http://BigDataCongress.org>)
 2/17/2019: ICIOT 2019 (<http://iciot.org>)
 2/17/2019: ICCC 2019 (<http://theCognitiveComputing.org>)
 2/17/2019: EDGE 2019 (<http://theEdgeComputing.org>)
 2/17/2019: ICBC 2019 (<http://Blockchain1000.org>)



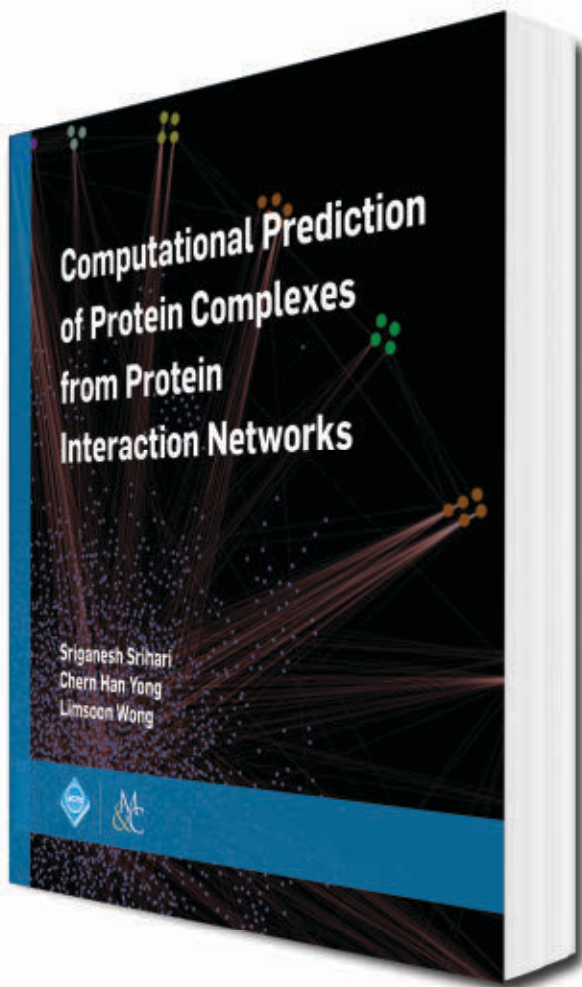
Email:
confs@ServicesSociety.org



Leading not-for-profits organization (US 501(c)(3))
 dedicated for serving 30,000+ worldwide services
 computing professionals



ICWS.ORG



A systematic walkthrough of computational methods for identifying protein complexes from the protein-protein interaction network.

Sriganesh Srihari, *University of Queensland Institute for Molecular Bioscience*
Chern Han Yong, *Duke - National University of Singapore Medical School*
Limsoon Wong, *National University of Singapore*

Complexes of physically interacting proteins constitute fundamental functional units that drive almost all biological processes within cells. A faithful reconstruction of the entire set of protein complexes (the “complexosome”) is therefore important not only to understand the composition of complexes but also the higher level functional organization within cells. In this book, we systematically walk through computational methods devised to date (approximately between 2000 and 2016) for identifying protein complexes from the network of protein interactions (the protein-protein interaction (PPI) network). We present a detailed taxonomy of these methods, and comprehensively evaluate them for protein complex identification across a variety of scenarios including the absence of many true interactions and the presence of false-positive interactions (noise) in PPI networks. Based on this evaluation, we highlight challenges faced by the methods, for instance in identifying sparse, sub-, or small complexes and in discerning overlapping complexes, and reveal how a combination of strategies is necessary to accurately reconstruct the entire complexosome.



ISBN: 978-1-970001-52-5 DOI: 10.1145/3064650
<http://books.acm.org>
<http://www.morganclaypoolpublishers.com/acm>

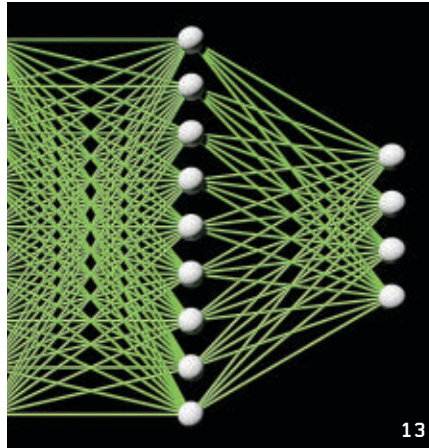
Departments

- 5 **Editor's Letter**
Open Collaboration in an Age of Distrust
By Andrew A. Chien
-
- 6 **Cerf's Up**
A People-Centered Economy
By Vinton G. Cerf
-
- 7 **Vardi's Insights**
Are We Having An Ethical Crisis in Computing?
By Moshe Y. Vardi
-
- 8 **BLOG@CACM**
Answering Children's Questions About Computers
Judy Robertson addresses the disconnect between what children are taught about computers and what they actually wish to know.
-
- 23 **Calendar**
-
- 115 **Careers**

Last Byte

- 120 **Upstart Puzzles**
Randomized Anti-Counterfeiting
By Dennis Shasha

News



- 10 **Quantum Leap**
A new proof supports a 25-year-old claim of the unique power of quantum computing.
By Don Monroe
-
- 13 **Hidden Messages Fool AI**
Forced errors focus attention on neural network quirks.
By Chris Edwards
-
- 15 **Who Owns 3D Scans of Historic Sites?**
Three-dimensional scanning can be used to protect or rebuild historic structures, but who owns that digital data?
By Esther Shein

Viewpoints

- 18 **Law and Technology**
Illegal Pricing Algorithms
Examining the potential legal consequences of uses of pricing algorithms.
By Michal S. Gal
-
- 21 **Technology Strategy and Management**
CRISPR: An Emerging Platform for Gene Editing
Considering a potential platform candidate in the evolving realm of gene-editing technologies research.
By Michael A. Cusumano
-
- 24 **Historical Reflections**
Hey Google, What's a Moonshot? How Silicon Valley Mocks Apollo
Fifty years on, NASA's expensive triumph is a widely misunderstood model for spectacular innovation.
By Thomas Haigh
-
- 31 **Viewpoint**
UCF's 30-Year REU Site in Computer Vision
A unique perspective on experiences encouraging students to focus on further education.
By Niels Da Vitoria Lobo and Mubarak A. Shah
-
- 35 **Viewpoint**
Modeling in Engineering and Science
Understanding behavior by building models.
By Edward A. Lee



Practice

- 38 **Using Remote Cache Service for Bazel**
Save time by sharing and reusing build and test output.
By Alpha Lam
-
- 43 **Research for Practice: Security for the Modern Age**
Securely running processes that require the entire syscall interface.
By Jessie Frazelle
-
- 46 **SQL Is No Excuse to Avoid DevOps**
Automation and a little discipline allow better testing, shorter release cycles, and reduced business risk.
By Thomas A. Limoncelli



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

Contributed Articles

- 50 **Autonomous Tools and Design: A Triple-Loop Approach to Human-Machine Learning**
In addition to having a detailed understanding of the artifacts they intend to create, designers need to guide the software tools they use.
By Stefan Seidel, Nicholas Berente, Aron Lindberg, Kalle Lyytinen, and Jeffrey V. Nickerson
-
- 58 **Framework for Implementing a Big Data Ecosystem in Organizations**
Featuring the various dimensions of data management, it guides organizations through implementation fundamentals.
By Sergio Orenza-Roglá and Ricardo Chalmeta
-
- 66 **The Church-Turing Thesis: Logical Limit or Breachable Barrier?**
In its original form, the Church-Turing thesis concerned computation as Alan Turing and Alonzo Church used the term in 1936—human computation.
By B. Jack Copeland and Oron Shagrir

Review Articles



- 76 **Intelligent Systems for Geosciences: An Essential Research Agenda**
A research agenda for intelligent systems that will result in fundamental new capabilities for understanding the Earth system.
By Yolanda Gil, Suzanne A. Pierce, Hassan Babaie, Arindam Banerjee, Kirk Borne, Gary Bust, Michelle Cheatham, Imme Ebert-Uphoff, Carla Gomes, Mary Hill, John Horel, Leslie Hsu, Jim Kinter, Craig Knoblock, David Krum, Vipin Kumar, Pierre Lermusiaux, Yan Liu, Chris North, Victor Pankratius, Shanan Peters, Beth Plale, Allen Pope, Sai Ravela, Juan Restrepo, Aaron Ridley, Hanan Samet, and Shashi Shekhar



Watch the authors discuss this work in the exclusive *Communications* video.
<https://cacm.acm.org/videos/intelligent-systems-for-geosciences>

- 85 **Deception, Identity, and Security: The Game Theory of Sybil Attacks**
Classical mathematical game theory helps to evolve the emerging logic of identity in the cyber world.
By William Casey, Ansgar Kellner, Parisa Memarmoshrefi, Jose Andre Morales, and Bud Mishra

Research Highlights

- 95 **Technical Perspective**
Photorealistic Facial Digitization and Manipulation
By Hao Li
-
- 96 **Face2Face: Real-Time Face Capture and Reenactment of RGB Videos**
By Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner



Watch the authors discuss this work in the exclusive *Communications* video.
<https://cacm.acm.org/videos/face2face>

- 105 **Technical Perspective**
Attacking Cryptographic Key Exchange with Precomputation
By Dan Boneh
-
- 106 **Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice**
By David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann



About the Cover:
This month's cover story illustrates the essence of Face2Face—an innovative approach for the highly convincing transfer of facial expressions from one source to a target video in real time. Cover illustration by Vault49.



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

Executive Director and CEO

Vicki L. Hanson

Deputy Executive Director and COO

Patricia Ryan

Director, Office of Information Systems

Wayne Graves

Director, Office of Financial Services

Darren Ramdin

Director, Office of SIG Services

Donna Cappel

Director, Office of Publications

Scott E. Delman

ACM COUNCIL

President

Cherri M. Pancake

Vice-President

Elizabeth Churchill

Secretary/Treasurer

Yannis Ioannidis

Past President

Alexander L. Wolf

Chair, SGB Board

Jeff Jortner

Co-Chairs, Publications Board

Jack Davidson and Joseph Konstan

Members-at-Large

Gabrielle Anderst-Kotis; Susan Dumais;

Renée McCauley; Claudia Bauzer Medeiros;

Elizabeth D. Mynatt; Pamela Samuelson;

Theo Schlossnagle; Eugene H. Spafford

SGB Council Representatives

Sarita Adve; Jeanna Neefe Matthews

BOARD CHAIRS

Education Board

Mehran Sahami and Jane Chu Prey

Practitioners Board

Terry Coatta and Stephen Ibaraki

REGIONAL COUNCIL CHAIRS

ACM Europe Council

Chris Hankin

ACM India Council

Abhiram Ranade

ACM China Council

Wenguang Chen

PUBLICATIONS BOARD

Co-Chairs

Jack Davidson; Joseph Konstan

Board Members

Phoebe Ayers; Edward A. Fox; Chris Hankin;

Xiang-Yang Li; Nenad Medvidovic;

Sue Moon; Michael L. Nelson;

Sharon Oviatt; Eugene H. Spafford;

Stephen N. Spencer; Divesh Srivastava;

Robert Walker; Julie R. Williamson

ACM U.S. Public Policy Office

Adam Eisgrau,

Director of Global Policy and Public Affairs

1701 Pennsylvania Ave NW, Suite 300,

Washington, DC 20006 USA

T (202) 659-9711; F (202) 667-1066

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

acmhhelp@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

2 Penn Plaza, Suite 701, New York, NY

10121-0701

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)

2 Penn Plaza, Suite 701

New York, NY 10121-0701 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

Lihan Chen

cacm.deputy.to.eic@gmail.com

SENIOR EDITOR

Moshe Y. Vardi

NEWS

Co-Chairs

Marc Snir and Alain Chesnais

Board Members

Monica Divitini; Mei Kobayashi;

Michael Mitzenmacher; Rajeev Rastogi;

François Sillion

VIEWPOINTS

Co-Chairs

Tim Finin; Susanne E. Hambrusch;

John Leslie King; Paul Rosenbloom

Board Members

Stefan Bechtold; Michael L. Best; Judith Bishop;

Andrew W. Cross; Mark Guzdial; Haym B. Hirsch;

Richard Ladner; Carl Landwehr; Beng Chin Ooi;

Francesca Rossi; 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 Frazzelle; 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

William Aiello; Robert Austin; Kim Bruce;

Alan Bundy; Peter Buneman; Jeff Chase;

Carl Gutwin; Yannis Ioannidis;

Gal A. Kaminka; Ashish Kapoor;

Kristin Lauter; Igor Markov; Bernhard Nebel;

Lionel M. Ni; Adrian Perrig; Marie-Christine

Rousset; Krishan Sabnani; m.c. schraefel;

Ron Shamir; Alex Smola; Josep Torrellas;

Sebastian Uchitel; Hannes Werthner;

Reinhard Wilhelm

RESEARCH HIGHLIGHTS

Co-Chairs

Azer Bestavros and Shiram Krishnamurthi

Board Members

Martin Abadi; Amr El Abbadi; Sanjeev Arora;

Michael Backes; Maria-Florina Balcan;

David Brooks; Stuart K. Card; Jon Crowcroft;

Alexei Efros; Bryan Ford; Alon Halevy;

Gernot Heiser; Takeo Igarashi; Sven Koenig;

Greg Morrisett; Tim Roughgarden;

Guy Steele, Jr.; Robert Williamson;

Margaret H. Wright; Nikolai Zeldovich;

Andreas Zeller

SPECIAL SECTIONS

Co-Chairs

Sriram Rajamani and Jakob Rehof

Board Members

Tao Xie; Kenjiro Taura; David Padua

ACM Copyright Notice

Copyright © 2019 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 acmhhelp@acm.org.

COMMUNICATIONS OF THE ACM

(ISSN 0001-0782) is published monthly by ACM Media, 2 Penn Plaza, Suite 701, New York, NY 10121-0701. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER

Please send address changes to *Communications of the ACM* 2 Penn Plaza, Suite 701 New York, NY 10121-0701 USA

Printed in the USA.



Association for Computing Machinery





Andrew A. Chien

DOI:10.1145/3162391

Open Collaboration in an Age of Distrust

FOR OVER 30 years, computing has been pursued in an environment of trust with computing research advances and publications shared openly within a truly integrated international community. At the heart is the explosive 20-year rise of open source software^a—shared touchstones sufficient to build enterprise-scale software systems and giving rise to multibillion-dollar companies and entire new service sectors.

The bounty of open sharing is the rapid advance of computing technologies—the Internet, WWW, and a wide variety of Internet and cloud services. Equally important, open source sharing has been a boon for education, building an open international community that included developed countries in Europe and North America as well as developing countries such as Brazil, Russia, India, and China. All have contributed and benefitted tremendously in return.

The global backdrop for computing's open sharing was an environment of international trust and secular trend toward global integration of economy and society. We are manifestly in a new era of international relations—"An Age of Distrust"—where the trend toward increased trade and integration has stalled, if not reversed. And, a new superpower competition between the U.S. and China for global scientific, economic, and other forms of leadership is reshaping perspective and strategy.^b

It is time for the computing community to begin thinking and discussing

what it means to engage in open collaboration in an Age of Distrust. Why must the computing community change?

While computing has supported military technology (design) and tactics (gunnery tables) from its earliest days,^c they were not the direct tools of aggression. The evidence is undeniable that computing is now a dual-use technology with capability for direct aggression.

► Cybersecurity technologies are used extensively as instruments of aggression by governments and non-governmental organizations for industrial espionage, sabotage, and subversion of elections,^d and even entire countries' infrastructure. Cybersecurity technology is used for asymmetric attacks on the wealthy and powerful—nations, companies, CEO's, but can also be turned on the poor, weak, and individuals.

► Artificial intelligence technologies have growing capabilities for surveillance, espionage, and more intimidating potential to create autonomous and robotic systems. So serious are these concerns that leading AI researchers have called for a ban on development of autonomous weapons,^e and others have protested and prevented their company's participation in military applications.^f Most countries believe AI is not only commercially important, but also strategic for intelligence and warfare cyberspace and the physical world.

Furthermore, computing's unique capability for instantaneous translation from commercial to military use—download, build, and incorporate—make traditional notions of control^g irrelevant.

Companies face increasing assertion of national sovereignty and control—government access to data, citizen data privacy rights, even information control.^h Universities and research institutes face increasing questions about whom to collaborate with, to share information with, and to allow to work on projects. At issue is the ethical and moral implications of research. Export control regulations proliferate, "deemed export" is increasingly challenging, and new regulations controlling information sharing and research seem likely.

Within science, the physics community has faced these concerns for much of the 20th century, and recently so has the biology community. Within computing, the cryptography community is no stranger to these concerns. We should seek to learn from them.

Let me be clear, I am not advocating banning, control, or classification of research topics. The computing community is too large and international for any single country or organization to limit the progress in computing technologies. However, such efforts will inevitably arise, so we, as computing professionals, must begin the difficult conversations of how to shape the development and use of technologies so that they can be a responsible and accountable force in society.

Let's begin the conversation! ■

Andrew A. Chien, EDITOR-IN-CHIEF

a S. Phipps. Open source software: 20 years and counting, (Feb. 3, 2018), opensource.com

b China v America: The end of engagement, how the world's two superpowers have become rivals. *Economist*, (Oct. 18, 2018); J. Perlez. U.S.-China clash at Asian summit was over more than words. *NY Times*, (Nov. 19, 2018).

c History of Computing Hardware; <https://bit.ly/2IHZgP4>.

d M.S. Schmidt and D.E. Sanger. 5 in China army face U.S. charges of cyberattack. *NY Times*, (May 19, 2014). A. Greenberg. How an entire nation became Russia's test lab for cyberwar. *WIRED*, (June 20, 2017); The untold story of NOTPETYA, the most devastating cyberattack in history. *WIRED*, (Aug. 22, 2018).

e Autonomous weapons: An open letter from AI & robotics researchers; <https://futureoflife.org/open-letter-autonomous-weapons/>

f D. Wakabayashi and S. Shane. Google will not renew Pentagon contract that upset employees. *NY Times*, (June 1, 2018).

g UN Office for Disarmament Affairs. *Treaty on the Non-Proliferation of Nuclear Weapons*; <https://bit.ly/2gxxd2j>

h E.C. Economy. The great firewall of China: Xi Jinping's Internet shutdown, *The Guardian*, (June 29, 2018) and European Union: General data protection regulation; <https://gdpr-info.eu/>



Vinton G. Cerf

DOI:10.1145/3292820

A People-Centered Economy

Innovation for Jobs (i4j.info) recently published a book^a describing a new, people-centered view of work. In some ways, this is a kind of revolutionary Copernican view of work.

Rather than organizing work around tasks, the idea is to organize work around people and their skills. One thesis of this book is that organizing work around tasks leads companies to focus on reducing the cost of tasks by increasing productivity, reducing the need for people to do work. Automation and robotics derive their attraction in part from this incentive. An alternative view seeks to increase the value of people by maximizing their utility and shaping work/jobs around their strengths. I have written before about strengths and noted, in particular, the Gallup Corporation's StrengthsFinder application^b that helps people discover and rank-order the skills and capabilities they have.


As we ponder the future of work, it is important to recognize how essential work is to global socioeconomic conditions and how important it is to the individuals who perform it. In a world in which money is the primary medium of exchange, payment for work is essential. The authors of *The People Centered Economy* recognize that much effort has gone into encouraging people to spend more (think *advertising*), but not so much into helping people earn more (that is, to make themselves more valuable). Meaningful work is fulfilling and payment for it enables people

to support their families and participate in the economy.

In capitalist societies, there is typically a distinction made between *owners* and *workers*. The owners participate in the value of the company while the workers are paid to work. This distinction creates a disparity between these two cohorts, particularly in the case of successful companies. With relatively few exceptions, the workers do not participate in the value of the company except to the extent they are paid for their work. Stockholders (that is, owners) participate in the value of the company. Gallup is an exception, for example, because the company is owned by its employees who participate in the value of the company as well as being paid for their work. Without the efforts of the workers the company would not have value so the idea that the workers and owners ought to be the same cohort has a great deal of attraction. Wealth creation is tied to ownership and the

work that creates value. One can see the attraction of linking these together in the form of owner-workers.

Making people more valuable is also tied to the capacity to produce value. Increasing skills and knowledge increases the potential to do valuable work so education is part of the equation. We are seeing new forms of education emerging, partly through online access to information and partly as a consequence of longer lives and thus longer careers. No longer does it seem possible to learn for a while, earn for a while, and then retire. Careers may extend over periods of six decades or more during which time technology will have changed society and its needs dramatically. Continued learning will be needed during the course of a working career. Indeed, long-lived people may have multiple careers over time.

As we contemplate the future of work, it seems inescapable that technology will play a major role in increasing human ability to do work that is of value to the society. While there is a popular meme today that seeks to demonize automation and robotics, the alternative view is that these technologies will enhance our ability to do productive work. I see them as a means for augmenting our capacity to be productive and innovative, making each of us potentially more valuable to each other and our society. 

No longer does it seem possible to learn for a while, earn for a while, and then retire.

a *The People Centered Economy; The New Ecosystem for Work*. IIIJ Foundation, 2018, ISBN: 1729145922

b <https://www.gallupstrengthscenter.com/home/en-us/benefits-of-cliftonstrengths-34-vs-top-5>

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

Copyright held by author/owner.



Moshe Y. Vardi

DOI:10.1145/3292822

Are We Having An Ethical Crisis in Computing?

THE COMPUTING FIELD WENT through a perfect storm in the early 2000s: the dot-com and telecom crashes, the offshoring scare, and a research-funding crisis. After its glamour phase in the late 1990s, the field seems to have lost its luster, and academic computing enrollments have declined precipitously. This was referred to as the Image Crisis. We seem to be going through another image crisis, of a different nature, these days. Last year the columnist Peggy Noonan described Silicon Valley executives as “moral Martians who operate on some weird new postmodern ethical wavelength.” Niall Ferguson, a Hoover Institution historian, described cyberspace as “cyberia, a dark and lawless realm where malevolent actors range.” Salesforce’s CEO, Marc Benioff, declared: “There is a crisis of trust concerning data privacy and cybersecurity.”

Many view this crisis as an ethical crisis. *The Boston Globe* asserted in March 2018, “Computer science faces an ethics crisis. The Cambridge Analytica scandal proves it!” *The New York Times* reported in October 2018, “Some think chief ethics officers could help technology companies navigate political and social questions.” Many academic institutions are hurriedly launching new courses on computing, ethics, and society. Others are taking broader initiatives, integrating ethics across their computing curricula. The narrative is that what ails tech today is a deficit of ethics, and the remedy, therefore, is an injection of ethics.

This narrative, however, leaves me deeply skeptical. It is not that I am against ethics, but I am dubious of the diagnosis and the remedy. As an example, consider the Ford Model T, the first mass-produced and mass-consumed automobile. The Ford Model T went

into production in 1908 and started the automobile age. With the automobile came automobile crashes, which today kill annually more than 1,000,000 people. But the fatality rate has been going down for the past 100+ years. Reducing the fatality rate has been accomplished by improving the safety of automobiles, the safety of roads, licensing of drivers, drunk-driving laws, and the like. The solution to automobile crashes is not ethics training for drivers, but public policy, which makes transportation safety a public priority.

Last year I wrote^a on how “information freedom” leads Internet companies to use targeted advertising as their basic monetization mechanism, which requires them to collect personal data and offer it to their advertisers. The social scientist Shoshana Zuboff described this business model in 2014 as “surveillance capitalism.” There is a direct line between this business model and the 2018 Facebook–Cambridge Analytica scandal, when it was revealed that Cambridge Analytica collected personal data of millions of people’s Facebook profiles without their consent and used it for political purposes. We must remember, however, that the advertising-based Internet business is enormously profitable. It is unlikely Internet companies will abandon this lucrative business model because of some ethical qualms, even under Apple’s CEO Tim Cook’s blistering attack on the “data industrial complex.”

The problem with surveillance capitalism is not that it is unethical, but that it is completely legal in many countries. It is unreasonable to expect for-profit corporations to avoid profitable and legal business models. In my opinion, the

criticism of Internet companies for “unethical” business models is misguided. If society finds the surveillance business model offensive, then the remedy is public policy, in the form of laws and regulations, rather than an ethics outrage. Of course, public policy cannot be divorced from ethics. We ban human-organ trading because we find it ethically repugnant, but the ban is enforced via public policy, not via an ethics debate.

The IT industry has successfully lobbied for decades against any attempt to legislate/regulate IT public policy under the mantra “regulation stifles innovation.” In response to the investigation of Tesla’s CEO Elon Musk by the U.S. Security and Exchange Commission for possible security-law violation, a recent *Wired* magazine headline proclaimed, “The case against Elon Musk will chill innovation!” Of course regulation chills innovation. In fact, the whole point of regulation is to chill certain kinds of innovation, the kind that public policy wishes to chill. At the same time, regulation also encourages innovation. There is no question that automobile regulation increased automobile safety and fuel efficiency, for example. Regulation can be a blunt instrument and must be wielded carefully; otherwise, it can chill innovation in unpredictable ways. Public policy is hard, but it is better than anarchy.^b

Do we need ethics? Of course! But the current crisis is not an ethics crisis; it is a public policy crisis. **□**

b See Point/Counterpoint debate in the December 2018 issue.

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

Copyright held by author.

a <https://bit.ly/2FvmGGt>

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

twitter

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

DOI:10.1145/3290404

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

Answering Children's Questions About Computers

Judy Robertson addresses the disconnect between what children are taught about computers and what they actually wish to know.



Judy Robertson
What Children Want to Know About Computers

<https://cacm.acm.org/blogs/blog-cacm/231993-what-children-want-to-know-about-computers/fulltext>

what-children-want-to-know-about-computers/fulltext

October 19, 2018

There is a mismatch between what we teach children about computing at school and what they want to know.

More than a decade ago, computer science educators coined the phrase *computational thinking* to refer to the unique cleverness of the way computer scientists approach problem solving. "Our thinking is based on abstraction, decomposition, generalization, and pattern matching," we said, "and everyone will find it useful to think like this in their everyday lives. So please stop asking us to fix your printer." Computational thinking has been a hugely successful idea and is now taught at school in many countries across the world.

Although I welcome the positioning of computer science as a respectable, influential intellectual discipline, in

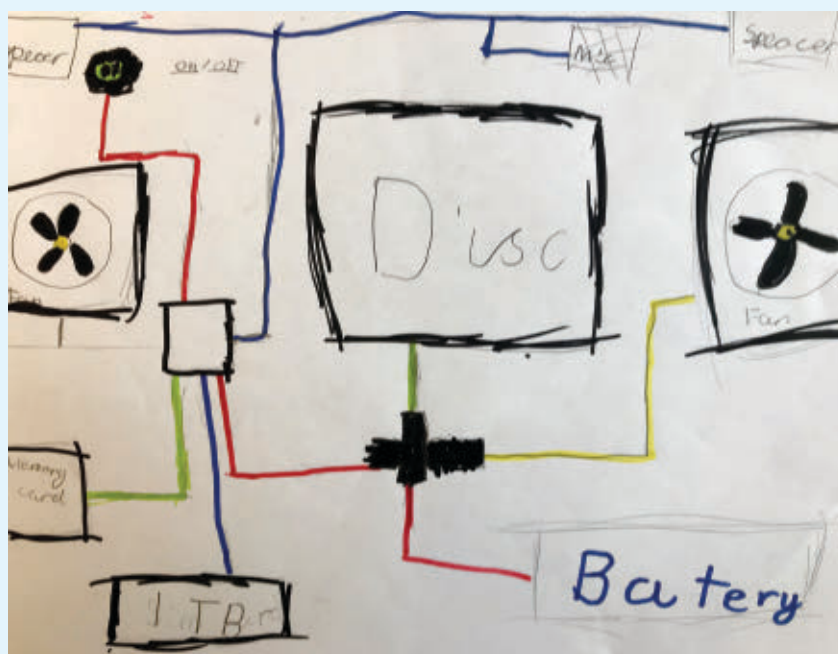
my view computational thinking has abstracted us too far away from the heart of computation—the machine. The world would be a tedious place if

we had to do all our computational thinking ourselves; that is why we invented computers in the first place. Yet, the new school curricula across the world have lost focus on hardware and how code executes on it.

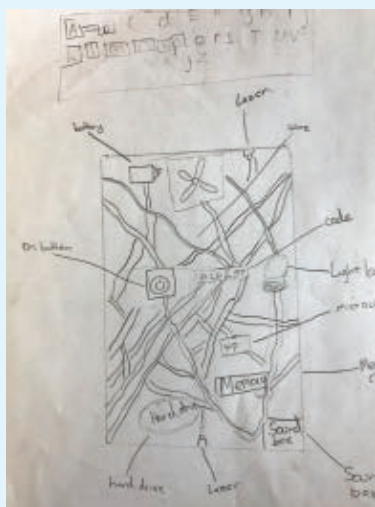
When visiting a series of eight primary school classrooms recently, I talked to children (5 to 12 years old) about how computers work. They drew pictures of what they thought is inside a computer, and then we discussed the drawings as a class.

Many of the children knew the names of the components within a computer: a chip, memory, a disc, and

Drawing 1.



Drawing 2.



they were often insistent that there should be a fan in there. They knew that there would be wires inside, and that it would need a battery to make it work. The child who created Drawing 1 has made a nice job of piecing together a possible design from what they knew about computers—can you spot what is missing, though?

The artist of Drawing 2 knows there is a chip inside (made by HP, in this case) and to their credit, they know there is code, too. Notice that the code is not physically located on the memory or the chip, but somewhere in the wires. In general, there was some puzzlement about how code related to the computer, as exemplified by the artist of Drawing 3, who confessed, “I know a computer is full of code and all devices. I am not sure what it looked like, so I just scribbled.”

Often, the children spent a while thinking about what is outside the computer and how information might get inside. It was quite common to see pictures in which the artist had folded the page to show this distinction but it was often a mystery how pressing a key or touching the screen might make something happen in the computer. Children who had spent time tinkering with computers at home had an advantage here: “I broke my keyboard once and I saw what was inside. It would send a signal from key to computer to the monitor.”

What the pictures and subsequent classroom discussions told me is that the children know names of components within a computer, and possibly

Drawing 3.



some isolated facts about them. None of the pictures showed accurately how the components work together to perform computation, although the children were ready and willing to reason about this with their classmates. Although some of the children had programmed in the visual programming language, none of them knew how the commands they wrote in Scratch would be executed in the hardware inside a computer. One boy, who had been learning about variables in Scratch the previous day, wanted to know whether if he looked in his computer, he would really see apps with boxes full of variables in them. I love that question, because it reveals the mysterious boundary between intangible, invisible information and the small lump of silicon that processes it.

To be clear, I am not criticizing the children, who were curious, interested, and made perfectly reasonable inferences based on the facts they picked up in their everyday lives. But I think that computer science educators can do better here. Our discipline is built upon the remarkable fact that we can write instructions in a representation that makes sense to humans, and then automatically translate them into an equivalent representation that can be followed by a machine dumbly switching electrical pulses on and off. Children are not going to be able to figure that out for themselves by dissecting old computers or by making the Scratch cat dance. We need to get better at explicitly explaining this in interesting ways.

Children are currently piecing to-

gether their everyday experiences with technology with facts that adults tell them to try to make sense of how computers work. This can lead to some confusion, particularly if the adults in their lives are also unsure. One child thought, for example, that if you paid more money, then it would make Wi-Fi stronger. Others were curious about how Wi-Fi works on a train, and whether you really need to stop using your phone on a plane. A student advised the class that if we needed to save space on our phones, then we should delete videos from YouTube. The children, like most Windows users, wanted to know why their computers “freeze,” speculating that it could be because the chip is asleep or that too many people are using Wi-Fi. There was also a sense of wonderment and curiosity. A young boy was fascinated when he read about supercomputers and wanted to know more: Do supercomputers have really big chips in them? A class of 11-year-olds gravely debated whether people would be more or less clever if the computer had never been invented. These are the sorts of questions about computers that children want to explore. It’s our job as computer scientists, and as educators, to help them.

[This article was based on a keynote talk at the Workshop in Primary and Secondary Computing Education (WiPSCE) 2018.]

Judy Robertson is professor of Digital Learning at the University of Edinburgh, U.K.

© 2019 ACM 0001-0782/19/1 \$15.00

Quantum Leap

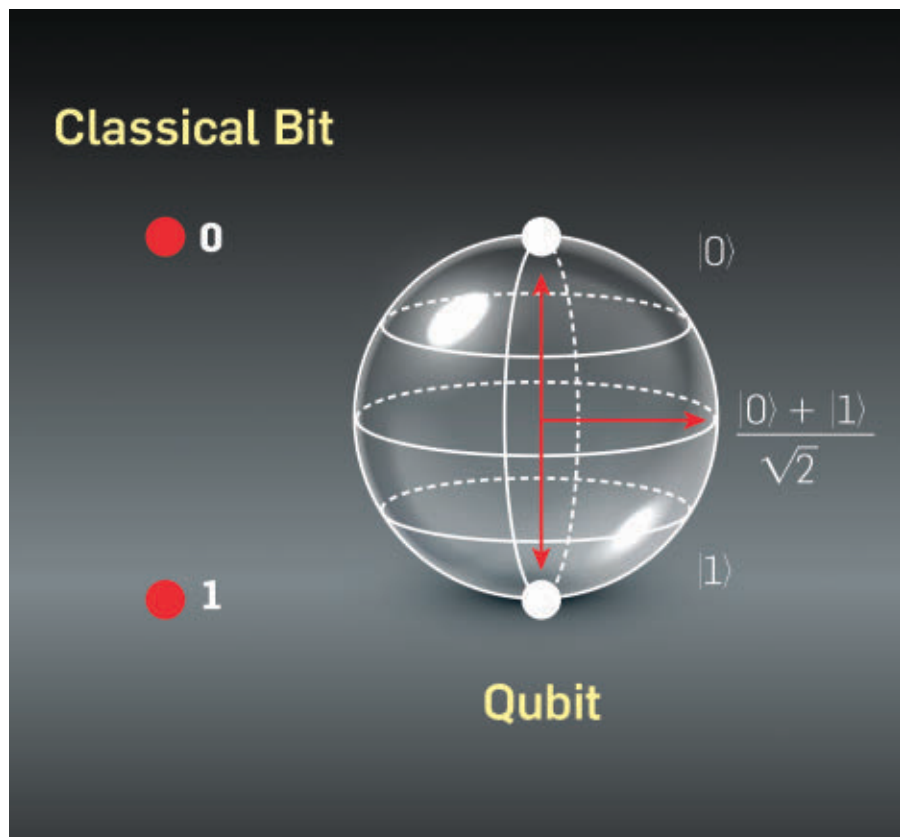
A new proof supports a 25-year-old claim of the unique power of quantum computing.

HOPES FOR QUANTUM computing have long been buoyed by the existence of algorithms that would solve some particularly challenging problems with exponentially fewer operations than any known algorithm for conventional computers. Many experts believe, but have been unable to prove, that these problems will resist even the cleverest non-quantum algorithms.

Recently, researchers have shown the strongest evidence yet that even if conventional computers were made much more powerful, they probably still could not efficiently solve some problems that a quantum computer could.

That such problems exist is a long-standing conjecture about the greater capability of quantum computers. “It was really the first big conjecture in quantum complexity theory,” said computer scientist Umesh Vazirani of the University of California, Berkeley, who proposed the conjecture with then-student Ethan Bernstein in the 1993 paper (updated in 1997) that established the field.

That work, now further validated, challenged the cherished thesis that any general computer can simulate any other efficiently, since quantum computers will sometimes be out of reach of conventional emulation. Quantum computation is the only computation-



al model, then or now, “that violates the extended Church-Turing thesis,” Vazirani said. “It overturned this basic fabric of computer science, and said: ‘here’s a new kid on the block, and it’s completely different and able to do totally different things.’”

Quantum Resources

Conventional “classical” computers store information as bits that can be in one of two states, denoted 0 and 1. In contrast, a quantum degree of freedom, such as the spin of an electron or the polarization of a photon, can exist

concurrently in a weighted mixture of two states. A set of, say, 50 of these “qubits” thus can represent all 2^{50} ($\sim 10^{15}$) combinations of the individual states. Manipulations of this assembly can be viewed as simultaneously performing a quadrillion calculations.

Performing a vast number of computations does not do much good, however, unless a unique answer can be extracted. Interrogating the qubits forces them into some specific combination of 0s and 1s, with probabilities that depend on their post-calculation weights. Critically, however, different initial configurations make quantum contributions to the weight that are complex numbers, which can cancel each other out as well as reinforce each other. The challenge is devising an algorithm for which this cancellation occurs for all configurations except the desired solution, so the eventual measurement reveals this answer.

Soon after Bernstein and Vazirani’s work, mathematician Peter Shor, working at AT&T Research in New Jersey as it spun off from Bell Labs, presented an algorithm that achieved this goal for determining the factors of a large integer. The security of public key cryptography schemes depends on this factorization being impractically time consuming, so the potential for a rapid quantum solution attracted a lot of attention.

Inspired by this and other concrete examples, researchers have been striving to assemble ever-larger systems of physical qubits in the lab that can preserve the delicate complex amplitudes of the qubits long enough to perform a calculation, and to correct the inevitable errors. In recent years, several competing implementations have gotten big enough (dozens of qubits) to achieve “quantum supremacy,” meaning solving selected problems faster than a conventional computer.

Classifying Complexity

Assessing comparative execution times is complicated by the fact that algorithms continually improve, sometimes dramatically. To compare techniques that have yet to be devised and machines that have yet to be built, computer scientists rely not on coding but on formal methods known as computational complexity theory.

These mathematical arguments can determine if an answer can be assured given access to specific resources, such as computational time or the depth of a corresponding circuit. An algorithm that is guaranteed to finish in “polynomial time,” meaning the runtime increases no faster than some fixed power of the size of the input, can be regarded as efficient. In contrast, many problems, notably those that require exhaustively searching many combinatorial possibilities, are only known to yield to methods whose execution time grows exponentially or worse with the size of the input.

Complexity theory divides problems into “complexity classes,” depending on the resources they need. Some of the best-known problems belong to the class P , consisting of problems whose solution can be *found* in polynomial time. A much larger class is NP , which includes problems for which a proposed solution can be *verified* as correct in polynomial time. NP includes such classic challenges as the traveling salesman problem and the graph vertex coloring problem, which researchers have been unable to show belong to P . Many experts strongly suspect that polynomial-time algorithms for many problems in NP have not been found because they do not exist, in which case $P \neq NP$. This important question, regarded by many as the most important open question in theoretical computer science, remains unresolved, and a \$1-million prize from the Clay Mathematics Institute awaits its answer.

To compare techniques that have yet to be devised and machines that have yet to be built, computer scientists rely on computational complexity theory.

ACM Member News

PRESERVING HISTORY IN A DIGITAL LIBRARY



Edward A. Fox, a professor of computer science at Virginia Polytechnic Institute and

State University (Virginia Tech), recalls joining ACM more than 50 years ago.

Fox first became a member of ACM in 1967, while an undergraduate at the Massachusetts Institute of Technology (MIT). During his first year as a member, he launched MIT’s ACM Student Chapter.

In 2017, Fox was named an ACM Fellow, cited for his contributions to information retrieval and digital libraries, the latter a field he helped to launch. “A lot of people don’t know what a digital library is,” Fox says, “So a way to think of it is as an information system tailored to a community of people.”

Fox has served in numerous positions and capacities within ACM over the years. He is currently co-chair (with Michael Nelson) of the ACM Publications Board’s Digital Library and Technology Committee, which works closely with the technical and publishing staff to review services offered by ACM in the context of competing and complementary primary and secondary online resources.

He first became interested in computer science in the mid-1960s when, as a junior in high school, he attended a computer course during a study program at Columbia University. He went on to earn his bachelor’s degree in electrical engineering from MIT, and both his master’s and Ph.D. degrees in computer science from Cornell University.

In the future, Fox hopes the technologies of information retrieval, digital libraries, and archiving will be even better integrated, as a means of helping to preserve our history and achievements for the future.

—John Delaney

Bernstein and Vazirani defined a new complexity class called BQP (Bounded Quantum Polynomial), which has access to quantum resources. BQP is closely analogous to the conventional class BPP (Bounded Probabilistic Polynomial), which has access to a perfect random-number generator and must not give a wrong answer too often. Currently, some problems having only stochastic solutions are known, but it is hoped that deterministic, “de-randomized” algorithms will eventually be found for them.

Consulting the Oracle

The relationship of the quantum class BQP to various conventional classes, however, continues to be studied, long after Bernstein and Vazirani suggested it includes problems beyond the scope of conventional techniques. “We have our conjectures and we can feel strongly about them, but every so often they are wrong,” Vazirani said. “A proof is really something to be celebrated.”

The new proof of separation does not apply to the pure versions of BQP and the other complexity classes addressed by the Vazirani-Bernstein conjecture. Similar to the long-standing unproven relationship of P and NP, “We almost never are able to actually separate these important classes of complexity theory,” said computer scientist Ran Raz of Princeton University in New Jersey and the Weizmann Institute in Israel. “We don’t know how.”

Instead, Raz and his former student Avishay Tal (now at Stanford University) performed what is called an oracle separation. Like its namesake from ancient Greece (or *The Matrix* movies), an oracle provides answers to profound questions without explaining how it got them. Roughly speaking, Raz and Tal compared the capabilities of quantum and classical algorithms that were given access to an oracle that answers a specific question. Provided with this oracle, they showed the quantum system could efficiently solve a carefully chosen problem more efficiently than the classical system could using the same oracle.

Lance Fortnow, a computer scientist at the Georgia Institute of Technology, said hundreds of proofs in complexity theory have relied upon such oracle

“The basic ability to do Fourier transformation, that’s the heart of the power of quantum, at least most of the algorithms we know.”

separations. “They are a way for us to understand what kinds of problems are hard to prove and what kinds of results might be possible, but they’re not a definite proof technique,” he said. “We didn’t prove a separation between these two classes,” Raz agreed. “I can’t imagine that [such a separation] will be proved in our lifetime.”

“Already there were oracle separations of BQP and NP, BQP and P, and other classes,” Raz said. He and Tal now extend the argument to a supercharged class called the polynomial hierarchy, or PH. “This is what is stronger in our result,” he said. PH can be viewed as an infinite ladder of classes, starting with P and NP, in which successive rungs can build on the earlier ones by using logical constructions. Later classes invoke the earlier ones rather like a subroutine, for example by defining problems using them in a phrase such as “for every,” or “there exists.” “Almost all the problems that we encounter in everyday life are somewhere in the polynomial hierarchy,” Raz said.

If all NP problems had polynomial-time solutions, though, it turns out that the entire polynomial hierarchy would collapse into one class, $PH=NP=P$. The new result, though, shows that oracle-assisted BQP would still be separate. “The way I view the Raz-Tal oracle is they’re saying that even if P happened to equal to NP—that’s an unlikely case,” Fortnow said, “it’s still possible that quantum can do more than classical machines can.”

What Is It Good For?

“If we choose the right oracle,” Raz said,

“we show that there is one problem that BQP will solve better than PH.” In addition to choosing the right oracle, he and Tal had to choose a problem that reveals quantum computation’s strength—and classical computation’s weakness—but they only needed one example.

They adapted an earlier suggestion by Scott Aaronson (then at the Massachusetts Institute of Technology) in which the computer must determine if one sequence of bits is (approximately) the Fourier transform of another. Computing such frequency spectra is a natural task for quantum computations, and Shor’s algorithm exploits precisely this strength to identify periodicities that expose prime factors of the target. “The basic ability to do Fourier transformation,” Fortnow said, “that’s the heart of the power of quantum, at least most of the algorithms we know.”

“The hard part is to give the lower bound for the polynomial hierarchy,” Raz said. To show that no such algorithm, even with access to the oracle, could solve it efficiently, he and Tal tweaked Aaronson’s suggestion so they could apply recent discoveries about pseudorandom sequences.

These and the earlier results illustrate what quantum computers will be able to do, once they get very large and perform like the idealized models, Vazirani said. What is less clear is how to effectively use the less-capable machines that are now being developed. “What will we be able to do with those?” he asked. “That’s one of the things that we are working hard to try to figure out.” ■

Further Reading

Bernstein, E. and Vazirani, E. Quantum Complexity Theory, *SIAM J. Comput.* 26, 1411 (1997).

Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal of Computing* 26, pp. 1484–1509 (1997).

Raz, R. and Tal, A. Oracle Separation of BQP and PH, *Electronic Colloquium on Computational Complexity*, Report No. 107 (2018).

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

Hidden Messages Fool AI

Forced errors focus attention on neural network quirks.

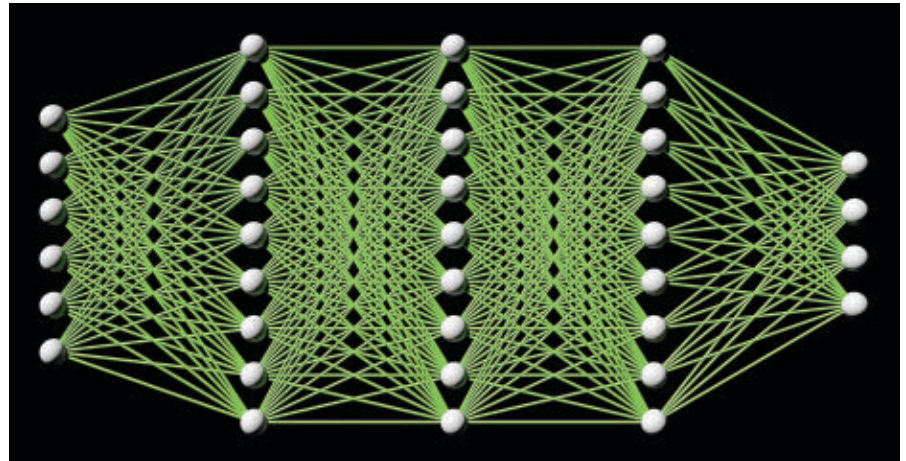
DEEP NEURAL NETWORKS (DNNs) have advanced to the point where they underpin online services from image search to speech recognition, and are now moving into the systems that control robots. Yet numerous experiments have demonstrated that it is relatively easy to force these systems to make mistakes that seem ridiculous, but with potentially catastrophic results. Recent tests have shown autonomous vehicles could be made to ignore stop signs, and smart speakers could turn seemingly benign phrases into malware.

Five years ago, as DNNs were beginning to be deployed on a large scale by Web companies, Google researcher Christian Szegedy and colleagues showed making tiny changes to many of the pixels in an image could cause DNNs to change their decisions radically; a bright yellow school bus became, to the automated classifier, an ostrich.

But the changes made were imperceptible to humans.

At the time, researchers questioned whether such adversarial examples would translate into the physical domain because cameras would smooth out the high-frequency noise mixed into the digitized images that Szegedy and others were presenting directly to their DNNs. Within several years, examples of real-world attacks appeared. In one case, stickers attached to a stop sign made a DNN interpret it as a 45 m.p.h. (miles per hour) sign even though the word ‘stop’ remained clearly visible.

Although most of the research into subverting DNNs using adversarial examples has been within the realm of image recognition and classification, similar vulnerabilities have been found in networks trained for other applications, from malware classification to robot control. Audio systems such as smart speakers seem just as susceptible to attack using the same concepts. Similar to the effects of camera processing on images, the low-pass filtering of microphones and speakers



make some attacks more feasible than others in the real world.

As a Ph.D. student working with David Wagner at the University of California at Berkeley, Nicholas Carlini started looking at fooling speech engines in 2015 as part of a project to examine the vulnerabilities of wearable devices. The UC Berkeley researchers thought practical wearable devices would rely on speech recognition for their user interfaces.

Their focus switched to in-home systems when products such as Amazon’s Echo started to become popular.

“We were able to construct audio that to humans sounded like white noise, that could get the device to perform tasks such as open up Web pages,” says Carlini, now a research scientist at Google Brain. “It was effective, but it was very clear to anyone who heard it that something was going on: you could hear that there was noise.”

In 2017, a team from Facebook AI Research and Bar-Ilan University in Israel showed it was possible to hide messages in normal speech, though a limitation of their so-called Houdini method was that it needed to use replacement phrases, the spoken versions of which were phonetically similar to those being targeted. In November of that year, Carlini found it was possible to push attacks on speech-based systems much further.

“I don’t like writing, and for two or three weeks I had been working on a paper and managed to submit it with 15 minutes to go on the deadline. I woke up the next morning and said, ‘let’s do something fun,’” Carlini explains.

The target was the DeepSpeech engine published as open-source code by Mozilla. “Fifteen hours of work later, I had broken it,” Carlini claims.

Rather than using noise to confuse the system, he had found the engine was susceptible to slightly modified recordings of normal speech or music. The system could be forced to recognize a phrase as something completely different to what a human would hear. The attacks buried subtle glitches and clicks in the speech or music at a level that makes it hard for a human hearing the playback to detect. Some glitches buried in normal phrases convinced the network it was hearing silence.

“I was incredibly surprised it worked so easily. You don’t expect things to break so easily. However, much of it was because I had spent a year and a half on developing attacks to break neural networks in general,” Carlini explains.

However, as a practical attack, the method did not work on audio played through a speaker and into a microphone. Distortions caused by amplifiers and microphones altered the glitches enough to cause the attacks to fail. In Carlini’s version, the adversarial exam-

ples needed to be presented to the DNN in the form of ready-made digitized audio files. This was in contrast to his earlier original attack, in which the added noise survived the filtering of physical speakers and microphones. As with other parts of the adversarial-examples space, the attacks have evolved quickly.

Early in the summer of 2018, a system called CommanderSong developed by a team led by researchers at the Chinese Academy of Sciences demonstrated it was possible to hide voice commands to speech-recognition systems in popular tunes played over the air. The victim system recognizes the altered signals as speech commands.

General concern over the susceptibility of DNNs to adversarial examples grew quickly after Szegedy's work. The attacks seem to work across many different implementations, suggesting there are common factors that make DNNs vulnerable. Numerous low-level countermeasures have been proposed, but almost all have been beaten within months of publication. The problem seems fundamental to systems that can learn.

Humans are susceptible to similar kinds of processing. In experiments intended to find connections between biological perception and AI, Gamaleldin Elsayed and colleagues at Google Brain and Stanford University made subtle changes to images that could fool both humans and DNNs. Neuroscientists believe exposure to images for less than a tenth of a second seems to cut out the brain's ability to use its complex array of feedback networks for recognition. The behavior becomes more consistent with feedforward networks similar to those used in DNNs.

"I don't think humans are perfect and we don't want these systems to be perfect, but we also do not want them to be obviously flawed in ways that humans are not," Carlini says.

Researchers see one reason for DNNs' susceptibility to attack being the enormous number of parameters their layers are called upon to process and how those parameters are set during training. One of the reasons it is so easy to force a misclassification is the way that DNNs perform weighted sums of many individual inputs. Small changes to each pixel in an image can shift the overall result to a different state. Carlini saw a similar effect in his

Misunderstanding the math of high-dimensional spaces may have led to false confidence in the ability of DNNs to make good decisions.

work with speech, but cautions against drawing direct parallels.

"With five seconds of audio, you have as many as 70,000 samples. Messing with only one sample gives you only a small gain. But we get to do it to a lot of samples. The more interesting question is why it is possible that for any target phrase there's a way to get to it without making too much of a change to the audio. I don't have an answer for that, and it is very hard to find a solution to a problem when you don't know why it happens," Carlini says.

The huge number of samples or pixels in the inputs means the DNN has to work on data with a huge number of dimensions. Misunderstanding the mathematics of high-dimensional spaces may have led users to place false confidence in the ability of DNNs to make good decisions. Carlini notes: "Lots of intuition turns out to be completely false in these higher dimensions. It makes things a lot harder to analyze."

In high-dimensional spaces, classifications do not have the clear boundaries we think they do. Relatively small distortions of a large number of pixels or samples in the input image or audio can push a sample from one classification into one of many near neighbors.

At the University of Virginia, work by Ph.D. student Mainuddin Jonas with supervisor David Evans has shown how adversarial examples tend to guide the network away from the correct classification progressively as an image is analyzed by each successive layer of neurons. Reducing the freedom of adversarial examples to push the classification process off course may yield a way to reduce their impact.

In parallel, several groups have looked at ways to harden classification boundaries. They are using the math-

ematics of these higher-dimensional spaces to indicate how best to keep classifications more clearly defined in the trained network. This could lead to methods that detect and discard ambiguous training data and so harden the classification boundaries. But this work remains at an early stage.

Evans has proposed a technique he calls feature-squeezing, which uses techniques such as reducing the bit-resolution of the data processed by neurons. "My goal is to try to reduce the adversary's search space. The perspective that we have on it is to take the high-dimensional search space that the adversaries can currently exploit and try to shrink that," he says. But he notes the problem of tackling adversarial examples and similar attacks will take a lot more effort. "It is definitely an area where there a lot of exciting work going on. We are at the very early stages of what may be a very long arms race."

Carlini believes it will be essential to explore the core mechanisms that drive DNNs to understand how adversarial examples succeed. "I don't think you can construct sound defenses without knowing why they work. We need to step back and figure out what's going on." ■

Further Reading

Szegedy, C., Zaremba, E., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. **Intriguing properties of neural networks** International Conference on Learning Representations 2014. ArXiv:1312.6199 (12/2013)

Carlini, N. and Wagner D. **Audio Adversarial Examples: Targeted Attacks on Speech-to-Text** 1st IEEE Deep Learning and Security Workshop (2018). ArXiv:1801.01944 (3/2018)

Jonas, M.A and Evans D. **Enhancing Adversarial Example Defenses Using Internal Layers** IEEE Symposium on Security and Privacy 2018. [https://www.ieee-security.org/TC/SP2018/poster-abstracts/oakland2018-paper29-poster-abstract.pdf]

Papernot, N. and McDaniel P. **Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning** ArXiv:1803.04765 (3/2018)

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

© 2019 ACM 0001-0782/19/1 \$15.00

Who Owns 3D Scans of Historic Sites?

Three-dimensional scanning can be used to protect or rebuild historic structures, but who owns that digital data?

HIGH ATOP THE Thomas Jefferson Memorial in Washington, D.C., is a layer of biofilm covering the dome, darkening and discoloring it. Biofilm is “a colony of microscopic organisms that adheres to stone surfaces,” according to the U.S. National Park Service, which needed to get a handle on its magnitude to get an accurate cost estimate for the work to remove it.

Enter CyArk, a non-profit organization that uses three-dimensional (3D) laser scanning and photogrammetry to digitally record and archive some of the world’s most significant cultural artifacts and structures. CyArk spent a week covering “every inch” of the dome, processed the data, and returned a set of engineering drawings to the Park Service “to quantify down to the square inch how much biofilm is on the monument,” says CEO John Ristevski.

“This is an example of where data is being used to solve a problem,” to help preserve a historical structure, he says. Ristevski says the Park Service was not charged for the data, and the work CyArk did was funded by individual donors in the San Francisco Bay Area, where the company is located.

CyArk is one of several organizations using 3D scanning to help protect and preserve historic structures from looting, destruction, urbanization, and mass tourism. Iconem, a French start-up founded in 2013, also specializes in the digitization of endangered cultural heritage sites in 3D. Like CyArk, Iconem works on-site with local partners; in its case, in 22 countries. One of those partners is Microsoft Research, and Iconem’s technology utilizes the software giant’s artificial intelligence and computer vision algorithms to integrate multiple levels of photogrammetry



Capturing photogrammetric data for the digital reconstruction of a badly damaged temple in the ancient city of Bagan, in central Myanmar.

data to build extremely precise 3D models, says Yves Ubelmann, an architect who co-founded the company.

This type of work has raised the tricky question of who owns the rights to these digital scans. Officials at organizations involved in utilizing these techniques for historic preservation say they address this up front to avoid any contentious battles later on.

Iconem’s projects are either self-financed or paid for by a client, says Ubelmann. “If Iconem is the sole stakeholder, we share the images with scientific or governmental authorities in the relevant country. They have the right

to use them to raise awareness of their historical sites,” he says. “It is vital to us that countries be able to share their cultural heritage with their citizens and the international community.”

When a client finances a project, the rights to the images are determined on a case-by-case basis, he notes. Iconem works with the client to determine if, how, and where the images can be circulated, but the client retains the rights to the images. “Our ultimate goal is to share the images and models with the widest audience possible while respecting the countries and their heritage.”

Ristevski also says ownership depends on the terms of a contract signed prior to any work being done. However, he adds that regardless of the way the agreement is worded, “the other party gets a free and fully unrestricted license. This is always articulated up front before we hit the ground and do the work. None of this is ambiguous.”

CyArk has been doing this type of work for almost 15 years, in more than 50 countries, “and if we were a bad player, we’d never be allowed back in these countries,” Ristevski says. He stresses that if CyArk owns the scanned data, it is the company’s policy to never monetize it.

CyArk has partnered with Google Arts & Culture on the Open Heritage Project, which is using the laser technology to capture relevant data and store it in Google Cloud. Ristevski thinks “people are suspicious whenever Google gets involved [in a project] and how they might monetize it,” but notes that many museums also work with the search giant’s research division. “There are some beautiful exhibits” housed in Google Cloud, he says, but “because Google is involved, there’s automatically an

There is concern such digital scanning “will recapitulate colonial museum practices that have involved the illicit acquisition of objects from dominated culture groups.”

assumption that there is some evil bent to it.”

Erich Hatala Matthes, an assistant philosophy professor and member of the advisory faculty for environmental studies at Wellesley College, says that from a moral perspective, anyone involved in 3D scanning work should keep the data open and available.

Matthes said three-dimensional scanning projects “that often origi-

nate in Europe and the U.S. and focus on threatened heritage in the Middle East should make every effort to make scans open and accessible to the people and institutions of those countries,” he says. “There is a worry that digital scanning efforts will recapitulate colonial museum practices that have involved the illicit acquisition of objects from dominated cultural groups, and the retention and control of those objects under the banner of preservation.”

Rather than using terms like “shared” or “universal” heritage as licensing claims to ownership or control, Hatala Matthes believes “We should view those ideals in terms of responsibilities, especially to those who are most vulnerable.”

Like CyArk and Iconem, the Institute for Digital Exploration (IDEx) at the University of South Florida (USF) works with local partners on the preservation of culturally sensitive areas that are under threat. “A lot of the work we do aims to help major tourist sites strike a balance between access and preservation,” explains co-founder Michael Decker.

For example, IDEx is working with Vil-

Milestones

Computer Scientists Named Packard Fellows

Two computer scientists were among the 18 early-career academics named 2018 Fellows by the David and Lucile Packard Foundation, each of whom will receive \$875,000 over five years to pursue their research.

The Packard Fellowships in Science and Engineering are among the U.S.’s largest nongovernmental fellowships.

The computer scientists newly named Packard Fellows are:

► Keenan Crane, an assistant professor in the Computer Science Department of Carnegie Mellon University. The Packard organization said Crane “explores how the shapes and motions we observe in nature can be faithfully expressed in a language that is completely finite and discrete, and can hence be understood by a computer. His exploration of this question provides both new foundations for

computation, as well as new ways to turn digital designs into physical, shape-shifting matter.”

► Mahdi Soltanolkotabi, an assistant professor in the Ming Hsieh Department of Electrical Engineering of the University of Southern California. Said the Packard organization, “Soltanolkotabi’s research aims to develop a theoretical foundation for design and analysis of reliable learning algorithms, with applications spanning high-resolution imaging to artificial intelligence.”

ADVE RECEIVES KEN KENNEDY AWARD
ACM and the IEEE Computer Society named Sarita Adve of the University of Illinois at Urbana-Champaign recipient of the 2018 ACM-IEEE CS Ken Kennedy Award, which is aimed at recognizing substantial contributions

to programmability and productivity in computing and significant community service or mentoring contributions.

Adve, the Richard T. Cheng Professor in the Department of Computer Science at University of Illinois at Urbana-Champaign, was cited for her research contributions and leadership in the development of memory consistency models for C++ and Java; for service to numerous computer science organizations; and for exceptional mentoring.

Adve co-developed the memory models for the C++ and Java programming languages (with Hans Boehm, Bill Pugh, and others) based on her early work on data-race-free (DRF) models (with Mark Hill), work that has influenced the worldwide software community and hardware design.

In addition to her teaching, Adve currently serves as chair of the ACM Special Interest Group on Computer Architecture (SIGARCH), as well as on the Defense Advanced Research Projects Agency on the DARPA Information Science and Technology study group.

She was named a Woman of Vision in innovation by the Anita Borg Institute for Women in Technology in 2012, an IEEE Fellow in 2012, and an ACM Fellow in 2010. She also received the SIGARCH Maurice Wilkes Award in 2008.

ACM and IEEE co-sponsor the Ken Kennedy Award, named for the late founder of Rice University’s computer science program and a world expert on high-performance computing. The award is accompanied by a \$5,000 honorarium, which is endowed by the SC Conference Steering Committee.

la Casale in Sicily, a UNESCO World Heritage Site, to both record mosaics there and to apply an advanced technique of digital forensics. This will help officials predict when the priceless mosaics form bubbles in their surface, which leads to their breakdown, says Decker, who also a professor and department chair in USF's College of Arts and Sciences. While the work is done to record history for posterity, it also creates a rich dataset. "We share all this data freely with our host countries and partners," he says.

The work is funded by private grants and the university's arts and sciences department, and the data is stored in the USF library.

"We also provide datasets to the governments with which we work and academic partners," Decker says. "However, the question of ownership is evolving with no clear international standards. We will see a lot of cases of commercial exploitation of cultural heritage being challenged in national courts over the next couple of years."

Decker cites a recent article by Elizabeth Thompson in the *Chapman Law Review* in which she wrote, "The cultural heritage objects in question are not protectable by copyright ... On the other hand, creators of digital models of these non-copyrightable cultural heritage artifacts most probably do have copyright protection."

Ping Hu, a partner and chair of the Intellectual Property Group at Massachusetts law firm Mirick O'Connell, says the issue is pretty clear-cut. "The person who creates the 3D scans is the copyright owner. I don't think there is much dispute about it," he says.

Creating a licensing arrangement is a logical solution, Hu says. For example, in exchange for providing commercial access to a historic site, an entity performing 3D scanning work would give the government a royalty-free license to the scans.

David Myers, senior project specialist and manager of the Getty Conservation Institute (GCI) recording and documentation unit, says there are instances where it makes sense for them to retain the data, such as when there is government upheaval. Myers recalls a field conservation project GCI did several years ago to assess flood risk in Egypt's Valley of the Queens.

"Ultimately, we'd like to see the sites themselves doing the work and adopting the tools and doing this continually."

"The number-one threat to those tombs is flash flooding, even in the desert," he notes. GCI commissioned a team to do laser scanning "because we needed a new and accurate topographic map of the valley and the location of tomb openings" and their elevation, to see how it affects the topography. Then the institute could hire a hydrologist to determine what a flood event would look like and which tombs would be at risk. GCI officials also designed flood-prevention interventions.

The Egyptian government funded the flood protection work, Myers says, but when construction was slated to begin in 2011, a revolution occurred. (The work also was interrupted in 2013). The institute has the scan data on its internal servers, and Myers says he is not aware of any issues over its ownership. "There have been particular projects where our partner has put restrictions on things like images and their use, so that's something we work into our agreements," but this only happens on occasion, he says.

If a local government requests the data, Myers says, "Typically we share [it] with our partners and ... I can't think of any case where there's been any reason why we wouldn't do that."

In some cases, GCI also trains local staff on best practices for conservation of archeological sites, so they can conduct preservation work, including 3D scanning, themselves, he says.

Likewise, Ristevski says when CyArk goes into a country, it offers a one-day workshop to local officials "on almost every project" to learn how to do 3D documentation. For example, his team

trained Syrian professionals in Lebanon on how to use Light Detection and Ranging (LIDAR) and photogrammetry. "They were able to use those skills [to scan] significant sites in Damascus when it wasn't feasible for us to enter the country," he says. "Ultimately, we'd like to see the sites themselves doing the work and adopting the tools and doing this continually," since CyArk is a small organization and can only take on so much.

Then it is up to the local government, or whomever oversees a historic site, to determine what to do with the scans, he says. "That decision should always rest ... with the National Park Service, the country ministries, whoever," Ristevski says. "I don't think it's our role, or even our right, to be able to do that."

Decker concurs. "We share our work both online and with researchers who want the raw data as well," he says. "We provide these to host governments and partners and have the view that this sort of research collaboration is in the spirit of the scientific method and good ethics." **□**

Further Reading

Abbot, E.

Reconstructing History: The Ethical and Legal Implications of 3D Technologies for Public History, Heritage Sites, and Museums, *Huron Research*, July 11, 2016, <http://bit.ly/2QCvsnw>

Mendis, D.

Going for Gold—IP Implications of 3D Scanning & 3D Printing, *CREATE*, Nov. 29, 2017, <http://bit.ly/2Nm8B1B>

Billingsley, S.

Intellectual Property in the Age of 3D Scanning and 3D Printing, *Spar3D*, July 25, 2016, <http://bit.ly/2POhKwL>.

Doctorow, C.

Why 3D scans aren't copyrightable, *Boing Boing*, June 21, 2016, <http://bit.ly/2NnQijq>

3D digitisation and intellectual property rights, *Jisc*, January 17, 2014, <http://bit.ly/2xtl3ls>

Wachowiak, M.J., and Karas, B.V.

3D Scanning and Replication for Museum and Cultural Heritage Applications, *JAIC* 48 (2009), 141–158, <https://s.si.edu/2NYouuN>

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

© 2019 ACM 0001-0782/19/1 \$15.00

Law and Technology

Illegal Pricing Algorithms

Examining the potential legal consequences of uses of pricing algorithms.

ON JUNE 6, 2015, the U.S. Department of Justice brought the first-ever online marketplace prosecution against a price-fixing cartel. One of the special features of the case was that prices were set by algorithms. Topkins and his competitors designed and shared dynamic pricing algorithms that were programmed to act in conformity with their agreement to set coordinated prices for posters sold online. They were found to engage in an illegal cartel. Following the case, the Assistant Attorney General stated that “[w]e will not tolerate anticompetitive conduct, [even if] it occurs...over the Internet using complex pricing algorithms.” The European Commissioner for Competition endorsed a similar position, stating that “companies can’t escape responsibility for collusion by hiding behind a computer program.”

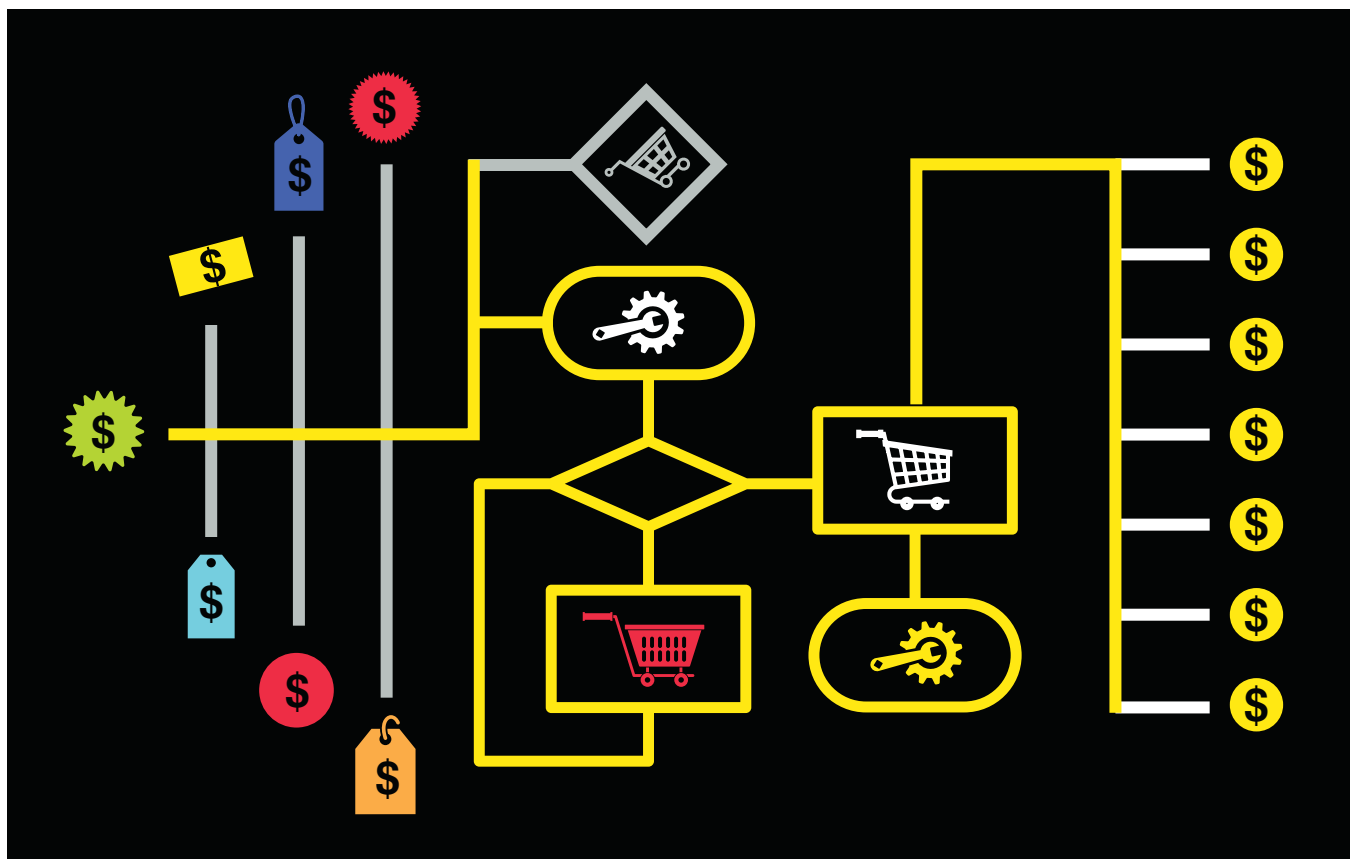
Competition laws forbid market players from engaging in cartels, loosely defined as agreements among market players to restrict competition, without offsetting benefits to the public. This prohibition is based on the idea that competition generally

increases welfare, and that for competition to exist, competitors must make independent decisions. Accordingly, price-fixing agreements among competitors are considered the “ultimate evil” and may result in a jail sentence in the U.S., as well as in other jurisdictions, unless the agreement increases consumers’ well-being.

Until recently, formation of a cartel necessitated human intent, engagement, and facilitation. But with the advent of algorithms and the digital economy, it is becoming technologically possible for computer programs to autonomously coordinate prices and trade terms. Indeed, algorithms can make coordination of prices much easier and faster than ever before, at least under some market conditions. Their speed and sophistication can help calculate a high price that reacts to changing market conditions and benefits all competitors; the speed at which they can detect and respond to deviations from a coordinated high price equilibrium reduces the incentives of competitors to offer lower prices. Indeed, if one algorithm sets a lower price in an attempt to lure more consumers, a competitor’s algorithm

may be designed to immediately respond by lowering its price, thereby shrinking the benefits to be had from lowering the price in the first place. Moreover, as John von Neumann suggested, algorithms serve a dual purpose: as a set of instructions, and as a file to be read by other programs. Accordingly, by reading another algorithm’s accessible source code, algorithms, unlike humans, can determine how other algorithms will react to their own actions, even before any action is performed by the other side. This enables competitors to design their coordinated reactions, even before any price is set.

The questions thus arise when the use of pricing algorithms constitutes an illegal cartel, and whether legal liability could be imposed on those who employ algorithms, as well as on those who design them. The stakes are high: if we cast the net too narrowly and algorithmic-facilitated coordination falls under the radar, market competition may be harmed and prices may be raised; if we cast the net too widely, we might chill the many instances in which algorithms bring about significant benefits.



To prove an illegal cartel, an agreement must be shown to exist. An agreement requires communication among competitors, which signals intent to act in a coordinated way, and reliance on the other to follow suit, in a manner that creates a concurrence of wills. Some scenarios that involve pricing algorithms easily fall within the definition. A simple scenario involves the use of algorithms to implement or monitor a prior agreement among competitors, as was done in the *Topkins* case, mentioned here. In such situations, a clear agreement exists, and the algorithms simply serve as tools for its execution. U.S. Federal Trade Commissioner Maureen Ohlhausen suggested a simple test that captures many of these easy cases: If the word “algorithm” can be replaced by the phrase “a guy named Bob,” then algorithms can be dealt with in the same way as traditional agreements.

A more complicated scenario arises when competitors deliberately use a joint algorithmic price setter, which is designed to maximize the profits of its users. Such a scenario was recently analyzed by Luxembourg’s Competition Authority. There, numerous taxi

drivers jointly used a booking platform that employed an algorithm to determine taxi prices for all participating drivers. The algorithm set the price based on predetermined criteria such as the length of journey, the hour of service, traffic congestion, and so on. The price was non-negotiable. This arrangement was found to constitute an agreement to fix prices. It was nonetheless exempted on the grounds that the efficiencies it generated (including reduction of wait time and lower prices for some consumers) were larger than the harm caused by the coordination, and that these efficiencies could not be achieved by less-restrictive means. Much depends, however, on the specific facts of a given case, including the price formula used by the algorithm and the efficiencies it creates.

Should the algorithm not create large, countervailing benefits for consumers, its employment might constitute an illegal cartel. The U.S. Department of Justice opposed the Google Books Settlement on such grounds. There, Google agreed with the associations of book authors and publishers that a pricing algorithm will set the default prices for the use of Google

Books. The U.S. Authority argued that it is unlawful for competitors to agree with one another to delegate pricing decisions to a common agent, unless the agreement creates countervailing benefits. Interestingly, the fact the pricing algorithm was designed to mimic pricing in a competitive market was regarded as insufficient. Actual bilateral negotiations on book prices were seen as preferable. This argument was not pursued further by the courts.

The more challenging cases arise when algorithms are designed independently by competitors to include decisional parameters that react to other competitors’ decisions in a way that strengthens or maintains a joint coordinated outcome. For example, suppose each firm independently codes its algorithm to take into account its competitors’ probable and actual reactions, as well as their joint incentive to cooperate, and the combination of these independent coding decisions leads to higher prices in the market. Coordination occurs even though no prior agreement to coordinate exists. Even more difficult questions arise when algorithms

are not deliberately designed in a way which facilitates coordination. Rather, the algorithm is given a general goal, such as “maximize profits,” and it autonomously determines the decisional parameters it will use. The interaction between such algorithms may lead to coordination and higher prices. Yet does an illegal agreement exist in such scenarios?

The answer is currently being debated by competition authorities, scholars, and courts worldwide. While it is currently impossible to draw clear bright lines, four basic guidelines already emerge. First, the fact that coordination is achieved through algorithmic interactions does not prevent proof of an agreement. This can be exemplified by the requirement of an intent to engage in an agreement. Obviously, algorithms cannot have a mental state of “intent.” Yet algorithms “intend” to achieve certain goals by using certain strategies, including reaching a coordinated equilibrium with other algorithms. Alternatively, the intent of the designer to create coordination through the use of algorithms, and the intent of the user to employ such algorithms, can sometimes fulfill this requirement. Likewise, while algorithms generally do not sign agreements, shake hands, wink to each other, or nod their consent, they can communicate through the decisional parameters coded into them or set by them in the case of machine learning. Competitors can then rely on such communications when determining their own actions.

Second, the mere use of algorithms does not prevent the imposition of legal liability on their designers and users. As the European Commissioner for Competition stated, “legal entities must be held accountable for the consequences of the algorithms they choose to use.” For legal liability to arise, the designer or the user should be aware of the pricing effects created by it. This can be exemplified by the European *Eturas* case, involving 30 Lithuanian travel agencies that used the same online booking system. The system operator programmed the algorithm so that the agencies could not offer discounts of more than 3%, and notified the agen-

Algorithms are not immune from competition laws.

cies of this restriction via its internal messaging system. The agencies employed the algorithm. The question was whether these events implied an agreement between the travel agencies to change the algorithm and reduce competition. The European Court of Justice made awareness of the change in the algorithm a necessary condition for a finding of a cartel. Disregard to the algorithm’s probable effects may also, under some circumstances, be sufficient to prove awareness. It remains an open question what type of awareness would be required in cases in which an algorithm, which is designed to autonomously determine the decisional parameters, facilitates collusion.

Third, the use of an algorithm is not prohibited if it simply reacts to market conditions set by others, without reaching an agreement. Accordingly, if a designer simply codes his algorithm to react to the prices set by other algorithms, this, by itself, will most likely not be treated as illegal by any jurisdiction. Accordingly, such algorithms fall within the secured zone.

Lastly, to help prove the existence of an agreement, many jurisdictions rely on evidence of intentional, avoidable actions that allow competitors to more easily and effectively coordinate, and that do not increase welfare. Such actions include, for example, exchanges of non-public information on future price increases. Under some circumstances, algorithms might be treated as such actions. To illustrate, red flags might be raised when competitors consciously use similar algorithms that generate relatively similar outcomes even when better algorithms are readily available; when programmers or users of learning algorithms consciously give them similar training data to that used to

train their competitors’ algorithms, despite it not being the best training data readily available; or when users artificially increase the transparency of their algorithms and/or databases to their competitors. In all these cases, competitors implicitly communicate their intentions to act in a certain way, as well as their reliance on one another to follow suit. They do so by using avoidable acts that facilitate coordination. Such conduct can, therefore, trigger deeper investigation.

Nonetheless, given that algorithms perform many beneficial functions in the digital environment, the algorithm’s ability to facilitate coordination must be balanced against its pro-competitive effects, including the potential efficiencies created by the speed of reacting to changes in market conditions. Accordingly, while competitors should not be allowed to mask their cartels through algorithms, regulators should also ensure what we gain by limiting the use of some algorithms is greater than what we lose by limiting the range of allowable design choices. Most courts around the world are already going in this direction, and computer scientists have an important role to play in educating enforcers on such matters. It should be stressed, however, that algorithms will not necessarily be treated as indivisible; a court might prohibit only the coordination-facilitating part of the algorithm.

Algorithms are not immune from competition laws. While the use of algorithms is not prohibited, certain uses of algorithms may be considered illegal. Programmers and users should be aware of the potential legal consequences of such uses. Yet, except in easy cases, regulators are still figuring out when the use of pricing algorithms is prohibited. Indeed, Part of the challenge is that “smart coordination” through algorithms requires “smart regulation”—setting rules that limit the harms of increased coordination, while ensuring the benefits of algorithms are not lost. ■

Michal S. Gal (mgalresearch@gmail.com) is Professor and Director of the Forum for Law and Markets, Faculty of Law, University of Haifa, Israel, and President of the Academic Society for Competition Law Scholars (ASCOLA).

Copyright held by author.



Technology Strategy and Management

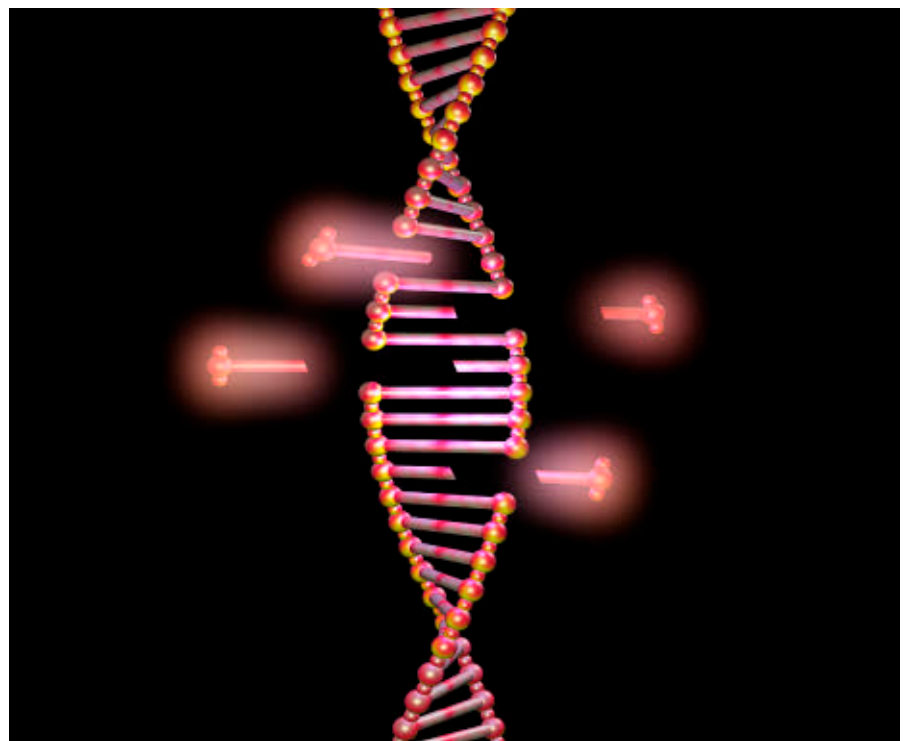
CRISPR: An Emerging Platform for Gene Editing

Considering a potential platform candidate in the evolving realm of gene-editing technologies research.

WHEN THINKING ABOUT which areas of research might form the basis for new industry platforms, in the past we have focused on information technologies such as computers, Internet software, smartphones, cloud services, artificial intelligence and machine learning, and even quantum computing (see “The Business of Quantum Computing,” *Communications*, Oct. 2018). These technologies early on had the potential to generate what we call “multi-sided markets” with powerful “network effects.” Network effects are self-reinforcing feedback loops where, as the number of users or complementary innovations increase, the more widely used and valuable the platform becomes (see “The Evolution of Platform Thinking,” *Communications*, Jan. 2010).

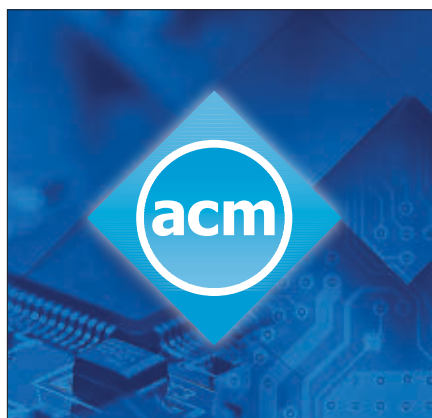
Another early-stage technology suited to platform dynamics is gene editing. Research began several decades ago, leading to various tools and techniques. It is still uncertain which approach will become the dominant foundation for further research and applications development, but there are some platform candidates.

One particularly promising technology is CRISPR, or “Clustered Regularly Interspaced Short Palindromic Repeats.”¹² CRISPR refers to small pieces of DNA that bacteria use to recognize



viruses. What scientists observed years ago is that specialized segments of RNA and associated enzymes in one organism can modify genes (DNA sequences) in another organism. For example, this happens naturally when the immune system in bacteria fight against an invading virus. In 2012, several scientists discovered they could use CRISPR sequences of DNA as well as “guide RNA” to locate target DNA and then deploy

CRISPR-associated enzymes as “molecular scissors” to cut, modify, or replace genetic material. The potential applications include diagnostic tools and treatments for genetic diseases as well as genetic reengineering more broadly.⁸ An August 2016 article in *National Geographic* magazine described CRISPR’s potential: “CRISPR places an entirely new kind of power into human hands. For the first time, scientists can



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



quickly and precisely alter, delete, and rearrange the DNA of nearly any living organism, including us. In the past three years, the technology has transformed biology ... No scientific discovery of the past century holds more promise—or raises more troubling ethical questions. Most provocatively, if CRISPR were used to edit a human embryo's germ line—cells that contain genetic material that can be inherited by the next generation—either to correct a genetic flaw or to enhance a desired trait, the change would then pass to that person's children, and their children, in perpetuity. The full implications of changes that profound are difficult, if not impossible, to foresee.”¹⁰

DNA resembles a programming language and data-storage technology, useful in different applications. Gene editing provides opportunities for companies to pursue product solutions, such as to build standalone diagnostic tools or gene therapies. It also enables some institutions and companies to create products, tools, or components that other firms can build upon. Like today's quantum computers, each use of CRISPR seemed to require specialized domain knowledge (that is, the genome of a particular organism and disease) and then tailoring to the application, such as to use CRISPR to design a diagnostic test or therapeutic product for a specific disease, or to reengineer a plant to fight off insects. But, along with rising numbers of CRISPR researchers, platform-like network effects and multisided market dynamics were also appearing and helping the industry evolve. In particular, more research publications have led to improvements in tools and reusable component libraries, which have attracted more re-

DNA resembles a programming language and data-storage technology, useful in different applications.

searchers and applications, which in turn have inspired more research, tool development, applications, venture capital investments, and so on.

At the center of an emerging CRISPR ecosystem is a non-profit foundation called Addgene, founded in 2004 by MIT students. It funds itself by selling plasmids, small strands of DNA used in laboratories to manipulate genes. Since 2013, it has been collecting and distributing CRISPR technologies to help researchers get started on their experiments.¹⁴ The Addgene tools library consisted of different enzymes and DNA or RNA sequences useful to identify, cut, edit, tag, and visualize particular genes.^a There were also numerous startups, some of which have already gone public. CRISPR Therapeutics (founded in 2013) was trying to develop gene-based medicines to treat cancer and blood-related diseases, and collaborating closely with Vertex and Bayer. Editas Medicine (2013) and Exonic Therapeutics (2017) were tackling diseases such as cancer, sickle cell anemia, muscular dystrophy, and cystic fibrosis.^b Beam Therapeutics (2018) planned to use CRISPR to edit genes and correct mutations.¹ Mammoth Biosciences (2018) was following more of a platform strategy and developing diagnostic tests that could be the basis for new therapies. It was broadly licensing its technology and encouraging other firms to explore therapies based on its testing technology.¹¹ In fact, Mammoth's goal was to create “a *CRISPR-enabled platform* [italics added] capable of detecting any biomarker or disease containing DNA or RNA.” In a recent public statement, the company summarized its strategy to cultivate an applications ecosystem: “Imagine a world where you could test for the flu right from your living room and determine the exact strain you've been infected with, or rapidly screen for the early warning signs of cancer. That's what we're aiming to do at Mammoth—bring affordable testing to everyone. But even beyond healthcare, we're aiming to *build the platform for CRISPR apps* [italics added] and offer the technology across many industries.”³

a See <https://www.addgene.org/crispr/>

b See A. Regalado, “Startup Aims to Treat Muscular Dystrophy with CRISPR,” *MIT Technology Review* (Feb. 27, 2017) and <http://www.editas-medicine.com/pipeline>

Commercialization of CRISPR systems was still years away, and the technology had limitations. It was better at screening, cutting, and rewriting rather than inserting DNA.⁴ And only recently have medical centers and companies applied to start CRISPR-related clinical trials. There were also alternative technologies with different strengths and weaknesses. In particular, TALEN (Transcription Activator-Like Effector Nucleases), another gene-cutting enzyme tool, was more precise than CRISPR and more scalable for some non-laboratory applications, though it was more difficult to use.⁶ In general, CRISPR was in the lead, with several universities and research centers, startup companies, and established firms actively publishing papers, applying for patents, and sharing their tools and depositories of genetic components. Most researchers were also focusing on CRISPR-Cas9, a specific protein that used RNA to edit DNA sequences.

One concern is that the business models of biotech startups and pharmaceutical companies depended on patent monopolies, making the industry ultra-competitive and locking applied research into protected silos. The result was potentially a “zero-sum game” mentality. This contrasted to the more cooperative (but still highly competitive) spirit of “growing the pie” together that we generally see with basic science and which we saw in the early days of the personal computer, Internet applications, and even smartphone platforms such as Google’s Android. Of course, CRISPR scientists openly shared and published their basic research.⁷ And though the U.S. Patent Office already has granted hundreds of patents related to CRISPR, patent holders usually offered free licenses to academic researchers, even those still under litigation.

Ethical and social issues might hinder widespread use of gene editing. The controversies centered on how much genetic engineering should we, as a society, allow? Experts already disagreed about the safety of genetically altered plants and animals that contributed to the human food supply.¹³ Scientists can deploy similar technology to change human embryos and cells, such as to treat genetic diseases or potential disabilities. But should we allow parents to edit their children’s genes, such as to select for blue versus

Ethics and social issues might hinder widespread use of gene editing.

brown eyes, or a higher IQ?⁵

In sum, platform dynamics were influencing areas outside of information technology. It was not so clear, though, how to use the power of the platform wisely and safely, and what types of government monitoring and self-regulation were most appropriate. These issues were likely to become fierce topics of debate as CRISPR and other gene-editing technologies evolved into widely used platforms for medical, food, and other applications. **Q**

References

1. Al Idrus, A. Feng Zhang and David Liu’s base-editing CRISPR startup officially launches with \$87 million. *FierceBiotech.com*, (May 14, 2018).
2. Boettcher, M. and McManus, M.T. Choosing the right tool for the job: RNAi, TALEN, or CRISPR. *Molecular Cell* 58, 4 (May 21, 2015), 575–585; <https://bit.ly/2DOHZB5>.
3. CRISPR company cofounded by Jennifer Doudna comes out of stealth mode. *Genome Web* (Apr. 26, 2018); <https://bit.ly/2QYHkjo>
4. Cyranoski, D. CRISPR alternative doubted. *Nature* (Aug. 11, 2016), 136–137.
5. Hayden, E.C. Should you edit your children’s genes? *Nature* (Feb. 23, 2016).
6. Labiotech Editorial Team. The most important battle in gene editing: CRISPR versus TALEN (Mar. 13, 2018); <https://bit.ly/2TwHlMl>.
7. Lander, E. The heroes of CRISPR. *Cell* (Jan. 14, 2016).
8. McKinsey & Company. Realizing the potential of CRISPR. (Jan. 2017); <https://mck.co/2Bl2MK0>
9. Molteni, M. A new startup wants to use CRISPR to diagnose disease. *Wired* (Apr. 26, 2018).
10. Specter, M. How the DNA revolution is changing us. *National Geographic* (Aug. 2016).
11. Vayas, K. New CRISPR-based platform could soon diagnose diseases from the comfort of your home. *Science* (Apr. 29, 2018).
12. Zimmer, C. Breakthrough DNA editor born of bacteria. *Quanta Magazine* (Feb. 6, 2015).
13. Zimmer, C. What is a genetically modified crop? A European ruling sows confusion. *The New York Times*, (July 27, 2018)
14. Zyontz, S. Running with (CRISPR) scissors: Specialized knowledge and tool adoption. Technological Innovation, Entrepreneurship, and Strategic Management Research Seminar, MIT Sloan School of Management (Oct. 22, 2018).

Michael A. Cusumano (cusumano@mit.edu) is a professor at the MIT Sloan School of Management and founding director of the Tokyo Entrepreneurship and Innovation Center at Tokyo University of Science.

The author thanks Samantha Zyontz as well as David Fritsche, Gigi Hirsch, and Pierre Azoulay for their comments. This column is derived from a forthcoming book by Michael A. Cusumano, Annabelle Gawer, and David B. Yoffie, *The Business of Platforms: Strategy in the Age of Digital Competition, Innovation, and Power*, Harper Business, June 2019.

Copyright held by author.

Calendar of Events

January 13–19

POPL ‘19: The 46th Annual ACM SIGPLAN Symposium on Principles of Programming Languages, Lisbon, Portugal, Sponsored: ACM/SIG, Contact: Fritz Henglein, Email: henglein@diku.dk

January 14–18

AFIRM ‘19: ACM SIGIR/SIGKDD African Workshop on Machine Learning for Data Mining and Search, Cape Town, South Africa, Co-Sponsored: ACM/SIG, Contact: Hussein Suleman, Email: hussain@cs.uct.ac.za

January 29–31

FAT* ‘19: Conference on Fairness, Accountability, and Transparency, Atlanta, GA, Sponsored: ACM/SIG, Contact: danah boyd, Email: danah@datasociety.net

February

February 11–15

WSDM 2019: The 12th ACM International Conference on Web Search and Data Mining, Melbourne, VIC Australia, Co-Sponsored: ACM/SIG, Contact: Alistair M. Moffat, Email: ammoffat@unimelb.edu.au

February 24–26

FPGA ‘19: The 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Seaside, CA, Sponsored: ACM/SIG, Contact: Kia Bazargan, Email: generalchair@isfpga.org

February 25–26

HotMobile ‘19: The 20th International Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, Sponsored: ACM/SIG, Contact: Alec Wolman, Email: alec.wolman@gmail.com

Historical Reflections

Hey Google, What's a Moonshot? How Silicon Valley Mocks Apollo

Fifty years on, NASA's expensive triumph is a widely misunderstood model for spectacular innovation.

THE RADIO IN my kitchen is tuned to a public station. One day it startled me by delivering a lecture, “The unexpected benefit of celebrating failure,” by the implausibly named Astro Teller who, according to his website, enjoys an equally idiosyncratic list of accomplishments: novelist, entrepreneur, scientist, inventor, speaker, business leader, and IT expert. That talk concerned his day job: “Captain of Moonshots” at X (formerly Google X, now a separate subsidiary of its parent company Alphabet).^a It centered on the classic Silicon Valley ideal of being prepared to fail fast and use this as a learning opportunity. Teller therefore advised teams to spend the first part of any project trying to prove it could not succeed. Good advice, but maybe not so new: even 1950s “waterfall” methodologies began with a feasibility stage intended to identify reasons the project might be doomed. Still, many of us have had the experience of putting months, or even years, into zombie projects with no path to success.^b The HBO television series “Silicon Valley” captured that problem, in an episode where a new executive asked for the status of a troubled



Astronaut Alan L. Bean walks from the moon-surface television camera toward the lunar module during the first extravehicular activity of the November 1969 Apollo 12 mission, the second lunar landing in the NASA Apollo program. The mission included placing the first color television camera on the surface of the moon but transmission was lost when Bean accidentally pointed the camera at the sun, disabling the camera.

project.^c Each level of management sugarcoated the predictions it passed upward and avoided asking hard questions of those below it.

To be honest, I was more intrigued

by the “moonshot captain” thing. Teller briefly paid homage to President Kennedy and the huge scope of the real moonshot achieved by the Apollo program of the 1960s. By promoting X as a “moonshot factory” he suggested plans to crank out Apollo-style triumphs regularly, at the intersection of “huge problems, breakthrough technologies, and radical

^a See <https://bit.ly/1TTLG9n>

^b Ed Yourdon wrote an interesting book about the tenacity of doomed projects: E. Yourdon, *Death March: The Complete Software Developer's Guide to Surviving "Mission Impossible" Projects*. Prentice Hall, 1997.

^c This incident occurs in “Server Space” (season 2, episode 5) and, ironically, is set in the Hooli XYZ “moonshot factory”—a rather crude parody of Google X.

solutions.”^d X boasts of uniting “inventors, engineers, and makers” including aerospace engineers, fashion designers, military commanders, and laser experts. Teller’s most dramatic example of an X moonshot that failed admirably was that staple technology of alternate worlds, an airship “with the potential to lower the cost, time, and carbon footprint of shipping.” According to Teller, X achieved the “clever set of breakthroughs” needed to mass produce robust, affordable blimps, but had to give up when it estimated a cost of “\$200 million to design and build the first one” which was “way too expensive.” X relies on “tight feedback loops of making mistakes and learning and new designs.” Spending that much “to get the first data point” was not remotely possible.

At this point, I would like you to imagine the record-scratching noise that TV shows use for dramatic interruptions. That’s what played in my head, accompanied by the thought “this guy doesn’t know what the moonshot was.” Teller’s pragmatic, iterative, product-driven approach to innovation is the exact opposite of what the U.S. did after Kennedy charged it to “commit itself to achieving the goal, before this decade is out, of landing a man on the moon and returning him safely to the earth.” Letting Silicon Valley steal the term “moonshot” for projects with quite different management styles, success criteria, scales, and styles of innovation hurts our collective ability to understand just what NASA achieved 50 years ago and why nothing remotely comparable is actually under way today at Google, or anywhere else.

The Actual Moonshot

As historians of technology Ruth Schwartz Cowan and Matthew Hersch tell the story: “Eight year later, on July 20, 1969, millions of people all over the world watched their televisions

^d X grew out of the lab that “graduated” to become Waymo, now a separate company successfully selling technology for self-driving cars. It was also the group responsible for Google Glass, whose camera/screen eyeglasses went abruptly from next big thing to epic flop in 2014, for the Loon project to deliver Internet access via high-altitude balloons, and for a fleet of experimental delivery drones. The most balanced portrait of its workings was given in <https://bit.ly/2gqMi8s>.

The moonshot was a triumph of management as much as engineering.

in wonder as Neil Armstrong and Edward Aldrin planted the American flag on the moon [after] the largest managed research project of all time.... The Saturn V rocket had a diameter of 33 feet (three moving vans could have been driven, side by side, into the fuel tanks for the first stage) and a height of 363 feet (about the size of a 36-story building). At liftoff, the vehicle weighed 6.1 million pounds, and when the five engines of the first stage were fired ... they generated 7.5 million points of thrust ... [burning] three tons of fuel a second ... ”³

Those statistics tell you something important: the moonshot was about doing something absurdly expensive and difficult once (followed by a few encore performances), not doing something useful cheaply and routinely. Apollo 11 pushed a gigantic rocket through the atmosphere and into space, launching three men toward the moon at more than 24,000 miles an hour. Two of them descended in a flimsy little box wrapped in foil, took pictures, collected rocks, and flew back into lunar orbit. All three returned to Earth, or rather to sea, hurtling back through the atmosphere in a tiny capsule that splashed into the ocean.

Apollo was the capstone to a series of gigantic American technological projects, beginning with the Manhattan Project of the 1940s and continuing into the Cold War with the development of nuclear submarines, Atlas and Minuteman missiles, and hydrogen bombs. It was shaped by a vision for the U.S. space program devised by former Nazi rocket engineer Wernher von Braun, whose heavily accented lectures on space stations and manned missions to the Moon and Mars were popularized during the 1950s with the all-American aid of Walt Disney. Their elaborate agenda came with a huge

price tag, but after the USSR checked off the first few items, by launching a satellite and sending a human into orbit, that suddenly looked like money worth spending. In 1961, Kennedy announced his intentions to Congress and won the first in a series of massive increases for NASA’s budget. Like Kennedy’s other initiatives, the moon program became more popular and politically secure after his death, thanks to Lyndon Johnson’s political arm twisting and huge congressional majorities.

Apollo, like Medicare, was part of a dramatic expansion in federal government spending. A future of interplanetary exploration and colonization was already an article of faith for American science fiction writers in the “golden age” of the 1940s, but they were better at imagining rockets than economic changes. One of Robert Heinlein’s most famous stories, “The Man Who Sold The Moon,” described a moon landing in the 1978 by an eccentric businessman. Described as the “last of the robber barons” he funded his dream by, among other things, promising to cancel postage stamps in a temporary lunar post office, sell the naming rights to craters, and engraving the names of supporters onto a plaque.^e Rather than the big government approach of NASA, had Heinlein imagined a space program run like a Kickstarter project. The government’s sudden and mobilization of overwhelming resources for the moonshot took science fiction writers by surprise.

The moonshot was a triumph of management as much as engineering. Meeting a fixed launch deadline meant working backward to identify the points by which thousands of subsystems had to be ready for testing and integration, and further back to the dates by which they had to be designed and ordered. Steven Johnson’s book *The Secret of Apollo* looked at the systems and techniques developed to turn the efforts of hundreds of subcontractors into a successful moonshot.⁷ As he points out, NASA and its partners succeeded in doing something apparently paradoxical: bureaucratizing innova-

^e This short story was written in 1949 and appeared as the title story in Robert A. Heinlein, *The Man Who Sold the Moon* (Shasta, 1950).

tion. Rather than attempt to do lots of new things at once, an approach that had produced problems for the early U.S. space program, von Braun enforced a careful step-by-step approach. These techniques built on those developed for other Cold War projects, described by historian Thomas Hughes in his book *Rescuing Prometheus*.⁶ For example, the PERT project management tool, now a crucial part of project management software, was developed in the 1950s to support the U.S. Navy's Polaris nuclear submarine project. So was MRP (Materials Requirements Planning), which evolved into the foundation for the enterprise software packages that run almost all modern corporations.

NASA management placed a series of milestones along the road to the moon landing, paralleling some aspects of the incremental approach practiced by modern technology leaders. That is why the moon landing was Apollo 11: previous flights had tested the rockets, the command module, the docking capabilities, and so on. Apollo 8, for example, flew a crew into lunar orbit and back, giving an integrated test of many of the key system components. Before those flights came a series of Gemini missions flown during the mid-1960s to test technologies and develop techniques for challenges such as orbital rendezvous and spacewalks. Systematic ground tests focused on space suits, engines, and other new technologies in isolation before integrating them into larger systems.

Teller stressed the need to prototype rapidly and cheaply and to be ready to kill any "moonshot" in its early stages, but NASA agreed to non-negotiable goals for time (by the end of 1969) and scope (landing and returning a man) without building testable prototypes. When Kennedy announced those objectives in 1961, NASA had achieved just 15 minutes of manned flight in space and its managers had not even decided whether to launch a single integrated spacecraft or send up modules to assemble in Earth orbit. One cannot plan out a schedule that depends on fundamental scientific breakthroughs, since those do not occur on a fixed timescale. A project of that kind is about spending money to mitigate risk, by pushing existing

Project management tools may have improved, but human nature continues to undercut best practices.

technologies to levels of performance, reliability, or miniaturization that would not otherwise be economically practical. Given a choice of two technologically workable ways to do something, NASA would take the better-proven and more expensive way.

Despite this technological conservatism, the focus on fixed deadlines still caused deadly trade-offs. After the Apollo 1 crew died when fire engulfed their capsule in a ground test in January 1967, manned flights were halted for 20 months. A review identified several management failures that had contributed to the accident, including a flawed escape system, poor wiring, and the use of pure oxygen instead of a less dangerous air-like mixture. Afterward, mission controller Gene Kranz confessed to his team that "We were too gung-ho about the schedule and we locked out all of the problems we saw every day in our work. Every element of the program was in trouble ... Not one of us stood up and said, 'Dammit, stop!'"⁷ Half a century later, the same words could be applied to many of Silicon Valley's highest-profile projects, from Tesla's spectacularly hubristic attempt to reinvent the assembly line to Uber's lethally ambitious self-driving car program. Project management tools may have improved, but human nature continues to undercut best practices.

Although Teller, as "Captain of Moonshots," wants to celebrate failure that is not how NASA reacted when it lost Gus Grissom, Ed White, and Robert B. Chaffee. Kranz named his memoir *Failure is Not an Option*, after "the creed we all lived by." Explaining the title, he wrote that in 1970, as his team struggled to save the crew of Apollo 13 after

an explosion in space, "each of us" was haunted by "indelible memories of that awful day three years earlier" when "we had failed our crew."

In the end the Apollo 13 astronauts were fine, but the space program was not. Diminishing political returns led to Apollo's early cancellation, like a briefly buzzy TV show that lost its audience and thus its reason to exist. No human has been further than 400 miles from Earth since 1972. With the Soviets defeated in the moon race there was no need to increase spending still further to tackle the remaining items on von Braun's to-do list: moon bases, space stations, manned Mars missions, and so on. Facing shrinking budgets and diminished political will, NASA instead delivered disconnected fragments of the plan—a space shuttle to assemble large structures in orbit and, many years later, a space station to give the shuttle something to do.

Twenty-first century America is not without enemies, but ISIS and the Taliban never developed space programs. Generations of American politicians have nevertheless tried to prove their visionary leadership by ordering new space missions. None committed anything like the funds needed for a true moonshot effort. George W. Bush dusted off von Braun's old dreams in 2004, terminating the space shuttle and directing NASA to restart manned moon missions by 2020 as a steppingstone to Mars. This set a leisurely 16-year schedule for a moon landing, but a progress review five years later concluded that the program was already so underfunded, overbudget, and behind schedule as to be unsalvageable. In 2012, Newt Gingrich, enjoying a brief surge in support for his presidential candidacy, promised voters he could build a permanent moon base and launch a manned Mars mission by 2020 while still slashing government spending and cutting taxes. Rather than prove Gingrich's gravitas on a trip to the White House, the moon base express took him straight back to the political fringes. More recently, President Trump held a ceremony to sign a policy directive directing NASA to head back to the moon and then onward to Mars. Cynics noted the directive made no mention of new funding and set no timeline.

A Moonshot Is Awesome and Pointless

In 1962, Kennedy campaigned for his plan by saying “We choose to go to the Moon in this decade and do the other things, not because they are easy, but because they are hard.” His moonshot was about spending a \$25 billion fortune to do something absurdly difficult with no direct economic return. It showed America’s technological capabilities, political will, and economic might in its long struggle with the Soviet Union (or, as Kennedy put it, “to organize and measure the best of our energies and skills ...”). Nothing economically viable or practical deserves to be called a moonshot. Scaled up for the size of the U.S. economy, a similarly impressive investment today would be approximately \$600 billion. Apollo was a monumental accomplishment, like the construction of the Pyramids. For Google to emulate that might mean erecting a 10-mile-high earthquake-resistant skyscraper, to literally overshadow Apple and provide an object of public marvel. Does that sound like something Google management would authorize a massive bond issue for? No, it does not—even though the project would surely spur advances in architectural engineering, improvements in materials science, and create a lot of engineering and construction jobs.

In his talk, Teller explained the true goal of his moonshot factory was “making the world a radically better place.” I was a little surprised to hear that cliché used in earnest, several years after “Silicon Valley” skewered it in a montage of fake TechCrunch pitches centered on phrases like “making the world a better place though scalable fault tolerant databases with ACID transactions.”^f I suppose that is why he had to promise “radical” global betterment.

I am having a hard time imagining Kennedy’s famous speech working as a TechCrunch pitch to “make the world a better place by spending billions dollars to harvest 381 kilos of rocks.” Was the Apollo program’s goal to make the world a radically better place? Enough

^f “Silicon Valley”’s relationship to real Silicon Valley culture is discussed in A. Marantz, “How ‘Silicon Valley’ nails Silicon Valley” *The New Yorker* (June 9, 2016) which reports that Teller was not amused when the show parodied his “moonshot factory.”

people doubted that at the time to make Apollo the most obvious symbol of the failure of technology to make the world a better place. “If they can put a man on the moon,” asked critics, “why can’t they do [X].” Common values for X were “cure the common cold,” “end urban poverty” and “fix traffic problems.” The modern version of that might be “If Elon Musk can launch a Tesla at Mars, why can’t his car factory come close to production metrics for quantity and quality that other car-makers hit routinely.” Sometimes the rocket science is the easy part.

The Apollo program did little to directly advance scientific understanding. The decision to meet arbitrary deadlines by rushing special purpose hardware, rather than maximizing the scientific value of the missions or their contribution to longer term goals, caused tensions within NASA at the time.^g Apollo did more to push technology and build engineering capabilities. Apollo created good jobs for scientists, mathematicians, programmers, and engineers, at NASA itself and with contractors. Political considerations spread the work out to facilities around the country, rather than concentrating it in a handful of urban areas. It is easy to decry that spending as corporate welfare or help for the already privileged but, as the recent movie *Hidden Figures* showed, the beneficiaries were not all white men with easy lives. The Apollo program also contributed to the development of software engineering techniques—the guidance code had to work reliably first time. Margaret Ham-

^g The scientific side of the Apollo program is the focus of W.D. Compton, *Where No Man Has Gone Before: A History of the Apollo Lunar Exploration Missions*. U.S. Government Printing Office, Washington, D.C., 1989.

The Apollo program did little to directly advance scientific understanding.

ilton, who led its software team, eventually won the Presidential Medal of Freedom for her work on the project.

There were some significant technology spin-offs from Apollo, though contrary to popular belief, the powdered drink Tang was developed previously, as were Velcro and Teflon. Space technology improved freeze-dried food, microelectronics, scratch-resistant sunglass lenses, and lightweight foil blankets. Most notably, the need for reliable, miniaturized control electronics drove the emergence of a commercial market for microchips, years before they were competitive for ground-based applications. Each Apollo guidance computer used approximately 5,000 simple chips of a standard design, providing enough demand to drop the cost per chip for around \$1,000 down to \$20 or so.² The technique of using redundant control computers, now a standard approach for “fly by wire” commercial airliners, was pioneered by IBM in its work on the Saturn V control systems. One of the most popular database management packages of the early 1970s, IBM’s Information Management System (IMS), had its roots in a system built with North American Rockwell in 1965 to handle the proliferation of Apollo parts.⁵ Despite those accomplishments, the moonshot was not a cost-effective way to boost technology. Giving a quarter of the money on the National Science Foundation would surely have accomplished more, as would directing NASA to spend it on satellites and unmanned space probes. But would politicians ever have made those choices? Spending the money to drop more napalm on Vietnam or stockpile more nuclear weapons would have accomplished less than nothing.

If the moonshot made the world a “radically better place” it was by redirecting history in subtle ways. Like medieval jousting, the space race offered a non-lethal, and proudly phallic, substitute for real military clashes. Despite the flag waving, people across the world thrilled to the spectacle and took collective pride in the accomplishments of our species. The “Earthrise” photograph of a gibbous Earth rising over the lunar horizon, was taken in 1968 by the first humans to venture beyond low Earth orbit. It has been credited with



NASA astronauts Neil A. Armstrong (right), Michael Collins (center), and Edwin E. (“Buzz”) Aldrin Jr. received a ticker-tape parade in New York City after returning from the Apollo 11 mission to the Moon.

inspiring the modern environmental movement. The similarly iconic “Blue Marble” photograph of a tiny, fragile, and complete planet floating in space, was taken by the crew of Apollo 17 in 1972 just as the short era of manned space exploration closed. That image inspired the *Whole Earth Catalog*, and hence the utopian aspirations of today’s tech culture.¹⁰ So in the end, moon rocks were not the only thing the astronauts carried back for us.

New Models of Space Flight

The master-planned monumentality of the moonshot is unfashionable today, even in space development. New space companies like Space X and Blue Origin were founded by Internet commerce pioneers (Elon Musk and Jeff Bezos respectively) to apply Silicon Valley approaches to space development. When the Bush-era Constellation moon program, which NASA had promoted as ‘Apollo on Steroids’ was canceled, Musk repurposed the description as an insult writing that the “new plan is to harness our nation’s unparalleled system of free enterprise (as we have done in all other modes of transport), to cre-

ate far more reliable and affordable rockets.”¹¹ Rather than the moonshot approach of launching gigantic rockets as political performance art, these companies have focused on bringing down the cost of launches to make spaceflight viable for more purposes. Instead of tech firms becoming more like NASA, space exploration has become more like information system development. They have exploited developments in computer hardware and software to build reusable rockets

¹¹ <https://bit.ly/2qCT9QY>

The master-planned monumentality of the moonshot is unfashionable today, even in space development.

able to guide themselves stably back to earth. The same advancements have greatly decreased the minimum size of useful satellites, reducing the mass that needs to be launched into space. (NASA itself anticipated some of this in the “faster, better, cheaper” push of the 1990s that produced the Mars Pathfinder rover). Starting with the smallest useful rockets and a modular architecture, they have been working incrementally to larger and more powerful models. Since the Obama administration, U.S. policy has shifted toward contracting with space companies to purchase the use of privately developed rockets, rather than the traditional government procurement model where companies are given up-front development contracts to supply equipment to government specifications.

Musk and Bezos hope that incrementally developing efficient and economically viable space systems will eventually lead to moon colonies, asteroid mining, and Mars missions. Like Delos D. Harriman, Heinlein’s space fairing businessman, Musk dreams of dying on another world. Yet the new approach has its limits. The \$30 million Google Lunar XPRIZE, for the first private landing of a robot on the moon, recently expired unclaimed 11 years after its announcement. The documentary commissioned to celebrate the competition was, of course, called “Moon Shot.” Private-sector ingenuity proved unable to deliver new Apollo on a shoestring budget, despite the considerable advantages of a longer timescale, 50 years of technological improvement, and an easier task (one way robot transport vs. round trip travel for humans).

Apollo vs. ARPANET

A few months after Neil Armstrong’s short step down to the lunar service, data packets started making longer hops up and down the West Coast. ARPANET’s first four nodes had gone live. Both were government projects, funded as part of the broader Cold War effort but not directly military. Apollo landed a total of 12 men on the moon, the last in 1972. By then ARPA had interconnected around 30 sites. By the time Apollo was officially shut down, after flying a final joint USA-USSR mis-

sion with spare hardware, the ARPANET had received less than one-thousandth of its funding.

The ARPANET was immediately useful and soon became more useful when network email, rather than the remote logins used to justify its construction, provided an unexpected “killer application.” It evolved continually, in response to the needs of its users. The Apollo program, in contrast, had accomplished its objective by the time the Apollo 11 astronauts rode in their tickertape parade down Broadway in New York City.

Since then the divergence of the moonshot and ARPANET approaches has been rather dramatic. As of this writing, only four of the planet’s seven billion human inhabitants have walked on the moon. The youngest of them is now 83 years old, so that number seems more likely to fall than rise. In contrast, approximately half of the world’s population uses the Internet, the direct descendent of ARPANET, and millions more connect to it every day. The incremental, exploratory development of the ARPANET provided the modern tech firms with their model of innovation as well as the Internet infrastructure they rely on.

The End of Innovation?

I am glad Google still spends some money exploring new product opportunities outside its core businesses, unlike many other modern firms, but do not forget that is something big companies used to do routinely without blathering about “moonshots.” Fifty years ago Ford, General Electric, Kodak, Xerox, RCA, AT&T, Kodak, Dow Chemical, 3M, and a host of aerospace firms were investing heavily in such projects. Consulting firm Arthur D. Little specialized in helping companies apply newly developed materials, with stunts like turning a sow’s ear into a silk purse.⁸ Many of those firms also supported labs doing basic research in relevant areas of science, which Google and its peers do not attempt. Today’s leading tech companies are not short of cash, but their focus is on minor improvements and the development of new features and applications within their existing platforms.

Tech companies have not always been so wary of moonshot-scale projects.

Tech companies have not always been so wary of moonshot-scale projects. In my January 2018 column I mentioned IBM’s System/360 development project in the 1960s, which reportedly required a commitment of twice the firm’s annual revenues when the project was launched. For Alphabet today, two years of revenue would be over \$200 billion. Yet its “moonshot captain” had to kill what he claims was a highly promising project, just because an initial investment of \$200 million was unworkable. Poor Astro was three zeros and one comma away from being able to live up to that ridiculous job title. (Talking of absurd job titles, X recently lost its ‘Head of Mad Science’ to a sexual harassment scandal.)

Perhaps that is a good thing. Apollo’s politically driven, money-no-object pushing of technology toward a fixed goal made for great television but did not bring us closer to routine space flight. Like the Concorde supersonic jetliner, sponsored by the French and British governments, it was a technological marvel but an economic dead end. On the other hand, the Silicon Valley model has not delivered nearly as much economic growth as all the talk about innovation and disruption might lead you to believe. Notwithstanding all the amazing things your cellphone does, technological change in the developed world has slowed to a fraction of its former rate. The 1960s were a highwater mark for confidence in the effectiveness of investment in bold technological projects like Apollo, System/360, or ARPANET. In *The Rise and Fall of American Growth*, economist Robert Gordon suggested a century of spectacular growth in living standards, life expectancy, and economic productivity began to stall

around 1970, just as the focus of technological innovation shifted toward computers and networks.⁴ These have not produced anything like the broad and sustained productivity gains created by electricity or assembly lines. Widespread adoption of the Internet gave productivity growth a significant jolt a decade ago, but that has already faded away.

It is inaccurate to blame this slowdown on public reluctance to fund moonshot-sized projects without direct economic returns. More likely, the end of rapid American growth and the end of moonshot projects are two consequences of a political and ideological shift away from long-term public and corporate investment in a range of areas, from infrastructure to education. At the height of the Apollo project, federal spending on research and development was more than twice its level in recent decades. A decades-long push for tax cuts, combined with rising government spending on healthcare and social security, has hollowed out investment in research and infrastructure and left massive deficit.

Companies are likewise more focused than ever on quarterly earnings and shareholder value. Alphabet has the money to fund something close to a real moonshot, if its investors allowed it. In 2015 its total spending on non-core business, not just the “moonshot factory” but potentially vast emerging business areas like fiber-optic Internet service, life sciences, home automation, venture capital, and self-driving cars, accounted for only approximately 5% of its revenues. Even that was viewed by investors as irresponsible, given that they generated less than 1% of its income, and in early 2017 Alphabet reportedly launched an “apparent bloodbath,” killing ambitious plans for delivery drones, modular cellphones, and the rollout of fiber-optic Internet access to more cities.¹ Subsequent reports tied a transition in which “futurism has taken a back seat to more pressing concerns” to the withdrawal of Google co-founder Larry Page from hands-on management.¹

What would modern tech companies do with a windfall big enough to fund an actual moonshot? Thanks to

i <https://bit.ly/2PLDigV>



Association for
Computing Machinery

ACM Conference Proceedings Now Available via Print-on-Demand!

Did you know that you can now order many popular ACM conference proceedings via print-on-demand?

Institutions, libraries and individuals can choose from more than 100 titles on a continually updated list through Amazon, Barnes & Noble, Baker & Taylor, Ingram and NACSCORP: CHI, KDD, Multimedia, SIGIR, SIGCOMM, SIGCSE, SIGMOD/PODS, and many more.

For available titles and ordering info, visit:
librarians.acm.org/pod



If you expect to live to see anything as intoxicatingly implausible as a moon landing was in 1969, you will have to pay for it too.

the recent corporate tax-cut bonanza this is not a hypothetical question. Rather than investing in new projects they purchased their own stock, to return surplus money to shareholders. In the first quarter of 2018, Alphabet announced a \$8 billion buyback. Cisco spent \$25 billion. Apple more recently launched a \$100 billion stock purchase program. Moves of this kind reflect a belief by management that they have no untapped opportunities, including new product development, to make better use of the money. (Thanks in part to those same tax cuts, the U.S. government deficit is expected to balloon to approximately \$1 trillion dollars this year, forestalling any possibility of new public investment).

The Internet approach of scaling up incrementally from a working prototype based on the needs of users has beaten out the centrally planned, all-or-nothing moonshot approach. Investment funds flow to companies with already viable prototypes in hot fields, as evidenced by the vivid but potentially baffling news headline “Bird races to become first scooter unicorn.”^j (Translation: urban scooter rental company Bird was about to pin down a new round of venture capital funding valuing it at more than a billion dollars, making it a “unicorn.”) Silicon Valley is trying to stop us from noticing the difference between the Apollo program and scooter unicorns by draping the heroic rhetoric

^j The headline, originally attached to a story posted by Bloomberg.com on May 29, 2018, has since been replaced with the less-evocative title “Sequoia Said to Value Scooter Company Bird at \$1 Billion.”

of “moonshots” over a far less-inspiring reality. You have probably heard the comment, “we were promised flying cars, but we got 140 characters” (a dismissive reference to Twitter). That is true, but let’s not forget that anyone old enough to have been promised a flying car, back in the 1950s when Ford promoted the idea heavily, was also promised a moon rocket by Disney. They got one too, but only because they were collectively willing to pay for it.

Many people now believe the moonshots were faked. Manned lunar flight remains prohibitively challenging today. Was it really achieved 50 years ago, before microprocessors and Twitter were invented? Yes, but if you hope to live to see anything as intoxicatingly implausible as a moon landing was in 1969, perhaps something to address the challenge posed by climate change, you will have to pay for it too. Otherwise—and I’m looking at you Google—please show some respect for the inspiringly unprofitable lunacy of the real moonshot by finding a different name for whatever Astro Teller and his colleagues are up to. “Research and development” has a nice ring to it. **□**

References

1. Bergen M. and Carr, A. Where in the world is Larry Page? *Bloomberg Businessweek*, (Sept. 17, 2018).
2. Ceruzzi, P.E. *A History of Modern Computing*. MIT Press, Cambridge, MA, 1998, 189.
3. Cowan, R.S. and Hersch, M.H. *A Social History of American Technology (2nd edition)*. Johns Hopkins University Press, Baltimore, MD, 2017, 243.
4. Gordon, R.J. *The Rise and Fall of American Growth: The U.S. Standard of Living Since the Civil War*. Princeton University Press, Princeton, NJ, 2016.
5. Haigh, T. How data got its base: Information storage software in the 1950s and 60s. *IEEE Annals of the History of Computing* 31, 4 (Oct.–Dec. 2009), 6–25.
6. Hughes, T.P. *Rescuing Prometheus*. Pantheon Books, New York, 1998.
7. Johnson, S.B. *The Secret of Apollo: Systems Management in American and European Space Programs*. Johns Hopkins University Press, Baltimore, MD, 2002.
8. Kahn, Jr., E.J. *The Problem Solvers: A History of Arthur D. Little, Inc.* Boston, MA, 1986.
9. Kranz, E. *Failure is Not an Option: Mission Control from Mercury to Apollo 13 and Beyond*. Simon & Schuster, New York, 2009.
10. Turner, F. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. University of Chicago Press, Chicago, 2006.

Thomas Haigh (Thomas.haigh@gmail.com) is an Associate Professor of History at the University of Wisconsin—Milwaukee and Comenius Visiting Professor for the History of Computing at Siegen University. Read more at www.tomandmaria.com/tom.

Thanks to Paul Ceruzzi of the National Air and Space Museum and Matthew Hersch of Harvard University for checking the discussion of the Apollo program for historical accuracy and making valuable suggestions.

Copyright held by author.

Viewpoint

UCF's 30-Year REU Site in Computer Vision

A unique perspective on experiences encouraging students to focus on further education.

THE U.S. GOVERNMENT'S National Science Foundation (NSF) started the Research Experiences for Undergraduates (REU) program in the mid-1980s to attract undergraduates in STEM fields into research careers and to consider going to graduate school. The REU program offers grants to universities to plan and oversee research experiences that enrich undergraduate students' educational experiences. It is believed these experiences encourage the participants to pursue leadership careers in the fields of science, technology, engineering, or mathematics.

The University of Central Florida's (UCF) Computer Vision group was in the selected first group of sites: only three REU sites in NSF's Division of Computer and Information Science and Engineering (CISE) were awarded funding in 1987. The grant duration was one year, so continued funding would require a new application for renewal the following year. A few years later, the grant duration was increased to three years, and remarkably for the past 30 years, UCF has kept continuously being funded, by a total of 14 grants. The NSF funded site pays stipends to 10 undergraduates each year who immerse in research and gain useful insight into the prospect of graduate education as an option for their careers.

Three hundred undergraduate researchers (UGRs) from 38 different states and 75 different institutions have participated in this program, and



The Harris Engineering Center, home of the School of Electrical Engineering and Computer Science at the University of Central Florida, USA.

about 80 have published their projects in high-quality venues. Each year, we solicit applications, and we receive well over 150. After a careful interview, we make offers until our 10 positions are filled. Given our successful streak, we try to shed some perspective over our efforts and experiences; see <http://crcv.ucf.edu/REU/>

Why UCF Has Kept Winning Renewals

It is instructive to contemplate our success and examine our evolution—there are several factors that appear

to have contributed independently to our longevity.

Focus: Computer vision. Our site is focused on exciting and appealing topics in computer vision, which facilitate a condensed short course covering key topics, coordination among faculty and graduate students mentors, and interaction and exchanging ideas among UGRs.

Duration: 12 weeks. While the duration of the program is the most controversial aspect of our site with reviewers (because it makes ineligible those students who have fewer weeks

ACM Journal on Computing and Cultural Heritage



ACM JOCCH publishes papers of significant and lasting value in all areas relating to the use of ICT in support of Cultural Heritage, seeking to combine the best of computing science with real attention to any aspect of the cultural heritage sector.



For further information
or to submit your
manuscript,
visit jocch.acm.org

available), it is the channel that gives us capacity for all our activities. We use the first two weeks to train UGRs in background material and then have a week of sufficient deliberation for topic selection, and the following nine weeks are for the UGR to conduct research. In contrast to our 12 weeks, many sites offer REU summers to students for as low as 8 weeks.

Immerse the UGR within the graduate students' lab. Experiencing work in a research laboratory environment with graduate students, has innumerable benefits; the undergraduates see in so many ways the metamorphosis from their current stage to more experienced researcher. We could not have accomplished our goals each year without a large, successful computer vision Ph.D. program. The Ph.D. program offers a scaffolding for the summer REU.

We shower the REU students with guidance and caring. Like helicopter parents, we keep the undergraduates feeling attended to, valued, and consequently focused. We expend large amounts of effort each year on our REU activities, and this appears to give each participant so much to take away to the next step of their journey in life.

What we wish our activities will deliver. Our activities during the summer and beyond are intended to provide the UGR with the following quality experiences.

A. Logistics (payments, housing, travel to/from the site, transportation for various events). We need to ensure everything happens seamlessly, smoothly, and in a timely manner, causing the least amount of stress and distress to the student.

B. Meeting senior people on the same journey, but quite advanced. UGRs need to meet fully matured researchers who have followed successful career pathways. This must give the UGR the concepts of the possible and achievable levels of success, and the amount of efforts required to achieve them.

C. Meeting those who are just a little more senior. This escalator through different levels of metamorphosis from young undergraduates into young researchers gives the UGRs the sense of what their next short-term steps need to be.

D. Meeting peers. These relationships will assist the students to build

networks of colleagues and acquaintances that will let them gain knowledge about the variety of career short-term steps that are available. It also provides important insights into their social roles within their peer groups of potential researchers. This is an initiation into the process that will accelerate in graduate school.

E. Training for understanding the research of others. This involves having the ability to obtain the necessary background to understand research papers, knowing what is needed to be known about those prior research activities, framing the correct questions to ask accomplished researchers, making connections between the research of others and one's own, accepting guidance from peers, graduate students, mentoring professors, and distinguished researchers.

F. Training for converting mathematical reasoning into implementable code. This is an important computational skill; the situation presents additional challenges when the mathematics is vague and unspecific in its formulation, and needs additional simplifications or boundary conditions to be implementable. Images and videos in computer vision are always helpful in this context, because they help to provide insight.

G. Developing persistence. This skill, possibly the most important for research and novel developments, is expected to be built around many successive failures, but with mentored patience, calm deliberation, and the search for clarity about what is not working.

H. Building presentation confidence—delivery. UGRs should feel

**We could not
have accomplished
our goals each year
without a large,
successful computer
vision Ph.D. program.**

comfortable speaking about topics they know about, even when they sometimes are unsure. They should get practice in making verbal mistakes, and being corrected, and learning to prepare themselves for presentations, anticipating audience questions, and being even more additionally prepared.

I. Building presentation confidence—visual. This is a difficult skill to learn. It is built with lots of practice, and watching the presentations of others, who are peers or more advanced and mature.

J. Building commitment to complete a task. UGRs learn about making commitments for short terms, they learn about daily commitments, weekly commitments, commitments for the 12 weeks, and they understand how to break daunting tasks into smaller chunks of smaller commitments.

K. Exposing UGRs to career possibilities in graduate school and industry. UGRs should feel they have good examples of how the career possibilities in graduate school and industry are realizable, and made real. They should have exposure to knowing where they can seek additional help for acquiring knowledge about these pathways.

The activities. At the end of each activity, we list the letters associated with the experiences that were previously described in this Viewpoint.

- ▶ Immerse the UGR in a research group made up by professor and at least one Ph.D. student (B, C, E, F, G, J).

- ▶ Initial two-week training in vision techniques and machine learning, a combination of lectures, tutorials, and homework (E, F).

- ▶ Each year the cohort is presented with more project choices than there are students, the UGRs select their top few choices, and then we begin the task of iteration until there is a stable student to project pairing; during this period there is a lot of contact between each UGR and the possible project groups; stable pairings are achieved by the end of week three (B, C, E, F, G, J).

- ▶ UGR must do a weekly presentation to a small group consisting of the mentor professor and graduate student and fellow undergraduates mentored by the same professor; the presentation is oral and visual (approximately 15 minutes) (H, I).

The field of computer vision is rapidly evolving and the REU site has kept pace with the changes.

- ▶ Social: Six lunches at Thai/Indian/ Buffet restaurants, picnic, graduating Ph.D. dinner, Distinguished Visitor Lunch/dinner, banquet dinner, certificate dinner (B, C, D, G).

- ▶ Field trips to three companies; during each field trip the company (involved in computer vision work) describes their products and their efforts and each UGR individually presents his/her project work for about 10 minutes (H, I, K).

- ▶ Graduate school workshop. Sessions are titled “Why Grad School?,” “Why I am Going?,” “How I won an NSF Graduate Fellowship?,” “Maximizing your chance of grad school acceptance,” “Doctoral Fellowships,” presented by the Graduate Deans and award winning students (K)

- ▶ Distinguished Visitor Colloquium, and Journey Talk, and group meeting where UGRs describe their summer projects (E, H, I, K).

- ▶ Ph.D. student Thesis Proposal, and Final Defense (C, E, H, I, K).

- ▶ Attend all-graduate students’ meeting where graduate students present their work (C, E, H, I).

- ▶ Meet with the co-director each day during the summer for quick report of how overall life is progressing; this acts as release of pressure (from hardware complaints to group dynamic issues, to scheduling adjustments for weekend trips) (A, G, J).

- ▶ Fall/Spring follow up work with each UGR to assist them to get industry internships, additional REU summers (at other institutions), or apply to permanent industry positions and/or graduate school (K).

At the core of all these activities lies the UGR’s immersion in the graduate environment. The UGR’s research

team is formed depending on the project topic. UGRs are given a desk proximal to the graduate student on their team. The graduate student meets with the UGR at different times of the day, as the UGR makes progress or has questions to discuss. Informal short meetings with the faculty mentor occur every one to three days. All these activities lead up to the weekly presentation by the UGR. Additionally, the UGR has opportunities to meet the faculty mentors and graduate students at social events, and the weekly research meeting for the larger graduate student group.

Our progress during the summer is evaluated by a professional assessment team, which provides us mid-summer feedback allowing us to adjust and adapt our strategies.

Changes Over the Years in Structure and Logistics

Our site has seen changes in many ways over the years. Initially, it offered a year-long REU; the summer was full-time research, while the Fall and Spring components involved part-time research due to full class load. The site was shared with another in-state institution, and half the UGRs were local to one institution while the other half were local to the other, so during the summer the UGRs commuted from home to their institution, and during the Fall and Spring semesters, they were able to take continued computer vision academic courses on site. The year-long duration allowed the training in background computer vision techniques to spill over many weeks and allowed some room for easy accommodation of project topic changes. The first change came with the program becoming a single site. Additional professors from our institution were added to the team as mentors.

The next change was when the site took participants from other states. This necessitated the move to on-campus housing, the transition to focus on the summer months, the need for logistics for managing the processing of the selected out-of-state students, and widespread advertising, recruitment, and interviewing procedures.

The focus on the summer months has led to annual review of the short

ACM Transactions on Accessible Computing



ACM TACCESS is a quarterly journal that publishes refereed articles addressing issues of computing as it impacts the lives of people with disabilities. The journal will be of particular interest to SIGACCESS members and delegates to its affiliated conference (i.e., ASSETS), as well as other international accessibility conferences.



For further information
or to submit your
manuscript,
visit taccess.acm.org

summer background training, inclusion of and proper scheduling of the vast variety of activities. The pre-summer activities of planning the research topics in advance has also taken greater attention.

The recent change of adding new faculty to the Center for Research in Computer Vision (CRCV) has permitted flexibility in how the 10 students are subgrouped for their weekly reporting meetings, how they are mentored each day, and has opened up new research areas within computer vision and machine learning.

Changes in Content

The field of computer vision is rapidly evolving and the REU site has kept pace with the changes. Machine learning approaches started to appear in computer vision, as they were able to contribute to object recognition solutions during the mid-1990s. Approaches such as neural networks, boosting, and support vector machines were actively competing for ascendance during the early 2000s. The advent of Deep Learning in the 2010s has slowly gained acceptance as the dominant paradigm in computer vision, and today, research in computer vision must start with a quick study of deep learning approaches and novices must acquire competence in running practical experiments with large data sets in deep learning implementation environments. Consequently, our own short course now has a strong emphasis on environments like Keras, Tensorflow, and a shift to teaching Python (away from MatLab).

Sample Topics. Looking at the topics pursued over the past 30 years indicates the student projects have evolved with the growth of computer vision. Over the six five-year periods, two topics per period are listed here.

► 1987–1992: Object Recognition using Multiple Sensors; Detection and Representation of Events in Motion Trajectories.

► 1992–1997: Visual Lipreading Using Eigensequences; Screening Mammogram Images for Abnormalities.

► 1997–2002: Person-on-Person Violence Detection in Video Data; Flame Recognition in Video.

► 2002–2007: A Vision-Based System for a UGV to Handle a Road Intersec-

tion; Scale Space Based Grammar for Hand Detection.

► 2007–2012: Optimizing One-Shot Recognition with Micro-Set Learning; Part-based Multiple-Person Tracking with Partial Occlusion Handling.

► 2012–2017: How to Take a Good Selfie?, GIS-Assisted Object Detection and Geo-spatial Localization.

Broadening Participation

UCF's REU has a strong commitment to broaden participation among underrepresented groups. Of the 50 participating UGRs in the past 5 years, 23 are female, and 10 of the 27 males are African-American or Hispanic. This diversity in the cohort contributes to increasing the pipeline of students pursuing graduate careers.

Conclusion

After 30 years (and approximately 300 students), some patterns have emerged. Approximately half the students have proceeded to graduate school. Many of the participants have proceeded to leadership positions in their professions: becoming faculty members, starting their own companies, and rising to managerial positions in Fortune 500 Technology companies. Details about student successes are provided in the booklet at http://crcv.ucf.edu/REU/Booklet_071117.pdf

UCF's CRCV has seen many benefits from its cultivated REU strength. UGRs have provided an opportunity to explore research directions, to develop mentoring skills among faculty (older and newer) and graduate students. CRCV-trained UGRs have populated graduate programs around the nation. Our models of evaluation and attentiveness have allowed for best practices to be tested and employed. The commitment of time, effort, and resources is expected to continue into future decades. ■

Niels Da Vitoria Lobo (niels@cs.ucf.edu) is an Associate Professor at the Department of Computer Science, University of Central Florida, Orlando, FL, USA.

Mubarak A. Shah (shah@cs.ucf.edu) is the founding Director of the Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA.

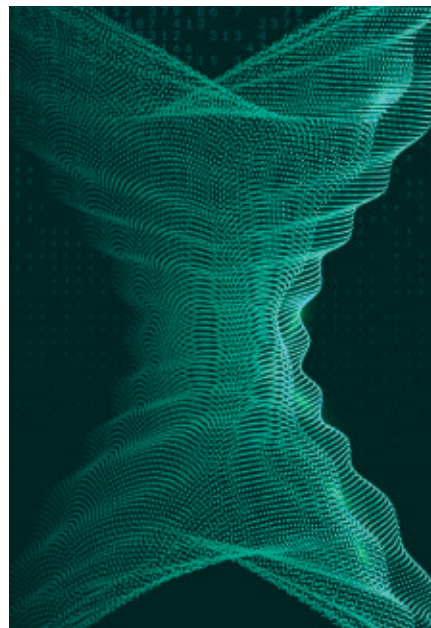
Viewpoint

Modeling in Engineering and Science

Understanding behavior by building models.

FOR MORE THAN 40 years—since 1978—I have been working on computers that interact directly with the physical world. People now call such combinations “cyber-physical systems,” and with automated factories and self-driving cars, they are foremost in our minds. Back then, I was writing assembly code for the Intel 8080, the first in a long line of what are now called x86 architectures. The main job for those 8080s was to open and close valves that controlled air-pressure driven robots in the clinical pathology lab at Yale New Haven Hospital. These robots would move test tubes with blood samples through a semiautomated assembly line of test equipment. The timing of these actions was critical, and the way I would control the timing was to count assembly language instructions and insert no-ops as needed. Even then, this was not completely trivial because the time taken for different instructions varied from four to 11 clock cycles. But the timing of a program execution was well defined, repeatable, and precise.

The models I was working with then were quite simple compared to today’s equivalents. My programs could be viewed as models of a sequence of timed steps punctuated with I/O actions that would open or close a valve. My modeling language was the 8080 assembly language, which itself was a model for the electrical behavior of NMOS circuits in the 8080 chips. What was ultimately happening in the



physical system was electrons sloshing around in silicon and causing mechanical relays to close or open. I did not have to think about these electromechanical processes, however. I just thought about my more abstract model.

Today, getting real-time behavior from a microprocessor is more complicated. Today’s clock frequencies are more than three orders of magnitude higher (more than 2GHz vs. 2MHz), but the timing precision of I/O interactions has not improved and may have actually declined, and repeatability has gone out the window. Today, even if we were to write programs in x86 assembly code, it would be difficult, maybe impossible, to use the same style of design. Instead, we use timer inter-

rupts either directly or through a real-time operating system. To understand the timing behavior, we have to model many details of the hardware and software, including the memory architecture, pipeline design, I/O subsystem, concurrency management, and operating system design.

During these 40-plus years, a subtle but important transformation occurred in the way we approach the design of a real-time system. In 1978, my models specified the timing behavior, and it was incumbent on the physical system to correctly emulate my model. In 2018, the physical system gives me some timing behavior, and it is up to me to build models of that timing behavior. My job as an engineer has switched from designing a behavior to understanding a behavior over which I have little control.

To help understand a behavior over which I have little control, I build models. It is common in the field of real-time systems, for example, to estimate the “worst case execution time” (WCET) of a section of code using a detailed model of the particular hardware that the program will run on. We can then model the behavior of a program using that WCET, obtaining a higher level, more abstract model.

There are two problems with this approach. First, determining the WCET on a modern microprocessor can be extremely difficult. It is no longer sufficient to understand the instruction set, the x86 assembly language. You have to model every detail of the sili-

con implementation of that instruction set. Second, the WCET is not the actual execution time. Most programs will execute in less time than the WCET, but modeling that variability is often impossible. As a consequence, program behavior is not repeatable. Variability in execution times can reverse the order in which actions are taken in the physical world, possibly with disastrous consequences. For an aircraft door, for example, it matters whether you disarm the automatic escape slide and then open the door or the other way around. In this case, as with many real-time systems, ordering is more important than speed.

The essential issue is that I have used models for real-time behavior in two very different ways. In 1978, my model was a specification, and it was incumbent on the physical system to behave like the model. In 2018, my model is a description of the behavior of a physical system, and it is incumbent on my model to match that system. These two uses of models are mirror images of one another.

To a first approximation, the first style of modeling is more common in engineering and the second is more common in science. A scientist is given a physical system and must come up with a model that matches that system. The value of the model lies in how well its behavior matches that of the physical system. For an engineer, however, the value of a physical system lies in how well it matches the behavior of the model. If the 8080 microprocessor overheats and fails to correctly execute the instructions I have specified, then the problem lies with the physical system, not with the model. On the other hand, if my program executes more quickly than expected on a modern microprocessor and the order of events gets reversed, the problem lies with my model, not with the physical system.

Some of humanity's most successful engineering triumphs are based on the engineering style of modeling. Consider VLSI chip design. Most chips are designed by specifying a synchronous digital logic model consisting of gates and latches. A physical piece of silicon that fails to match this logic model is just beach sand. One level up in abstraction, a synchronous digital logic model

Science and engineering are both all about models.

that fails to match the Verilog or VHDL program specifying it is similarly junk. And a Verilog or VHDL model that fails to correctly realize the x86 instruction set is also junk, if an x86 is the intended design. We can keep going up in levels of abstraction, but the essential point is that at each level, the lower level must match the upper one.

In science, models are used the other way around. If Boyle's Law were not to accurately describe the pressure of a gas as it gets compressed, we would hold the model responsible. In science, the upper level of abstraction must match the lower one, the reverse of engineering.

The consequences are profound. A scientist asks, "Can I build a model for this thing?" whereas an engineer asks, "Can I build a thing for this model?" In addition, a scientist tries to shrink the number of relevant models, those needed to explain a physical phenomenon. In contrast, an engineer strives to grow the number of relevant models, those for which we can construct a faithful physical realization.

These two styles of modeling are complementary, and most scientists and engineers use both styles. But in my experience, they usually do not know which style they are using. They do not know whether they are doing science or engineering.

Nobel prizes are given for science, not for engineering. But in 2017, Rainer Weiss, Barry Barish, and Kip Thorne won the Nobel Prize in physics "for decisive contributions to the LIGO detector and the observation of gravitational waves." The LIGO detector is an astonishing piece of engineering, an instrument that can measure tiny changes in distance between objects four kilometers apart, even changes much smaller than the diameter of a

proton. They engineered a thing for a model, and that thing has enabled science. Their decisive engineering triumph, the LIGO detector, enabled experimental confirmation of a scientific model of a physical phenomenon in nature, gravitational waves. Gravitational waves are a 100-year-old model due to Einstein, but LIGO has also enabled new science because it has detected more black hole collisions than astronomers expected. This will require revising our models of the universe. Here, science precedes engineering and engineering precedes science.

Returning to real-time systems, the problem today is that we are doing too much science and not enough engineering. As a community, people who work in real-time systems resign themselves to the microprocessors given to us by Intel and Arm. Those are definitely engineering triumphs, but the models that they realize have little to do with timing. Instead of accepting those microprocessors as if they were artifacts found in nature, we could design microprocessors that give us precise and controllable timing, processors that we call PRET machines.¹ Then we could specify real-time behaviors, and the hardware will need to match our specification. We have shown that such microprocessors can be designed, and that at a modest cost in hardware overhead, there is no need to sacrifice performance.²

Science and engineering are both all about models. But their uses of models are different and complementary. Any model is built for a purpose, and if we do not understand the purpose, the model is not likely to be very useful. To read more about the relationship between engineering and scientific models, see my recent book.³ ■

References

1. Edwards, S.A. and Lee, E.A. The case for the precision timed (PRET) machine. In *Proceedings of the Design Automation Conference (DAC)*, San Diego, CA, 2007.
2. Lee, E.A., Reineke, J., and Zimmer, M. Abstract PRET machines. In *Proceedings of IEEE Real-Time Systems Symposium (RTSS)*, Paris, France, 2017.
3. Lee, E.A. *Plato and the Nerd—The Creative Partnership of Humans and Technology*. MIT Press, 2017.

Edward A. Lee (eal@berkeley.edu) is Professor in the Graduate School and the Robert S. Pepper Distinguished Professor Emeritus and in EECS at UC Berkeley.

Inviting Young Scientists



HEIDELBERG
LAUREATE
FORUM



Association for
Computing Machinery

Meet Great Minds in Computer Science and Mathematics

As one of the founding organizations of the Heidelberg Laureate Forum <http://www.heidelberg-laureate-forum.org/>, ACM invites young computer science and mathematics researchers to meet some of the preeminent scientists in their field. These may be the very pioneering researchers who sparked your passion for research in computer science and/or mathematics.

These laureates include recipients of the ACM A.M. Turing Award, the Abel Prize, the Fields Medal, and the Nevanlinna Prize.

The 7th Heidelberg Laureate Forum will take place **September 22–27, 2019** in Heidelberg, Germany.

This week-long event features presentations, workshops, panel discussions, and social events focusing on scientific inspiration and exchange among laureates and young scientists.

Who can participate?

New and recent Ph.Ds, doctoral candidates, other graduate students pursuing research, and undergraduate students with solid research experience and a commitment to computing research

How to apply:

Online: <https://application.heidelberg-laureate-forum.org/>
Materials to complete applications are listed on the site.

What is the schedule?

The application deadline is **February 15, 2019**.

We reserve the right to close the application website early depending on the volume

Successful applicants will be notified by **mid April 2019**.

More information available on Heidelberg social media



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

Save time by sharing and reusing build and test output.

BY ALPHA LAM

Using Remote Cache Service for Bazel

SOFTWARE PROJECTS TODAY are getting more and more complex. Code accumulates over the years as organization growth increases the volume of daily commits. Projects that used to take minutes to complete a full build now start with fetching from the repository and may require an hour or more to build.

A developer who maintains the infrastructure constantly has to add more machines to support the ever-increasing workload for builds and tests, at the same time facing pressure from users who are unhappy with the long submit time. Running more parallel jobs helps, but this is limited by the number of cores on the machine and the parallelizability of the build. Incremental builds certainly help, but might not apply if clean builds are needed for production releases. Having many build machines also increases maintenance.

Bazel (<https://bazel.build/>) provides the power to run build tasks remotely and massively parallel. Not every organization, however, can afford to have an in-house

remote execution farm. For most projects a remote cache is a great way to boost performance for build and test by sharing build outputs and test outputs among build workers and workstations. This article details the remote cache feature in Bazel (<https://docs.bazel.build/versions/master/remote-caching.html>) and examines options for building your own remote cache service. In practice, this can reduce the build time by almost an order of magnitude.

How Does It Work?

Users run Bazel (<https://docs.bazel.build/versions/master/user-manual.html>) by specifying targets to build or test. Bazel determines the dependency graph of actions to fulfill the targets after analyzing the build rules. This process is incremental, as Bazel will skip the already completed actions from the last invocation in the workspace directory. After that, it goes into the execution phase and executes actions according to the dependency graph. This is when the remote cache and execution systems come into play.

An action in Bazel consists of a command, arguments to the command, and the environment variables, as well as lists of input files and output files. It also contains the description of the platform for remote execution, which is outside the scope of this article. The information about an action can be encoded into a protocol buffer (<https://developers.google.com/protocol-buffers/>) that works as a fingerprint of the action. It contains the command, arguments, and environment variables combined as a digest and a Merkle tree digest from the input files. The Merkle tree is generated as follows: files are the leaf nodes and are digested using their corresponding content; directories are the tree nodes and are digested using digests from their subdirectories and children files. Bazel uses SHA-256 as the default hash function to compute the digests.

Before executing an action, Bazel constructs the protocol buffer using the process described here. The buffer is then digested to look up the remote ac-



tion cache, known as the action digest or action key. If there is a hit, the result contains a list of output files or output directories and their corresponding digests. Bazel downloads the contents of a file using the file digest from the CAS (content-addressable store). Looking up the digest of an output directory from the CAS results in the contents of the entire directory tree, including subdirectories, files, and their corresponding digests. Once all the output file directories are downloaded, the action is completed without the need to execute locally.

The cost of completing this cached action comes from the computation of digests of input files and the network round trips for the lookup and transfer of the output files. This cost is usually substantially less than executing the action locally.

In case of a miss, the action is executed locally, and each of the output files is uploaded to the CAS and indexed by the content digests. Standard output and error are uploaded similarly to files. The action cache is then updated to record the list of output files, directories, and their digests.

Because Bazel treats build actions and test actions equally, this mechanism also applies to running tests. In this case, the inputs to a test action will be the test executable, runtime dependencies, and data files.

The scheme does not rely on incremental state, as an action is indexed by a digest computed from its immediate inputs. This means once the cache is populated, running a build or test on a different machine will reuse all the already-computed outputs as long as the source files are identical. A developer can iterate on the source code; then build outputs from every iteration will be cached and can be reused.

Another key design element is that cache objects in the action cache and CAS can be independently evicted, as Bazel will fall back to local execution in the case of a cache miss or error reading from either one. The number of cache objects will grow over time since Bazel does not actively delete. It is the responsibility of the remote cache service to perform eviction.

Remote Cache Usage

Two storage buckets are involved in the remote cache system: a CAS that stores files and directories and an action cache that stores the list of output files and directories. Bazel uses the HTTP/1.1 protocol (<https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>) to access these two storage buckets. The storage service needs to support two HTTP methods for each of the storage buckets: the PUT method, which uploads the content for a binary blob, and the GET method, which downloads the content of a binary blob.

The most straightforward way to enable this feature with Bazel is to add the flags in the following example to the `~/.bazelrc` file:

```
build --remote_http_cache=http://build/cache
build --experimental_remote_spawn_cache
```

This enables remote cache with local sandboxed execution.

The first flag, `--remote_http_cache`, specifies the URL of the remote cache service. In this example, Bazel uses the path `/ac/` (that is, `http://build/cache/ac`) to access the action cache bucket and the path `/cas/` (`http://build/cache/cas`) to access the storage bucket for the CAS.

The second flag, `--experimental_remote_spawn_cache`, enables the use of remote cache for eligible actions with sandboxed execution in case of a cache miss. When downloading from or uploading to a bucket, the last segment of the path (aka a slug) is a digest.

The next example shows two possible URLs that Bazel might use to access the cache service:

```
http://build/cache/cas/cf80c-d8aed482d5d1527d7dc72fceff84e6326592848447d2dc0b0e87dfc9a90
http://build/cache/ac/cf80c-d8aed482d5d1527d7dc72fceff84e6326592848447d2dc0b0e87dfc9a90
```

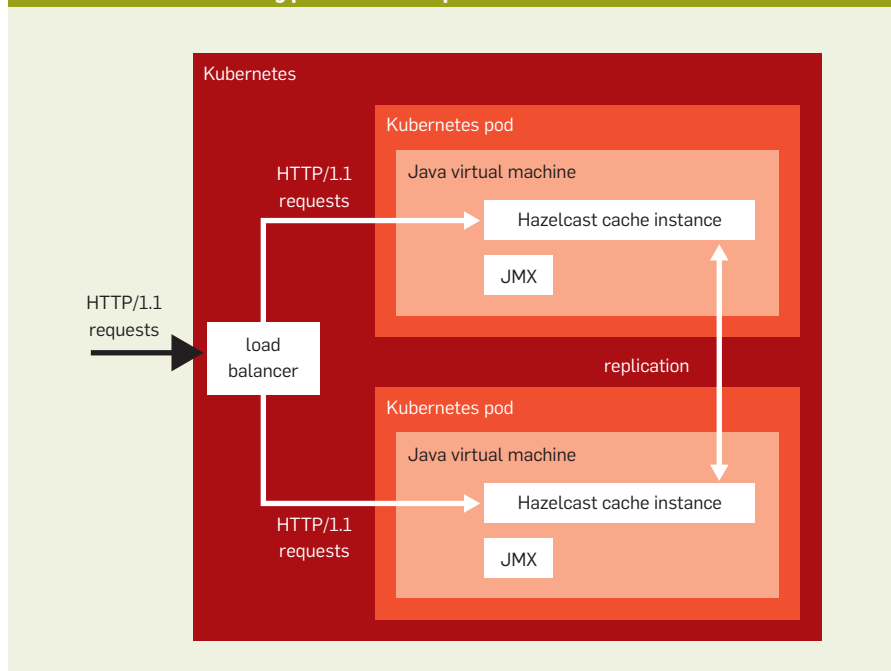
To more finely control the kinds of actions that will use the remote cache without local sandboxed execution, you can use the flags shown in the following example. Individual actions can be opted in to use the remote cache service by using the flag

```
--strategy=<action_name>=remote.
build --remote_http_cache=http://build/cache
build --spawn_strategy=remote
build --genrule_strategy=remote
build --strategy=Javac=remote
```

The default behavior of Bazel is to read from and write to the remote cache, which allows all users of the remote cache service to share build and test outputs. This feature has been used in practice for a Bazel build on machines with identical configurations in order to guarantee identical and reusable build outputs.

Bazel also has experimental support for using a gRPC (gRPC Remote Procedure Call) service to access the remote cache service. This feature might pro-

Remote cache service using pen source components.



vide better performance but may not have a stable API. The Bazel Buildfarm project (<https://github.com/bazelbuild/bazel-buildfarm>) implements this API.


Implementing a Cache Service

An HTTP service that supports PUT and GET methods with URLs in forms similar to the second example in the previous section can be used by Bazel as the remote cache service. A few successful implementations have been reported.


Google Cloud Storage (<https://cloud.google.com/storage/>) is the easiest to set up if you are already a user. It is fully managed, and you are billed depending on storage needs and network traffic. This option provides good network latency and bandwidth if your development environment and build infrastructure are already hosted in Google Cloud. It might not be a good option if you have network restrictions or the build infrastructure is not located in the same region. Similarly, Amazon S3 (Simple Storage Service; <https://aws.amazon.com/s3/>) can be used.

For onsite installation, nginx (<https://nginx.org/en/>) with the WebDAV (Web Distributed Authoring and Versioning) module (http://nginx.org/en/docs/http/nginx_http_dav_module.html) will be the simplest to set up but lacks data replication and other reliability properties if installed on a single machine.

The accompanying figure shows an example system architecture implementation of a distributed Hazelcast (<https://hazelcast.com/>) cache service (<https://hazelcast.com/use-cases/caching/cache-as-a-service/>) running in Kubernetes (<https://kubernetes.io/>). Hazelcast is a distributed in-memory cache running in a JVM (Java Virtual Machine). It is used as a CaaS (cache-as-a-service) with support for the HTTP/1.1 interface. In the figure, two instances of Hazelcast nodes are deployed using Kubernetes and configured with asynchronous data replication within the cluster. A Kubernetes Service (<https://kubernetes.io/docs/concepts/services-networking/service/>) is configured to expose a port for the HTTP service, which is load-balanced within the Hazelcast cluster. Access metrics and data on the health of the JVM are collected via JMX (Java Management Extensions). This example architecture is more reliable than a single-machine installation and easily



Bazel is an actively developed open source build and test system that aims to increase productivity in software development.



scalable in terms of QPS (queries per second) and storage capacity.

You can also implement your own HTTP cache service to suit your needs. Implementing the gRPC interface for a remote cache server is another possible option, but the APIs are still under development.

In all implementations of the cache service it is important to consider cache eviction. The action cache and CAS will grow indefinitely since Bazel does not perform any deletions. Controlling the storage footprint is always a good idea. The example Hazelcast implementation in the figure can be configured to use a least recently used eviction policy with a cap on the number of cache objects together with an expiration policy. Users have also reported success with random eviction and by emptying the cache daily. In any case, recording metrics about cache size and cache hit ratio will be useful for fine-tuning.

Best Practices

Following the best practices outlined here will avoid incorrect results and maximize the cache hit rate. The first best practice is to write your build rules without any side effects. Bazel tries very hard to ensure hermeticity by requiring the user to explicitly declare input files to any build rule. When the build rules are translated to actions, input files are known and must present during execution. Actions are executed in a sandbox by default, and then Bazel checks that all the declared output files are created. You can, however, still write a build rule with side effects using `genrule` or a custom action written in the Skylark language (<https://docs.bazel.build/versions/master/skylark/language.html>), used for extensions. An example is writing to the temporary directory and using the temporary files in a subsequent action. Undeclared side effects will not be cached and might cause flaky build failures regardless of whether remote cache is used.

Some built-in rules such as `cc_library` and `cc_binary` have implicit dependencies on the toolchain installed on the system and on system libraries. Because they are not explicitly declared as inputs to an action, they are not included in the computation of the action digest for looking up the action cache. This can lead to the reuse of object files compiled with a

different compiler or from a different CPU architecture. The resulting build outputs might be incorrect.

Docker containers (<https://www.docker.com/what-container>) can be used to ensure that all build workers have exactly the same system files, including toolchain and system libraries. Alternatively, you can check in a custom toolchain to your code repository and teach Bazel to use it, ensuring all users have the same files. The latter approach comes with a penalty, however. A custom toolchain usually contains thousands of files such as the compiler, linker, libraries, and many header files. All of them will be declared as inputs to every C and C++ action. Digesting thousands of files for every compilation action will be computationally expensive. Even though Bazel caches file digests, it is not yet smart enough to cache the Merkle tree digest of a set of files. The consequence is that Bazel will combine thousands of digests for each compilation action, which adds considerable latency.

Nonreproducible build actions should be tagged accordingly to avoid being cached. This is useful, for example, to put a timestamp on a binary, an action that should not be cached. The following `genrule` example shows how the `tags` attribute is used to control caching behavior. It can also be used to control sandboxing and to disable remote execution.

```
genrule(  
  name = "timestamp",  
  srcs = [],  
  outs = ["date.txt"],  
  cmd = "date > date.txt",  
  tags = ["no-cache"],  
)
```

Sometimes a single user can write erroneous data to the remote cache and cause build errors for everyone. You can limit Bazel to read-only access to the remote cache by using the flag shown in the next example. The remote cache should be written only by managed machines such as the build workers from a continuous integration system.

```
build --remote_upload_local_results=false
```

A common cause of cache miss is an environment variable such as `TMPDIR`. Bazel provides a feature to standardize environment variables such as `PATH` for running actions. The next example shows how `.bazelrc` enables this feature:

```
build --experimental_strict_action_env
```

Future Improvements

With just a few changes, the remote cache feature in Bazel will become even more adept at boosting performance and reducing the time necessary to complete a build.

Optimizing the remote cache. When there is a cache hit after looking up the remote cache using the digest computed for an action, Bazel always downloads all the output files. This is true for all the intermediate outputs in a fully cached build. For a build that has many intermediate actions this results in a considerable amount of time and bandwidth spent on downloading.

A future improvement would be to skip downloading unnecessary action outputs. The result of successfully looking up the action cache would contain the list of output files and their corresponding content digests. This list of content digests can be used to compute the digests to look up the dependent actions. Files would be downloaded only if they are the final build artifacts or are needed to execute an action locally. This change should help reduce bandwidth and improve performance for clients with weak network connections.

Even with this optimization, the scheme still requires many network round trips to look up the action cache for every action. For a large build graph, network latency will become the major factor of the critical path.

Buck has developed a technique to overcome this issue (<https://bit.ly/2OiFDzZ>). Instead of using the content digests of input files to compute a digest for each action, it uses the action digests from the corresponding dependency actions. If a dependency action outputs multiple files, each can be uniquely identified by combining the action digest from its generating action and the path of the output file. This mechanism needs only the content digests of the source files and the action dependency graph to compute every action digest in the entire graph. The remote cache service can be queried in bulk, saving the network round trips.

The disadvantage of this scheme is that a change in a single source file—even a trivial one such as changing the code comments—will invalidate the cache for all dependents. A potential


solution is to index the action cache with the action digests computed using both methods.

Another shortcoming in the implementation of remote cache in Bazel is the repeated computation of the Merkle tree digest of the input files. The content digests of all the source files and intermediate action outputs are already cached in memory, but the Merkle tree digest for a set of input files is not. This cost becomes evident when each action consumes a large number of input files, which is common for compilation using a custom toolchain for Java or C and C++. Such build actions have large portions of the input files coming from the toolchain and will benefit if parts of the Merkle tree are cached and reused.

Local disk cache. There is ongoing development work to use the file system to store objects for the action cache and the CAS. While Bazel already uses a disk cache for incremental builds, this additional cache stores all build outputs ever produced and allows sharing between workspaces.

Conclusion

Bazel is an actively developed open source build and test system that aims to increase productivity in software development. It has a growing number of optimizations to improve the performance of daily development tasks.

Remote cache service is a new development that significantly saves time in running builds and tests. It is particularly useful for a large code base and any size of development team. 

Related articles on queue.acm.org

Borg, Omega, and Kubernetes

Brendan Burns, Brian Grant, David Oppenheimer, Eric Brewer, and John Wilkes
<https://queue.acm.org/detail.cfm?id=2898444>

Nonblocking Algorithms and Scalable Multicore Programming

Samy Al Bahra
<https://queue.acm.org/detail.cfm?id=2492433>

Unlocking Concurrency

Ali-Reza Adl-Tabatabai, Christos Kozyrakis, and Bratin Saha
<https://queue.acm.org/detail.cfm?id=1189288>

Alpha Lam is a software engineer. His areas of interest are video technologies and build systems. Most recently he worked at Two Sigma Investments. He currently works at Google.

Copyright held by author/owner.

Securely running processes that require the entire syscall interface.

BY JESSIE FRAZELLE

Research for Practice: Security for the Modern Age



WHEN DEPLOYING APPLICATIONS in the cloud, practitioners seek to use the most operable set of tools for the job; determining the “right” tool is, of course, nontrivial. Back in 2013, Docker won the hearts of developers by being easy to use, but Linux containers themselves

have been around since 2007, when control groups (cgroups) were added to the kernel. Today, containers have spawned a large ecosystem of new tools and practices that many professionals are using on a daily basis. The foundational technologies making up containers are not new, however. Unlike Solaris Zones or FreeBSD Jails, Linux containers are not discrete kernel components built with isolation in mind. Rather, Linux containers are a combination of technologies in the kernel: namespaces, cgroups, AppArmor, and SELinux, to name a few.

Containers are not the abstraction an application developer typically encounters today. The trend is toward functions and “serverless,” allowing the user to run a single function in the cloud. Because of the way applications and functions are run in the cloud, there will likely be a new generation of isolation techniques built around running a single process securely in an easy and minimal way.

While evidence has shown that “a container with a well-crafted secure computing mode (seccomp) profile (which blocks unexpected system calls) provides roughly equivalent security to

a hypervisor” (<https://bit.ly/2K5tzNi>) methods are still needed for securely running those processes that require the entire syscall interface. Solving this problem has led to some interesting research.

Let’s take a look at some of the research being done in these areas.

Virtual Machines Versus Containers

Filipe Manco et al.

My VM is Lighter (and Safer)

Than Your Container; <https://dl.acm.org/citation.cfm?id=3132763>

Containers became popular as an alternative to virtual machines (VMs) because they are better in the areas of fast boot, small memory overhead, and allowing high density on a single machine. This paper explores creating VMs that meet those same requirements, along with the container features of pause and unpause.

Taking into consideration that the required functionality for most containers is a single application, the authors explored unikernels (minimal VMs where the operating system is linked directly to the application) and TinyX (a tool to create minimal Linux distributions for an application). The smaller the VM image is, the smaller the memory footprint will be and the faster the image will boot.

For containers, just like a typical process running on a host, the number of processes or containers you start does not affect the time to start them, given the usual caveats about resources not being infinite, even in the cloud. This is not true for VMs. The overhead to start a VM increases as more of them are run. The authors found, in the case of Xen, this is a result of both device-creation time and interactions with the XenStore. The authors implemented their own LightVM to solve a lot of the algorithmic and design problems they found with Xen.

The result of their efforts is minimal VMs that can be booted in as little as 2.3ms. A standard Linux process starts in about 1ms, and a docker container starts in about 40ms, depending on the size of the image. The boot time remains constant the more VMs are launched, which is in stark contrast to typical VMs. Unikernels, however, are not as easy to create as containers and require individual de-

velopment time to be made functional for each application.

Isolation of Applications In a Minimal Way

Dan Williams and Ricardo Koller

Unikernel Monitors: Extending

Minimalism Outside of the Box;

<https://dl.acm.org/citation.cfm?id=3027053>

Minimal software has the benefits of reducing attack surface and making software more understandable with less overhead. Unikernels are frequently discussed in the context of minimal and secure ways to run programs in the cloud. In the traditional approach a unikernel is a VM and, as such, is run in a VM monitor, which is a program that watches and controls the lifecycle of VMs, such as VMWare, QEMU, or VirtualBox. Unikernel monitors are bundled into the unikernel. This creates a minimal way to boot unikernels without the added complexity of using a stand-alone VM monitor.

Most VM managers/monitors are heavyweight, with features for devices that are not used in modern or cloud environments. Take QEMU, for example: it comes with the emulation for devices such as keyboards and floppy drives. If there is an exploit in the floppy-drive emulator, it is game over for the whole system, even though a floppy drive obviously has no usefulness in the cloud.

If a monitor is purpose-built for booting unikernels, its computing base is much more minimal than the VM monitors in use today (about five percent of the size). The authors of this paper created a monitor that has only two jobs: creating the isolation to run the unikernel and performing actions when the unikernel exits. The monitor is also baked into the executable for the unikernel, creating a simplistic and minimal approach for distributing and executing unikernels.

The boot time for their prototype was 10ms, which is eight times faster than a traditional monitor. This paper has a positive vision of the future, running applications in a minimal and secure way in the cloud. IBM recently released a container runtime called Nabla (<https://nabla-containers.github.io/>) around the topics and implementations of this paper.

Virtualize at the Runtime Layer

James Larisch, James Mickens, and Eddie Kohler

Alto: Lightweight VMs using

Virtualization-Aware Managed Runtimes;

<https://mickens.seas.harvard.edu/files/mickens/files/alto.pdf>

Traditional virtual machines, like Xen, virtualize at the hardware layer. Docker, on the other hand, virtualizes at the POSIX layer. This paper suggests a new approach to virtualize at the runtime layer.

One of the more difficult questions in this space is how to handle state. In traditional environments, state for the file system and network is handled in the kernel. The authors suggest moving as much kernel state as possible into the virtual machine through a user-space networking stack and FUSE filesystem. They also suggest explicitly depicting each state object as an addressable server (each with its own IP address), allowing operators to easily migrate and update applications since there is clean separation of a program’s code, stack, and heap.

Through innovations in memory allocation, garbage collection, and managing state, Alto seems to be the closest solution to securing processes minimally while giving a new set of controls to operators. As someone who has spent quite a bit of time thinking about the problems faced by creating a minimal, virtualized container runtime, I truly enjoyed the problem statements and solutions this paper laid out.

Deterring Attackers In Your Application

Zhengkao Hu, Yu Hu, and Brendan Dolan-Gavitt

Chaff Bugs: Deterring Attackers

by Making Software Buggier;

<https://arxiv.org/abs/1808.00659>

Defense of software and systems usually consists of correcting bugs that can be exploitable and building software with more than one layer of security, meaning that even if attackers penetrate one layer of the system, they must also penetrate another layer to discover anything of value. Static analysis of code helps automate some of this today but is still not a guarantee of software security.

People tend not to take “security through obscurity” seriously, but there is some value to the technique. Address space layout randomization is an example of this approach, however, it comes at a performance cost.


This paper describes a new approach to slowing down attackers trying to exploit your system. Because this approach automatically injects nonexploitable bugs into software, an attacker who finds said bugs will waste precious time triaging the bug in order to use it maliciously and will fail. In some cases the bugs injected will cause the program to crash, but in modern distributed systems this is unlikely to be an issue because many programs use process pools, and high-availability systems, like those that use containers, typically have a policy for automatically restarting the program on crash.

The bugs injected come in two forms: those that overwrite unused data, and those that overwrite sensitive data with nonexploitable values. The former is fairly straightforward: inject unused variables into the code and ensure the dummy variable is placed directly adjacent to the variable that will be overflowed. In the latter case of overwriting sensitive data, the attacker's input value is overconstrained, meaning it has a defined set of constraints that are by design forced eventually to be zero, through bitmasks and controlling the pathway that the data is passed through.


The key insight in this paper is that instead of trying to decrease the number of bugs in your program, you could increase them but make them nonexploitable, thereby deterring attackers by wasting their time. There is still a performance overhead brought on by the overconstrained checking of inputs, and it is an open question whether the attackers could find patterns in the injected bugs to rule them out automatically. This was, however, enough to fool tools such as gdb, which considered the bugs "exploitable" and "probably exploitable." Could future versions of this approach be designed differently to be more useful to open-source projects? Having the source code would surely give attackers an advantage in discovering which bugs were real and which were injected.

The Future of Securing Applications in a Usable Way

The container ecosystem is very fast paced. Numerous companies are building products on top of existing technologies, while enterprises are using these technologies and products to run their infrastructures. The focus of the three



Containers became popular as an alternative to virtual machines (VMs) because they are better in the areas of fast boot, small memory overhead, and allowing high density on a single machine.




papers described here is on advancements to the underlying technologies themselves and strategic ways to secure software in the modern age.

The first paper rethinks VMs in modern environments purely as mechanisms for running applications. This allows for the creation of minimal VMs that can behave just like containers in terms of memory overhead, density, and boot time. The second paper takes this a bit further by packaging the monitor in the unikernel. This is an extremely usable way to execute unikernels since the operator does not have to install a VM manager. It also allows for a more minimal monitor, limiting the attack surface. IBM's recently launched Nabra container runtime is an example of those approaches. Both papers leverage unikernels and have an open question as to whether unikernels can eventually be as easy to build as containers are today. This will be a hurdle for those implementations to overcome.

The third paper suggests a whole new approach, which also gives operators a new set of controls for managing state. Through isolation at the address space and tying each piece of state to an IP address, operators gain clear controls over a program's code, stack, and heap. Alto not only innovated as far as isolation techniques but also in terms of operability and control.

This should push forward methods for easily debugging the applications running in minimal VMs. Until these applications can be debugged as easily as standard Linux containers, adoption by most practitioners will be slow, as the learning curve is higher.

Finally, isolation is not the only way to secure applications. The last paper could inspire others to devise new methods of automating ways to deter attackers.

Giving operators a usable means of securing the methods they use to deploy and run applications is a win for everyone. Keeping the usability-focused abstractions provided by containers, while finding new ways to automate security and defend against attacks, is a great path forward. 

Jessie Frazelle works for Microsoft in the cloud organization. She was a maintainer of Docker and has been a core contributor to many different open source projects in and out side of the container ecosystem.

Copyright held by author/owner.

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

Automation and a little discipline allow better testing, shorter release cycles, and reduced business risk.

BY THOMAS A. LIMONCELLI

SQL Is No Excuse to Avoid DevOps

A FRIEND RECENTLY said to me, “We can’t do DevOps, we use a SQL database.” I nearly fell off my chair. Such a statement is wrong on many levels.

“But you don’t understand our situation!” he rebuffed. “DevOps means we’ll be deploying new releases of our software more frequently! We can barely handle deployments now and we only do it a few times a year!”

I asked him about his current deployment process.

“Every few months we get a new software release,” he explained. “Putting it into production requires a lot of work. Because we use SQL, the deployment looks something like this: First, we kick out all the users and shut down the application. Next the DBAs (database administrators) modify the database schema. Once their work is done, the new software release is installed

and enabled. The process takes many hours, so we tend to do it on the weekend, which I hate. If it fails, we have to revert to the backup tapes and restore everything from scratch and start again.”

He concluded, “Just scheduling such an event takes weeks of negotiation. We usually lose the negotiation, which is why we end up doing it on the weekend. Doing this every few months is painful and the number-one source of stress around here. If we had to do this for weekly releases, most of us would just quit. We would have no weekends! Heck, I’ve heard some companies do software releases multiple times a day. If we did that, our application would always be down for upgrades!”

Wow. There is a lot to unpack there. Let me start by clearing up a number of misconceptions, then let’s talk about some techniques for making those deployments much, much easier.

First, DevOps is not a technology, it is a methodology. The most concise definition of DevOps is that it is applying Agile/lean methods from source code all the way to production. This is done to “deliver value faster,” which is a fancy way of saying reducing the time it takes for a feature to get from idea to production. More frequent releases means less time a newly written feature sits idle waiting to be put into production.

DevOps does not require or forbid any particular database technology—or any technology, for that matter. Saying you can or cannot “do DevOps” because you use a particular technology is like saying you cannot apply Agile to a project that uses a particular language. SQL may be a common “excuse of the month,” but it is a weak excuse.

I understand how DevOps and the lack of SQL databases could become inexorably linked in some people’s minds. In the 2000s and early 2010s companies that were inventing and popularizing DevOps were frequently big websites that were, by coincidence, also popularizing NoSQL (key/value store) databases. Linking the two, however, is confusing correlation with causation. Those same companies were also populariz-



ing providing gourmet lunches to employees at no charge. We can all agree it is not a prerequisite for DevOps.

Secondly, I'm not sure if someone can "do DevOps." You can use DevOps techniques, methods, and so on. That said, people use that phrase often enough that I think I have lost that battle.

My friend and I discussed his situation further, and soon he realized that DevOps would not be impossible; it would simply be a difficult transition. Once the transition was complete, however, life would actually be much easier.

My friend had one more concern. "Look," he confessed, "these deployments are risky. Every time we do one I risk the company's data and, to be honest, my job. I just don't want to do them. Doing them every few months is stressful enough. Doing them more frequently? No, sir, that's just irresponsible."

As I discussed in a previous article ("The Small Batches Principle," *Communications*, July 2016), when something is risky there is a natural inclination to seek to do it less. Counterintuitively, this actually in-

creases risk. The next time you do the risky thing, you will be even more out of practice, and the accumulated changes to the surrounding environment become larger and larger, making failure-by-unknown-side-effect nearly guaranteed. Instead, DevOps takes the radical stance that risky things should be done *more frequently*. The higher frequency exposes the minor (and major) issues that have been swept under the rug because "this happens only once a year." It forces us to automate the process, automate the testing of the process, and make the process so smooth that risk is reduced. It gives the people involved more practice. Practice makes perfect. Rather than running away from what we fear, it bravely takes risk head on and overcomes it. Like anyone who has experienced post-op recovery, you repeat the exercise until it is no longer painful.

There is always some fixed cost to deploy. You should always, in principle, be driving down the fixed cost of deployment toward zero. Increasing

deployment frequency without driving down that fixed cost is detrimental to the business and irresponsible.

The rest of this article describes two practices that enable rapid releases in an environment that uses SQL. Implementing them requires developers, quality assurance, and operations to get out of their silos and collaborate, which is unheard of in some organizations but is the essence of DevOps. The result will be a much smoother, less painful, and certainly less stressful way of conducting business.

Technique 1: Automated Schema Updates

In the old methodology, any schema change requires the entire application to be shut down while a team of experts (or one very overworked DBA) modifies the schema manually. If you are going to do fully automated deployments, you need to have fully automated schema updates.

To that end, the application should manage the schema. Each version of the schema should be numbered. An ap-

The Five Phases of a Live Schema Change

1. **The running code reads and writes the old schema, selecting just the fields that it needs from the table or view. This is the original state.**
2. **Expand:** The schema is modified by adding any new fields but not removing any old ones. No code changes are made. If a rollback is needed, it's painless because the new fields are not being used.
3. **Code is modified to use the new schema fields and pushed into production. If a rollback is needed, it just reverts to phase 2. At this time any data conversion can be done while the system is live.**
4. **Contract:** Code that references the old, now unused, fields is removed and pushed into production. If a rollback is needed, it just reverts to phase 3.
5. **Old, now unused, fields are removed from the schema. In the unlikely event that a rollback is needed at this point, the database would simply revert to phase 4.**

plication starts with schema version 1. That value is stored in the database (imagine a one-row table with a single field that stores the value "1"). When the application starts, it should know that it is compatible with schema version 1, and if it doesn't find that version in the database, it refuses to run.

To automate schema updating, however, the next release of the software knows it requires version 2 of the schema, and knows the SQL command that will upgrade a version 1 schema to version 2. On startup, it sees the version is 1, runs the appropriate schema upgrade command, updates the version number stored in the database to 2, and then proceeds to run the application.

Software that does this typically has a table of SQL schema update commands. The command in array index n upgrades the schema from version $n-1$ to n . Thus, no matter which version is found, the software can bring the database to the required schema version. In fact, if an uninitialized database is found (for example, in a testing environment), it might loop through dozens of schema changes until it gets to the newest version. Not every software release requires a schema change; therefore, separate version numbers are used for schema and software.

There are open source and commercial systems that implement this process. Some of these products are more sophisticated than others, supporting a variety of languages, database systems, error-handling sophistication, and whether or not they also support rollbacks. A Web search for "sql change automation" will find many. I am most

familiar with the open source projects Mayflower for .NET code (<https://github.com/bretcope/Mayflower.NET>) and Goose for Go (<https://bitbucket.org/liamstask/goose>).

Schema modifications used to lock the database for minutes and possibly hours. This would cause applications to time out and fail. Modern SQL databases have reduced or eliminated such problems, thanks to lockless schema updates and online reindexing features. These features can be found in all recent SQL products, including open source products such as MariaDB, MySQL, and PostgreSQL. Check the documentation for details of what can and cannot be done without interruption.

Once your software uses these techniques, adopting continuous integration (CI) becomes significantly easier. Your automated testing environment can include tests that build a database in the old schema, upgrade it, and run the new software release. Your schema upgrade process may be tested hundreds of times before it goes into production. This should bring new confidence to the process, reduce the risk of schema upgrades, and decouple the DBAs' personal involvement in upgrades. They will appreciate getting their weekends back.

My favorite part of this technique is that your schema is now being treated like code. Manual work at the console has been eliminated and you have gained the ability to do the process over and over—in developer sandboxes, testing environments, user acceptance test (UAT) environments, and production. You can run the process multiple times, fixing and fine-tuning

it. Now that it is code, you can apply the best code-management and software-engineering techniques to it.

Technique 2: Coding For Multiple Schemas

How can you upgrade a database schema in a distributed computing environment?

Imagine a typical Web-based application that is many instances (replicas) of the same software running behind a Web load balancer. Each instance receives its share of the HTTP traffic. The instances access the same database server.

When the software is tightly coupled to the database schema it becomes impossible to perform software upgrades that require a database schema change. If you first change the schema, the instances will all die or at least get confused by the change; you could run around upgrading the instances as fast as possible, but you have already lost the game because you suffer an outage.

Ah ha! Why not upgrade the instances first? Sadly, as you upgrade the instances' software one by one, the newly upgraded instances fail to start as they detect the wrong schema. You will end up with downtime until the schema is changed to match the software.

The obvious solution is to defy the laws of physics and change the database schema at the exact same time as you upgrade the software on all the instances. If you could do that, everything would be just fine.

Sadly, ACM has a policy against defying the laws of physics, as do most employers. This is why the traditional method is to shut down the entire application, upgrade everything, and then bring it back online. It's the best we can do until our friends at IEEE figure out how to pause time.

Whether you stop the world by defying physics or by scheduling downtime, you have introduced an even bigger problem: You have made many individual changes, but you don't know if any of them were successful until the system is running again. You also don't know which of the accumulated changes caused things to break.

Such "big bang" changes are risky. It is less risky to make and validate the changes one at a time. If you make multiple changes all at once, and there is a problem, you have to start binary search-

ing to figure out which change caused the problem. If you make one change at a time, and there is a failure, the search becomes a no-brainer. It is also easier to back out one change than many.

Heck, even Google, with its highly sophisticated testing technologies and methodologies, understands that subtle differences between the staging environment and the production environment may result in deployment failures. They “canary” their software releases: upgrading one instance, waiting to see if it starts properly, then upgrading the remaining instances slowly over time. This is not a testing methodology, this is an insurance policy against incomplete testing—not that their testing people are not excellent, but nobody is perfect. The canary technique is now an industry best practice and is even embedded in the Kubernetes system. (The term *canary* is derived from “canary in a coalmine.” The first instance to be upgraded dies as a warning sign that there is a problem, just as coal miners used to bring with them birds, usually canaries, which are more sensitive to poisonous gas than humans. If the canary died, it was a sign to evacuate.)

Since these problems are caused by software being tightly coupled to a particular schema, the solution is to loosen the coupling. These can be decoupled by writing software that works for multiple schemas at the same time. This is separating rollout and activation.

The first phase is to write code that doesn’t make assumptions about the fields in a table. In SQL terms, this means SELECT statements should specify the exact fields needed, rather than using SELECT *. If you do use SELECT *, don’t assume the fields are in a particular order. LAST_NAME may be the third field today, but it might not be tomorrow.

With this discipline, deleting a field from the schema is easy. New releases are deployed that don’t use the field, and everything just works. The schema can be changed after all the instances are running updated releases. In fact, since the vestigial field is ignored, you can procrastinate and remove it later, much later, possibly waiting until the next (otherwise unrelated) schema change.

Adding a new field is a simple matter of creating it in the schema ahead of the first software release that uses it. We use Technique 1 (applications manage their

own schema) and deploy a release that modifies the schema but doesn’t use the field. With the right transactional locking hullabaloo, the first instance that is restarted with the new software will cleanly update the schema. If there is a problem, the canary will die. You can fix the software and try a new canary. Reverting the schema change is optional.

Since the schema and software are decoupled, developers can start using the new field at their leisure. While in the past upgrades required finding a maintenance window compatible with multiple teams, now the process is decoupled and all parties can work in a coordinated way but not in lockstep.

More complicated changes require more planning. When splitting a field, removing some fields, adding others, and so on, the fun really begins.

First, the software must be written to work with both the old and new schemas and most importantly must also handle the transition phase. Suppose you are migrating from storing a person’s complete name in one field, to splitting it into individual fields for first, middle, last name, title, and so on. The software must detect which field(s) exist and act appropriately. It must also work correctly while the database is in transition and both sets of fields exist. Once both sets of fields exist, a batch job might run that splits names and stores the individual parts, nulling the old field. The code must handle the case where some rows are unconverted and others are converted.

The process for doing this conversion is documented in the accompanying sidebar “The Five Phases of a Live Schema Change.” It has many phases, involving creating new fields, updating software, migrating data, and removing old fields. This is called the McHenry Technique in *The Practice of Cloud System Administration* (of which I am coauthor with Strata R. Chalup and Christina J. Hogan); it is also called *Expand/Contract in Release It!: Design and Deploy Production-Ready Software* by Michael T. Nygard.

The technique is sophisticated enough to handle the most complex schema changes on a live distributed system. Plus, each and every mutation can be rolled back individually.

The number of phases can be reduced for special cases. If one is only

adding fields, phase 5 is skipped because there is nothing to be removed. The process reduces to what was described earlier in this article. Phases 4 and 5 can be combined or overlapped. Alternatively, phase 5 from one schema change can be merged into phase 2 of the next schema change.

With these techniques you can roll through the most complex schema changes without downtime.


Summary

Using SQL databases is not an impediment to doing DevOps. Automating schema management and a little developer discipline enables more vigorous and repeatable testing, shorter release cycles, and reduced business risk.

Automating releases liberates us. It turns a worrisome, stressful, manual upgrade process into a regular event that happens without incident. It reduces business risk but, more importantly, creates a more sustainable workplace.

When you can confidently deploy new releases, you do it more frequently. New features that previously sat unreleased for weeks or months now reach users sooner. Bugs are fixed faster. Security holes are closed sooner. It enables the company to provide better value to customers.

Acknowledgments

Thanks to Sam Torno, Mark Henderson, and Taryn Pratt, SRE, Stack Overflow Inc.; Steve Gunn, independent; Harald Wagener, iNNOVO Cloud GmbH; Andrew Clay Shafer, Pivotal; Kristian Köhntopp, Booking.com, Ex-MySQL AB. 

Related articles on queue.acm.org

The Small Batches Principle

Thomas A. Limoncelli

<https://queue.acm.org/detail.cfm?id=2945077>

Adopting DevOps Practices in Quality Assurance

James Roche

<https://queue.acm.org/detail.cfm?id=2540984>

Thomas A. Limoncelli is the SRE manager at Stack Overflow Inc. in New York City. His books include *The Practice of System and Network Administration*, *The Practice of Cloud System Administration*, and *Time Management for System Administrators*. He blogs at EverythingSysadmin.com and tweets at [@YesThatTom](https://twitter.com/YesThatTom).

Copyright held by owner/author.
Publication rights licensed to ACM. \$15.00

DOI:10.1145/3210753

In addition to having a detailed understanding of the artifacts they intend to create, designers need to guide the software tools they use.

BY STEFAN SEIDEL, NICHOLAS BERENTE, ARON LINDBERG, KALLE LYYTINEN, AND JEFFREY V. NICKERSON

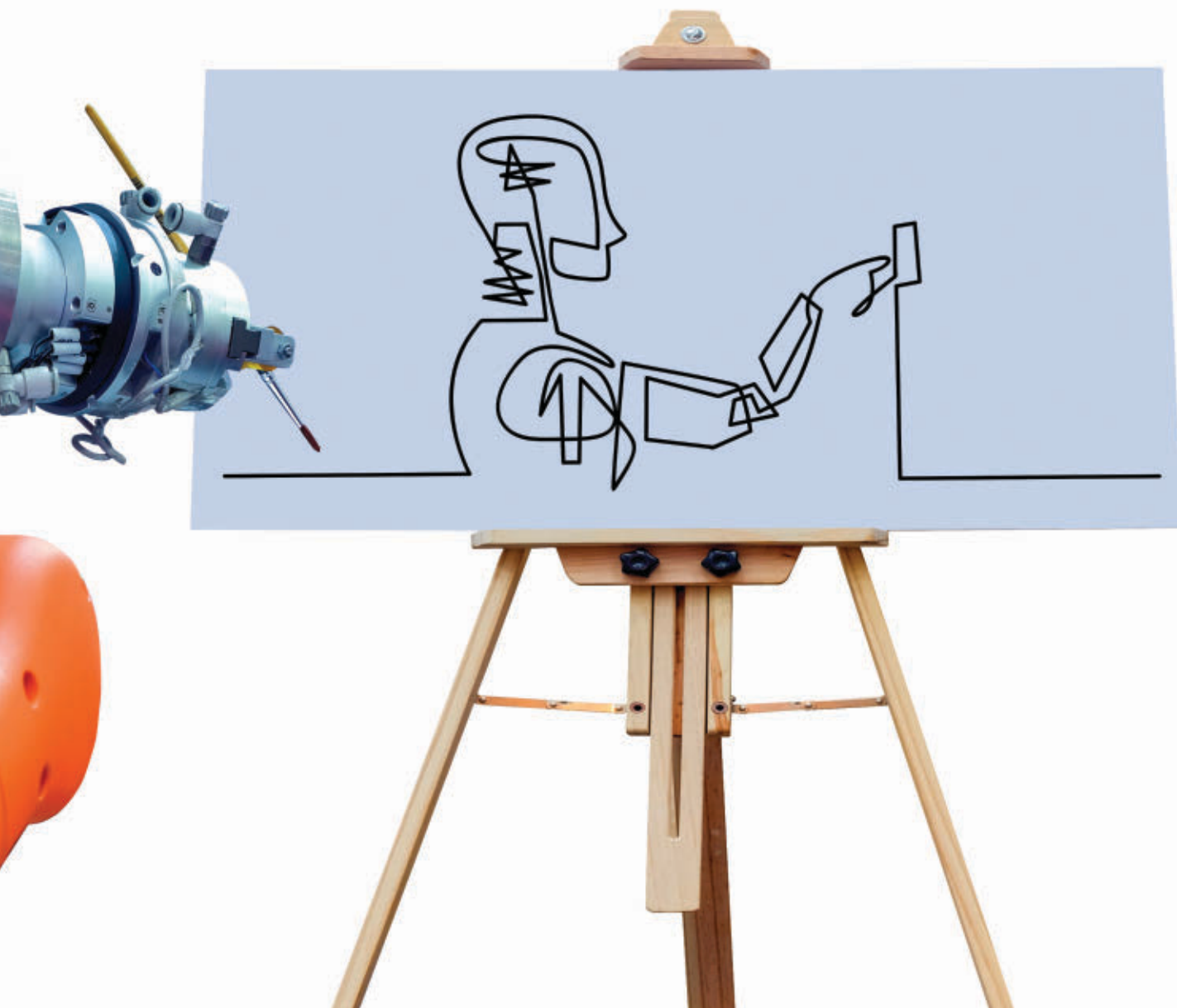
Autonomous Tools and Design: A Triple-Loop Approach to Human-Machine Learning

DESIGNERS INCREASINGLY LEVERAGE autonomous software tools that make decisions independent of the designer. Examples abound in virtually every design field. For example, semiconductor chip designers use tools that make decisions about placement and logic checking. Game designers rely on software that generates initial drafts of virtual worlds. Autonomous tools employ artificial intelligence methods, including machine learning, pattern recognition, meta-heuristics,



» key insights

- Autonomous tools are able to generate remarkable design outcomes, but designers using them also need to change the way they do their design work.
- Designing with autonomous tools requires that designers understand and actively interact with the “mental models” of the tools, in addition to the design artifact and the design process, what we call the “triple loop” model of learning.
- Designers working with autonomous tools need to build capabilities described here in terms of framing, evaluating, and adjusting to navigate this new design process.



and evolutionary algorithms to generate design artifacts beyond any human's capabilities.

A naïve view suggests these tools will someday replace human designers in the design process. An alternative perspective is that humans will continue to play an important role but also that this role is changing. To account for the changing role of humans in design processes powered by autonomous tools, we describe in this article a “triple-loop approach” to human-machine learning. Increasing amounts of design activity are most certainly being carried

out by autonomous tools, but humans must still actively frame, evaluate, and adjust the “mental” models embedded in autonomous tools, in the form of algorithms.^a Organizations employing autonomous tools in their design processes must thus account for these activities in their design processes.

^a We say “mental model embedded in an autonomous tool” to indicate that just as human designers have mental models that guide their design activity, including their use of tools, autonomous tools also have models that guide their design activity. Both types of model change over time.

In what follows, we describe our triple-loop approach, followed by illustrative examples from research into the design of semiconductors, video games, software interfaces, and artificial intelligence. We conclude by identifying practices that enable designers to frame, evaluate, and adjust the mental models embedded in autonomous tools.


Design as Triple-Loop Human-Machine Learning

Design processes are a form of knowledge-intensive work that relies on designers' capacity to learn. In his semi-


nal work, Chris Argyris^{2,3,4} explained how humans, in knowledge-intensive work, follow a double-loop process of learning. In the context of design work, the first loop involves learning about the design artifact. As designers experiment with alternatives, they correct errors and respond to feedback on design results (see Figure 1, loop 1). The second loop involves a designer's reflection on the ongoing process of design. Over time, designers learn, through reflection, to adjust their approaches and discover new processes and perhaps incorporate new tools that help them expand their thinking around the process of design. The second loop captures meta-level learning about the design process (see Figure 1, loop 2), highlighting how designers reflect on the mental models—goals, cognitive rules, and reasoning—they apply.

Triple-loop human-machine learning occurs whenever humans and autonomous computational tools interact in generating design outcomes. It is important for designers to understand how their own mental models interact with mental models embedded in the logic of autonomous tools. This process is distinct from conventional design work where tools are limited to supporting ongoing design tasks but do not play an independent role in shaping design outcomes.

Argyris calls mental models “master programs.” In the case of designing with autonomous tools, the master program of the designer—the “designer’s mental model”—may not be aligned with the master program of the autonomous tool, or “autonomous tool mental model,” for a variety of reasons, including, for example, that the design activity usually involves more than one person; the designer using the tool is probably not the same person who programmed the tool; multiple designers may have different conceptions about what a master program does; and these conceptions may differ from what the programmers intended. Moreover, programmers may move on to other projects, along with the designers who originally informed the design of the tools; increasingly, neither the tool programmers nor the designers understand the implications of their decisions on what the tool is able to do. The mental models of designers



It is important for designers to understand how their own mental models interact with mental models embedded in the logic of autonomous tools.



and those embedded in autonomous tools as master programs for design activity thus capture a mutual learning process, suggesting a third loop in the classic model of the design process (see Figure 2).

The first loop in the triple-loop model is similar to the original loop in the double-loop model, involving designers and tools interacting to generate design outcomes. However, in the triple-loop model, it is the tool that primarily generates the design alternatives. A given configuration of the tool generates alternatives from a set of input parameters and then evaluates them according to a set of predefined criteria.

The second loop can take two alternative forms—human learning or machine learning—as in Figure 2, loop 2a and loop 2b. From a human perspective, the second loop involves the human designer evaluating the alternatives and modifying input parameters, tool settings, and evaluation criteria for a given design problem in order to run the next round of generating design alternatives. The second loop, from a machine perspective, involves the program learning from designer feedback in the design process in order to modify itself and improve its model so it can generate better alternatives in subsequent rounds of design activity.

The third loop involves human designers learning about the mental model embedded in the tool and/or the tool learning about the human designers’ mental models—through either explicit feedback or analyzing the usage patterns of the human designers. The machine models of designers are sometimes called “user models.”¹ When machines run learning algorithms, the human designers may not fully understand the computations. What they thought the tool would do may or may not be what it actually does or was even designed to do, though designers collect feedback that can help them align their mental models and the mental models embedded in the tool (such as by adjusting the algorithm used). This process of learning about the mental model embedded in the autonomous tool and modification of the tool constitutes a second meta-level of learning during design processes involving

autonomous tools. Moreover, the tool may change its own model as it relates to what the human wants and how the human perceives the tool; this may result in changes in the interface or the design parameters being applied. Much like two people with different mental models learn from each other and work together to reconcile their mental models, autonomous tools and humans likewise have different models related to design goals and processes they may seek to reconcile through various loops of learning.

Illustrations

Here, we provide four examples of triple-loop human-machine learning, including in semiconductor design, software interface design, video game design, and artificial intelligence design. They highlight different aspects of the interaction between designers and autonomous tools.

Semiconductor design at Intel and AMD. Since the early 2000s, a new breed of tooling strategies based on genetic algorithms has emerged in semiconductor design,⁶ commonly called “physical synthesis.” Such tools offer a powerful way to improve the productivity of physical chip design by autonomously generating full layout solutions for whole sections of a chip (such as caches and USB ports). Major semiconductor manufacturers, including Intel and AMD, use the program-synthesis approach to generate full-layout designs of particular chip sections for a given set of parameter values. A program-synthesis designer starts each design cycle by defining a new set of design-parameter values that specify directions and constraints for the design search to be carried out by the tool (see Figure 3). When a solution is found through such search, the tool autonomously delivers a complete design solution for the given layout problem. After each such cycle, the designer manipulates the design by modifying the parameters based on the design outcomes generated during the previous cycle. Designers refer to the automated generation of design alternatives as “experiments” for a given set of parameters and interact with the algorithmic results in order to evaluate alternatives, given the input parameters and design goals (see Figure 2,

loop 1). Designers then learn from the experiments in a way that helps them improve the input parameters for the next round of experiments, as in Figure 2, loop 2. Developers of the algorithms interact with the chip designers in order to learn how the chip designers

change the parameters interactively after evaluating design outcomes. The developers learn about the effects of the mental models embedded in the tools, as well as the designers’ mental models. This involves learning about the specific assumptions of the design-

Figure 1. Double-loop learning; based on Argyris.^{3,4}

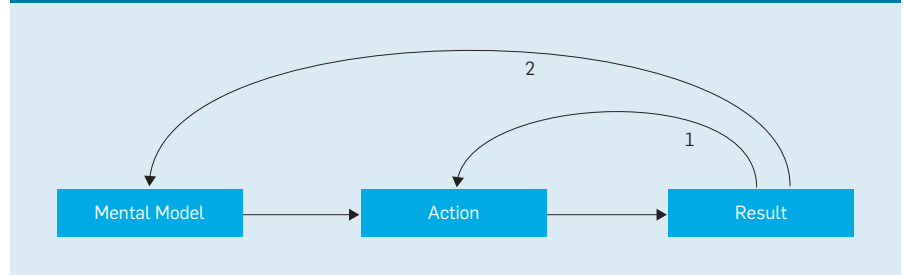


Figure 2. Triple-loop human-machine learning with autonomous tools.

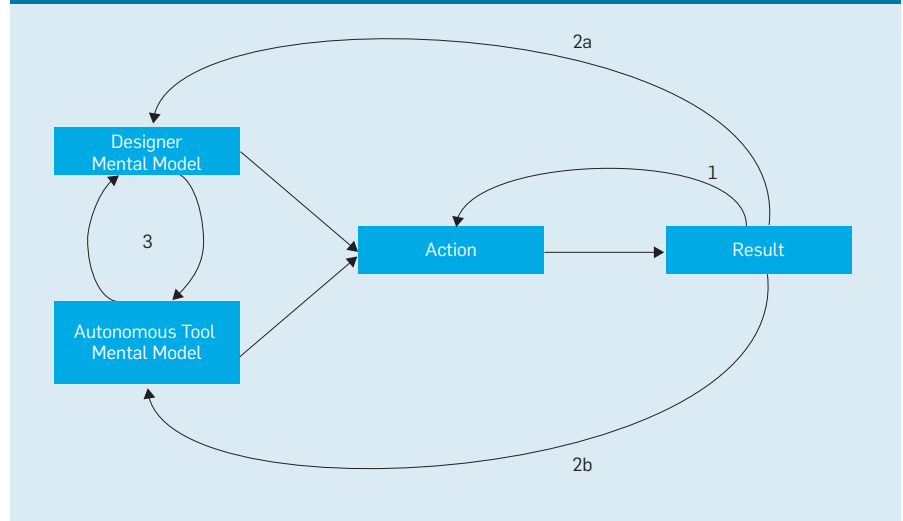
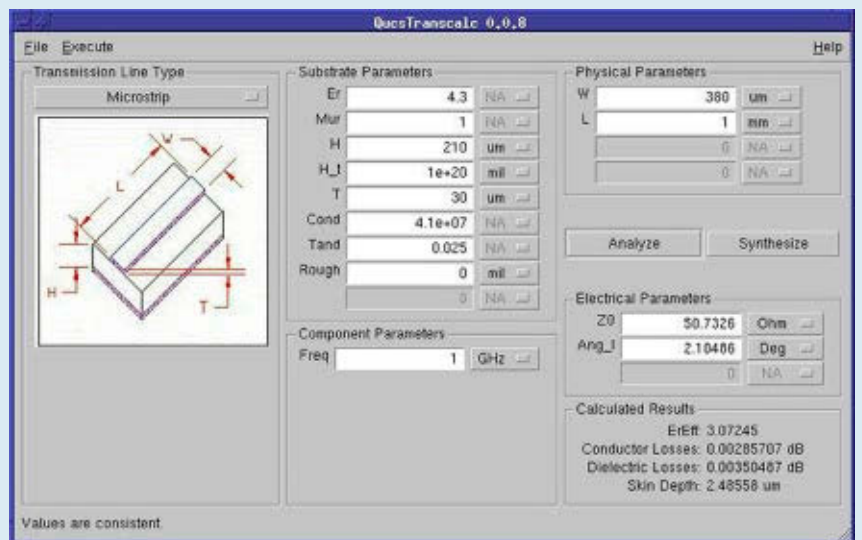


Figure 3. Computational design tool for semiconductor design.

“Quite Universal Circuit Simulator” is hosted on Sourceforge (<http://sourceforge.net/projects/qucs/>) and made available under the GNU General Public License version 2.0.



ers while rooting out inefficiencies of the tools by updating and rewriting the source code for the tools, as in Figure 2, loop 3. Tool developers then carefully calibrate the mental models embedded in the autonomous tool to fit with the mental models of the designers.

Software interface design at Adobe Labs. Interface designers today make extensive use of machine learning to improve interface designs. For example, researchers at Adobe Labs created tools intended to control complex user processes.¹³ In particular, visual designers wanted to be able to control procedural models that render complex shapes (such as trees and bushes) by growing them artificially from digital seeds into mature plants. Designers had difficulty harnessing these models because the array of approximately 100 parameters controlling such a growth process had to be manipulated in unison, thereby making it an incredibly complex problem space. Machine

learning provided a solution, enabling Adobe Labs’ developers to reduce this high-dimensionality problem to a three-dimensional space comprehensible by human designers. Moreover, the three-dimensional space was controllable through three slider bars. Using this intuitive interface, designers can more easily configure the model to match a given image. The example shows that autonomous tools do not have to correspond precisely to the mental models of humans. Instead, they often provide an expressive but low-dimensional interface. Humans learn through interacting with this interface, and the machine and the human both participate in learning. The interface amplifies the ability of a human designer to explore a large design space. In this design process, the autonomous tools create an interface a designer can use to generate alternative outputs, as in Figure 2, loop 1. Through practice, designers learn

how to control the outputs (see Figure 2, loop 2). Over time, the designer’s experience can be used to refine the interface, as in Figure 2, loop 3. In such user-interface design, the machine-learning system begins with the goal of reducing the dimensionality of the interface from 100 dials to three slider bars. Although the mental model of the human can never be entirely aligned with the underlying mental model embedded in the tool due to the limits of human cognition, the new interface provides a level of abstraction necessary for effective learning.

Designing Landscapes at Ubisoft. Tools have a long track record in video game development. Algorithmically generated content may include a variety of game elements, including textures, buildings, road networks, and component behaviors like explosions.⁷ In extreme cases, autonomous tools are able to generate large parts of the game content that only later are


Figure 4. Procedural generation in *Ghost Recon Wildlands*; source: Ubisoft.




combined with specific handcrafted elements. Hence, the interplay of automated and manual generation of content is crucial to game development, as humans are looking for a rich and unique experience, and undirected automated generation could lead to results that are not perceived as authentic. Ubisoft's *Ghost Recon Wildlands*, an action-adventure game originally published in 2017, is an example in which designers used autonomous tools to generate the game environment.¹² Designers handcrafted elements in the game environment while algorithms procedurally generated much of the background content. In this process, the tools would generate, say, large amounts of detailed terrain; the Figure 4 screenshots show how the terrain evolved as a road network was added procedurally, based on a few waypoints on a landscape. The designers would then modify the terrain further and create extra detail.

Some areas of the game environment were still generated in a traditionally manual fashion. The combined process required selecting appropriate tools and models that would align with the game idea in a way that was shared by a team of Ubisoft designers and developers. This example of “hybrid” development highlights how, although the tool autonomously generated significant portions of the design, designers still had a significant role in the design process. In such a “hybrid” model of autonomous design, the tool and the designer jointly generate the design in a given problem space (see Figure 2, loop 1); based on feedback generated by the tool, designers make adjustments and design decisions (see Figure 2, loop 2); and the team learns holistically from the experience of using the tool, reflecting on the alignment of their mental models with the outcomes of the use of the tool (see Figure 2, loop 3).

Artificial intelligence design and Atari games. Many researchers today engage in designing artificial intelligence solutions, using machine learning in their solutions. For example, researchers recently created an artificial intelligence system to play Atari games. The experimental system was a deep convolutional neural network with inputs wired to a video game display



Tool developers then carefully calibrate the mental models embedded in the autonomous tool to fit with the mental models of the designers.



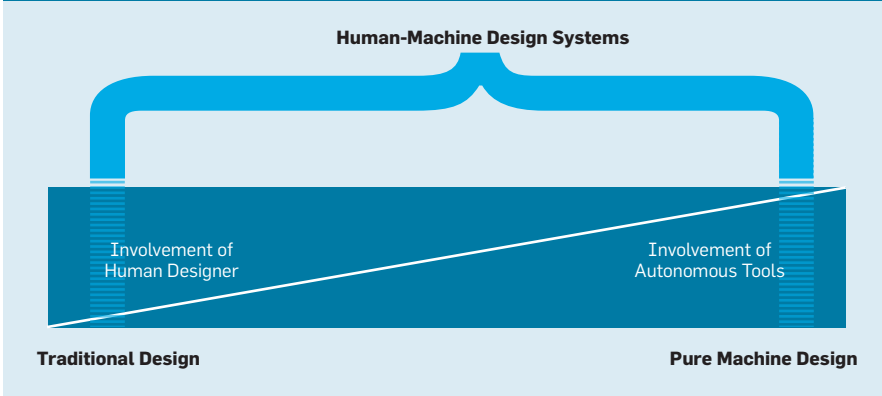
and outputs wired to a joystick controller.¹⁰ The system used a reinforcement-learning algorithm to train itself. After training, it scored as well or better than humans on 29 out of 49 Atari games. But some games proved challenging for the algorithm, and games that require a player to acquire objects early in the game that will prove useful only near the end were especially difficult for the algorithm. Taking such decisions across long time scales is more difficult to learn than are subsecond reactions to, say, attacking enemies. As a result, the designers of the system made modifications to the training algorithm that significantly increased its performance on difficult games, though they still require hundreds of hours of training.⁸ In this case, when the algorithm is exposed to gameplay events, it learns, as in Figure 2, loop 1). When the machine fell short on certain games, the designers adjusted the training regimen (see Figure 2, loop 2). In creating the system, the designers first created an environment tools explore and receive feedback on, similar to the way humans interact with the physical environment.

At a higher level of abstraction, the process can be viewed as part of a meta learning process in which humans create autonomous machines, monitor their progress, and iterate across multiple configurations of the machines while ultimately confronting the limits of both machine and human intelligence (see Figure 2, loop 3). The shortcomings of the algorithm indeed pose a challenge some researchers argue are best addressed through techniques developed in cognitive science.⁹ Even the use and development of autonomous systems are examples of triple-loop learning, as these systems need to be designed, monitored, and improved by humans.

Designers and Triple-Loop Design Activities

Traditional designers intentionally craft artifacts by applying their deep knowledge of materials and tools, moving them forward toward a preferred, future condition.¹¹ However, autonomous tools change the role of designers, including focus, activities, and required skills. Designers are increasingly focused on manag-

Figure 5. Shifting control in design processes.



New design practices.

Design Practice	Example	Description
Framing	Parameterization	Designers have a deep understanding of the software tool and its parameters, as well as some understanding of the consequences of setting specific parameters; and
		They formulate hypotheses with regard to what sets of inputs will have the desired design consequences in lieu of carrying out the entire design process in an incremental, iterative, primarily manual fashion.
Evaluation	Process Analysis	Designers evaluate the overall design outcome, investigating sources of misalignment, as in assumptions embedded in the tool; and
		They formulate hypotheses about the process and test whether they hold.
Adjustment	Modifying Algorithms	Designers continuously align their mental models with mental models embedded in the autonomous tools;
		They consider how changes in the constraints and propensities of the tool may require changes in their mental models in terms of assumptions and goals; and
		They consider how changes in assumptions and goals may require changes in the mental models embedded in the autonomous tools.

ing tools—and their embedded mental models—and understanding the often-surprising behaviors of tools as they generate design artifacts. This new type of designer needs a better understanding of the tools, in addition to a detailed understanding of the underlying anatomy of the artifact to be designed. The locus of control of the design process is moving away from the designer toward the tool and its underlying model. An important causal force behind the tool is the tool designer who defines and implements the algorithmic choices. The tool designer thus creates the initial version of the mental model embedded in the tool, a model that will change as the tool itself learns. As illustrated in our examples, there can be different de-

grees to which control shifts toward the tools and away from the designer (see Figure 5).

Rather than incrementally build and modify design artifacts, designers become engaged in new design practices that fall into three categories: framing, evaluation, and adjustment.


Framing. “Framing” occurs as designers, based on their mental models, construct their understanding of the problem situation or task and thus how the tool, with its underlying, embedded mental model, should be used, thereby making decisions about the solution space. The actual design activity is thus informed by both the mental model of the designer and the mental model embedded in the autonomous tool.

Framing in the examples outlined earlier notably involves specification of varying sets of inputs that can include numerical and non-numerical variables, thus enabling the tool to do its work. This is the process of “parameterization,” which requires a deeper understanding of the tool, as well as an intuitive understanding of both the problem space being worked on and the solution space of the tool so hypotheses can be formulated with regard to what sets of inputs will have the desired design consequences. Parameterization thus precedes the actual design process (see Figure 2, loop 1) and follows the evaluation of the design product.

Evaluation. Once the autonomous tool has generated outcomes, these outcomes must be evaluated to inform decisions about further design actions (see Figure 2, loop 1), as well as to inform the mental model of the designers (see Figure 2, loop 2a) and the mental models embedded in autonomous tools being used (see Figure 2, loop 2b). While loop 1 activities lead to a different use of the tool through, say, a different set of input parameters, loop 2 activities lead to changes in mental models that affect future design decisions.

As parameters can be changed and various design alternatives explored, autonomous tools allow more iterations of the design outcome and thus for experimentation. For instance, Ubisoft’s video game *Ghost Recon Wildlands* presents an experience to users that is possible only because a relatively small team of designers could experiment with various computationally generated design outcomes.

Because the algorithmic processes of autonomous tools are typically complex, they tend to overwhelm humans’ bounded cognitive abilities. It is difficult for human designers to predict what the tools will produce, so they must be able to evaluate the design products generated by the tool. Such evaluation may range from specific aspects of the outcome (such as elements in the game space of a video game) to some holistic outcome (such as in the process of generating the layout for a semiconductor chip). Once a tool has been run, and has generated outputs, designers evaluate the

Foundation under grants IIS-1422066, CCF-1442840, IIS-1717473, and IIS-1745463. 

References

- Allen, R.B. Mental models and user models. Chapter in *Handbook of Human-Computer Interaction, Second Edition*, M.G. Helander, T.K. Landauer, and P.V. Prabhu, Eds. North-Holland, Amsterdam, the Netherlands, 1997, 49–63.
- Argyris, C. The executive mind and double-loop learning. *Organizational dynamics* 11, 2 (Autumn 1982), 5–22.
- Argyris, C. Teaching smart people how to learn. *Harvard Business Review* 69, 3 (May–June 1991).
- Argyris, C. *Double-loop learning*. Chapter in *Wiley Encyclopedia of Management*, C.L. Cooper, P.C. Flood, and Y. Freezey, Eds. John Wiley & Sons, Inc., New York, 2014.
- Austin, R.D. and Devin, L. *Artful Making: What Managers Need to Know About How Artists Work*. Financial Times Press, Upper Saddle River, NJ, 2003.
- Brown, C. and Linden, G. *Chips and Change: How Crisis Reshapes the Semiconductor Industry*. MIT Press, Cambridge, MA, 2009.
- Hendrikx, M., Meijer, S., Van Der Velden, J., and Iosup, A. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* 9, 1 (Feb. 2013), 1.
- Jaderberg, M., Mnih, V., Czarnecki, W.M., Schaul, T., Leibo, J.Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. In *Proceedings of the Fifth International Conference on Learning Representations* (Toulon, France, Apr. 24–26, 2017).
- Lake, B., Ullman, T., Tenenbaum, J., and Gershman, S. Building machines that learn and think like people. *Behavioral and Brain Sciences* 40, E253 (2017).
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., and Petersen, S. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (Feb. 2015), 529–533.
- Sennet, R. *The Craftsman*. Allen Lane, London, U.K., 2008.
- Werle, G. and Martinez, B. *Ghost Recon Wildlands: Terrain tools and technologies*. Game Developers Conference (San Francisco, CA, Feb. 27–Mar. 3, 2017); https://666uille.files.wordpress.com/2017/03/gdc2017_ghostreconwildlands_terrainandtechnologytools-onlinevideos1.pdf
- Yumer, M.E., Asente, P., Mech, R., and Kara, L.B. Procedural modeling using autoencoder networks. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, Nov. 11–15). ACM Press, New York, 2015, 109–118.

Stefan Seidel (stefan.seidel@uni.li) is a professor and the Chair of Information Systems and Innovation at the Institute of Information Systems at the University of Liechtenstein, Vaduz, Liechtenstein.

Nicholas Berente (nberente@nd.edu) is an associate professor of IT, analytics, and operations in the Mendoza College of Business at the University of Notre Dame, Notre Dame, IN, USA.

Aron Lindberg (alindberg@stevens.edu) is an assistant professor of information systems in the School of Business of Stevens Institute of Technology, Hoboken, NJ, USA.

Kalle Lyytinen (kalle@case.edu) is a Distinguished University Professor and Iris S. Wolstein Professor of Management Design at Case Western Reserve University, Cleveland, OH, USA.

Jeffrey V. Nickerson (jnickers@stevens.edu) is a professor of information systems and the Associate Dean of Research of the School of Business at Stevens Institute of Technology, Hoboken, NJ, USA.

Copyright held by authors.

outputs in a way that leads to new hypotheses with regard to what sets of input parameters should be tested in the next batch of experiments.

Adjustment. Evaluation by human designers can lead to the adjustment of parameter values (see Figure 2, loop 1) or even to changes in the mental model embedded in the autonomous tool, resulting in changes in the algorithms used; moreover, it might also change the mental models of human designers in terms of goals, cognitive rules, and underlying reasoning. Changes of the mental model embedded in the autonomous tool could change the tool’s constraints and propensities and require changes to the mental models of designers; likewise, changes in the mental models of designers could require changes to the algorithms and thus the mental model embedded in the tool. Following each experiment, designers might thus have to continuously reconcile their mental models with the counterpart models embedded in the autonomous tool (see Figure 2, loop 3).

In order to change the mental model embedded in an autonomous tool, designers have to modify the underlying algorithm. The original mental model embedded in the tool—the one implemented by the tool designer—can thus evolve over time.

Competencies related to these design practices become critically important for achieving complex design outcomes. Having a detailed understanding of the designed artifact, as well as of the consequences of specific local decisions, becomes less important. This explains why, in the context of, say, chip design, we see software engineers displacing electrical engineers with a deep understanding of physical aspects of chip design. Because the design is increasingly mediated by software that needs to be parameterized and evaluated, designers’ software skills become crucial; the table here outlines key implications in terms of emergent interrelated designer activities.

Some substitution of human design activity through autonomous tools is indeed occurring. To a certain degree, demand for specific, manual-type competencies in design professions, including software de-

velopment, is decreasing, while the demand for skills focused on how to work with software tools is increasing. Organizations need to engage more effectively with new forms of autonomous tools supporting design processes. This is not simply a shift of tasks from humans to machines but a deeper shift in the relationship between humans and machines in the context of complex knowledge work. The shift puts humans in the role of coaches who guide tools to perform according to their expectations and requirements (see Figure 2, loop 1) or in the role of laboratory scientists conducting experiments to understand and modify the behavior of complex knowledge artifacts (see Figure 2, loop 2 and loop 3).

The Road Ahead

Engaging with autonomous tools requires reshaping the competencies designers need. Designers envision certain results and thus need to interact with autonomous tools in ways that help them realize their design vision. At the same time, the use of autonomous tools opens unprecedented opportunities for creative problem solving. Consider the example of video game production, where autonomous tools are increasingly able to procedurally generate artifacts of a scope and scale that was not possible in the past. Future designers will constantly be challenged to rethink their mental models, including their general approach to design. The continuous reconciliation of mental models embedded in both designer cognition and their tools is an extension of traditional design processes that involve artful making where human actors gradually adjust their mental models to converge on solutions.⁵

The proposed three-loop model contributes to the ongoing debate on how artificial intelligence will change knowledge work, challenging knowledge workers to operate at a different level. Designers may become increasingly removed from the actual artifact but still use tools to create artifacts of a complexity never imagined before.

Acknowledgments

This material is based in part on work supported by the National Science

DOI:10.1145/3210752

Featuring the various dimensions of data management, it guides organizations through implementation fundamentals.

BY SERGIO ORENGA-ROGLÁ AND RICARDO CHALMETA

Framework for Implementing a Big Data Ecosystem in Organizations

ENORMOUS AMOUNTS OF data have been generated and stored over the past few years. The McKinsey Global Institute reports this huge volume of data, which is generated, stored, and mined to support both strategic and operational decisions, is increasingly relevant to businesses, government, and consumers alike,⁷ as they extract useful knowledge from it.¹¹

There is no globally accepted definition of “big data,” although the Vs concept introduced by Gartner analyst Doug Laney in 2001 has emerged as a common structure to describe it. Initially, 3Vs were used, and another 3Vs were added later.¹³ The 6Vs that characterize big data today are volume, or very large amounts of data; velocity, or data generated and processed quickly; variety, or a large number of structured and unstructured data types processed;

value, or aiming to generate significant value for the organization; veracity, or reliability of the processed data; and variability, or the flexibility to adapt to new data formats through collecting, storing, and processing.

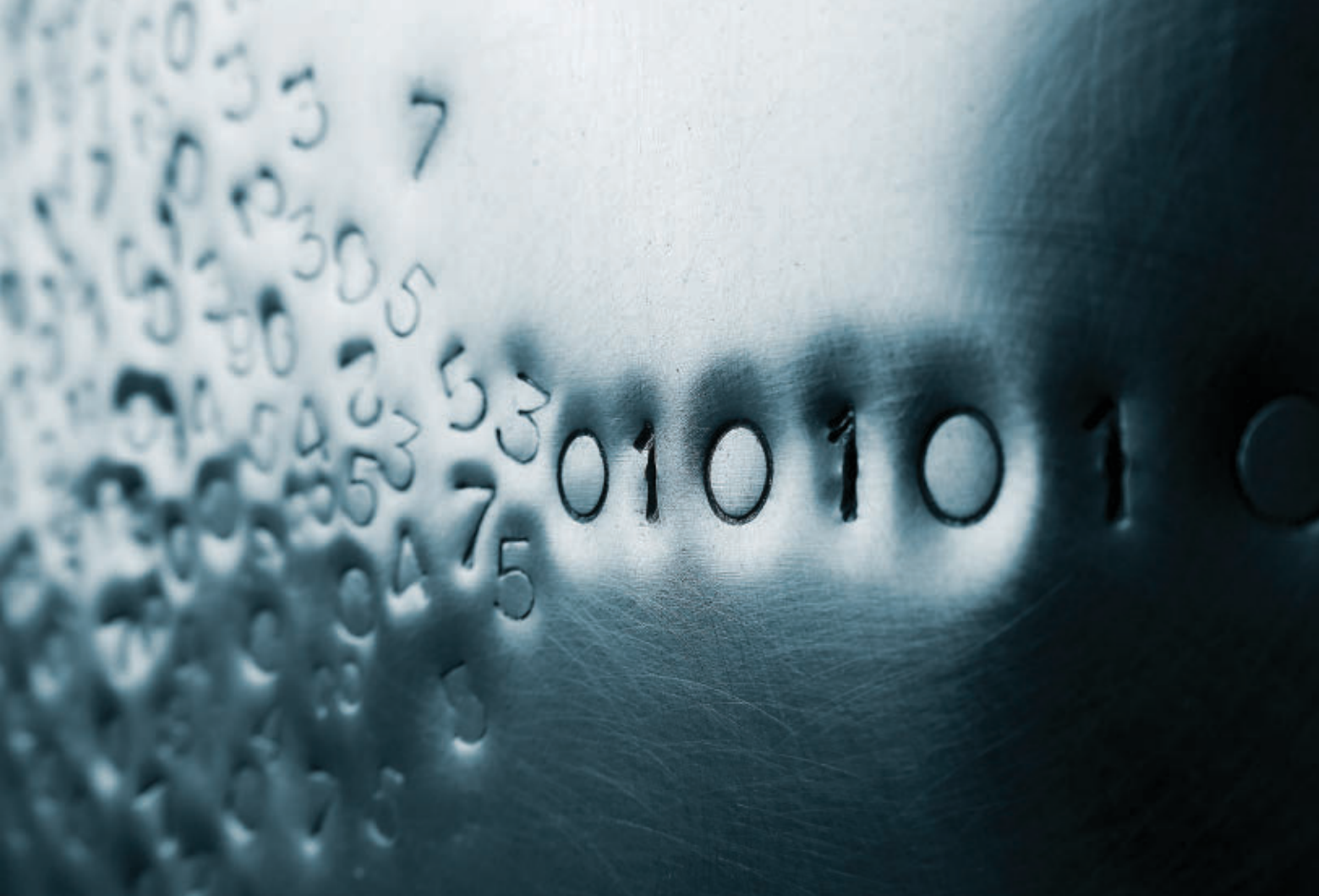
Big data sources can include an overall company itself (such as through log files, email messages, sensor data, internal Web 2.0 tools, transaction records, and machine-generated), as well as external applications (such as data published on websites, GPS signals, open data, and messages posted in public social networks).

This data cannot be managed efficiently through traditional methods¹⁷ (such as relational databases) since big data requires balancing data integrity and access efficiency, building indices for unstructured data, and storing data with flexible and variable structures. Aiming to address these challenges, the NoSQL and NewSQL database systems provide solutions for different scenarios.

Big data analytics can be used to extract useful knowledge and analyze large-scale, complex data from applications to acquire intelligence and extract unknown, hidden, valid, and useful relationships, patterns, and information.¹ Various methods are used to deal with such data, including text analytics, audio analytics, video analytics, social media analytics, and predictive analytics; see the online appendix “Main Methods for Big Data Analytics,” dl.acm.org/citation.cfm?doid=3210752&picked=formats.

» key insights

- **This fresh approach to the problem of creating frameworks helps project managers and system developers implement big data ecosystems in business organizations.**
- **The related literature review of big data for business management covers some of the existing frameworks used for this purpose.**
- **The methodology dimension of the proposed framework covers the big data project life cycle and defines when and how to use the framework's other six dimensions.**



Big data reflects a complex, interconnected, multilayered ecosystem of high-capacity networks, users, and the applications and services needed to store, process, visualize, and deliver results to destination applications from multiple data sources.²⁶ The main components in that ecosystem include properties, infrastructure, life cycle, models and structures, and security infrastructure.¹⁰

Big data and business management.

In order to succeed in today's complex business world, organizations have to find ways to differentiate themselves from their competitors. With the rise of cloud computing, social media, and mobile devices, the quantity and quality of data generated every moment of every day is constantly being enhanced, and organizations need to take advantage of it. If they use data properly, they can become more collaborative, accurate, virtual, agile, adaptive, and synchronous. Data and information are thus primary assets for organizations, with most trying to collect, process, and manage the potential offered by big data.⁵ To take advantage, organizations need to generate or obtain a large

amount of data, use advanced analytical tools, and staff with appropriate skills to manage the tools and the data.³

Big data is a key factor for organizations looking to gain a competitive advantage,⁴ as it can help develop new products and services, make automated strategic and operational decisions more quickly, identify what has happened and predict what will happen in the immediate future, identify customer behavior, guide targeted marketing, produce greater return on investments, recognize sales and market opportunities, plan and forecast, increase production performance, guide customer-based segmentation, calculate risk and market trends, generate business insight more directly, identify consumer behavior from click sequences, understand business disruption, implement product changes that prevent future problems, obtain feedback from customers, calculate price comparisons proactively, recommend future purchases or discounts, and refine internal processes.²⁵

Big data analytics can be seen as a more advanced form of business intelligence that works with structured

company databases, focusing on obtaining reports and indicators to measure and assess business performance. Conversely, big data works with semi-structured and unstructured data from multiple sources, focusing on extracting value related to exploration, discovery, and prediction.⁹

Big data frameworks. Developing and implementing a big data ecosystem in an organization involves not only technology but management of the organization's policies and people.²⁸ A number of frameworks have thus been proposed in the literature.^{8,10,12,14,18,27,28}

A framework might describe concepts, features, processes, data flows, and relationships among components (such as software development), with the aim of creating a better understanding (such as descriptions of components or design) or guidance toward achieving a specific objective.²³ Frameworks consist of (usually interrelated) dimensions or their component parts.


Big data frameworks focus on assisting organizations to take advantage of big data technology for decision making. Each has its good points, although

each also has weaknesses that must be addressed, including that none include all dimensions (such as data architecture, organization, data sources, data quality, support tools, and privacy/security). Moreover, they lack a methodology to guide the steps to be followed in the process of developing and implementing a big data ecosystem, making the process easier. They fail to provide strong case studies in which they are evaluated, so their validity has not been proved. They do not consider the impact of the implementation of big data on human resources or organizational and business processes. They do not consider previous feasibility studies of big data ecosystem projects. They lack systems monitoring and a definition of indicators. They fail to study or identify the type of knowledge they need to manage. Moreover, they fail to define the type of data analysis required to address organizational goals; see the online appendix for more on the frameworks and their features and weaknesses.


In addition to big data frameworks, system developers should also consider big data maturity models that define the states, or levels, where an enterprise or system can be situated, a set of good practices, goals, and quantifiable parameters that make it possible to determine on which of the levels the enterprise stands, and a series of proposals with which to evolve from one level of maturity to a higher level.² Several such models have been proposed,^{1,5,16,24} all focused on assessing big data maturity (the “as is”) and building a vision for what the organization’s future big data state should be and why (the “to be”). There is thus a need for a new framework for managing big data ecosystems that can be applied effectively and simply, accounting for the main features of big data technology and avoiding the weaknesses so identified.

Proposed Framework

In this context, the IRIS (the Spanish acronym for Systems Integration and Re-Engineering) research group at the Universitat Jaume I of Castellón, Spain, has proposed the Big Data IRIS (BD-IRIS) framework to deal with big data ecosystems, reflecting the literature dealing with this line of research. The BD-IRIS framework focuses on data and the tasks of collecting, storing, processing, analyz-



Data and information are thus primary assets for organizations, with most trying to collect, process, and manage the potential offered by big data.



ing, and visualizing necessary to make use of it. However, unlike other frameworks, it focuses not only on operations affecting data but also other aspects of management like human and material resources, economic feasibility, profit estimation, type of data analysis, business processes re-engineering, definition of indicators, and system monitoring.

The BD-IRIS framework includes seven interrelated dimensions (see Figure 1): methodology, data architecture, organization, data sources, data quality, support tools, and privacy/security. The core is the methodology dimension that serves as a guide for the steps involved in implementing an ecosystem with big data technology includes phases, activities, and tasks supported by the six other dimensions. These other dimensions include various techniques, tools, and good practices that support each phase, activity, and task of the methodology. Additionally, they include properties and characteristics that must be fulfilled in certain stages of such development. With the exception of a methodology, the other six dimensions are included in some of the seven frameworks outlined earlier, though none includes all dimensions.

Methodology dimension. This is the main axis of the framework; the other dimensions are techniques, tools, and good practices that support each phase, and the activities and tasks within it. The methodology provides practical guidance for managing an entire project life cycle by indicating the steps needed to execute development and implementation of big data ecosystems. The methodology consists of phases that in turn consist of activities that in turn consist of tasks, whereby each one must be completed before the next one can begin. Table 2 (see in the online appendix) lists the phases and activities that constitute the methodology, along with the main dimensions that support execution of the activities and tasks. The support-tools dimension is not included in Table 2 because it is present or can be present in all tasks of the methodology, as different information technology tools are available to support each of them.

The methodology can be applied in waterfall mode, or sequentially, for each phase, activity, and task. It can also be applied iteratively, whereby the project is divided into subprojects executed in

waterfall mode, with each subproject begun when the previous one has finished; for example, each subproject can cover an individual knowledge block or a tool.

Data architecture dimension. This dimension identifies the proposed steps the software engineer performs during data analysis. The order in which each task is executed in each of the steps and its relationship with the other dimensions of the framework are specified in the methodology dimension. The data architecture dimension is divided into levels ranging from identifying the location and structure of the data to the display of the results requested by the organization. Figure 2 outlines the levels that make up the data architecture, including:

Content. Here, the location and characteristics of the data are identified (such as format and source of required data, both structured and unstructured). In addition, the software engineer performs a verification process to check that data location and characteristics are valid for the next level. Data can be generated offline, through the traditional ways of entering data (such as open data sources and relational databases in enterprise resource planning, customer relationship management systems, and other management information systems). In addition, data can also be obtained online through social media (such as LinkedIn, Facebook, Google+, and Twitter).

Acquisition. Here, filters and pat-

terns are applied by software engineers to ensure only valuable data is collected. Traditional data sources are easier to link to because they consist of structured data. But social software poses a greater technological challenge, as it contains human information that is complex, unstructured, ubiquitous, multi-format, and multi-channel.

Enhancement. The main objectives here are to endow the collected data with value, identify and extract information, and discover otherwise unknown relationships and patterns. To add such endowment, various ad-

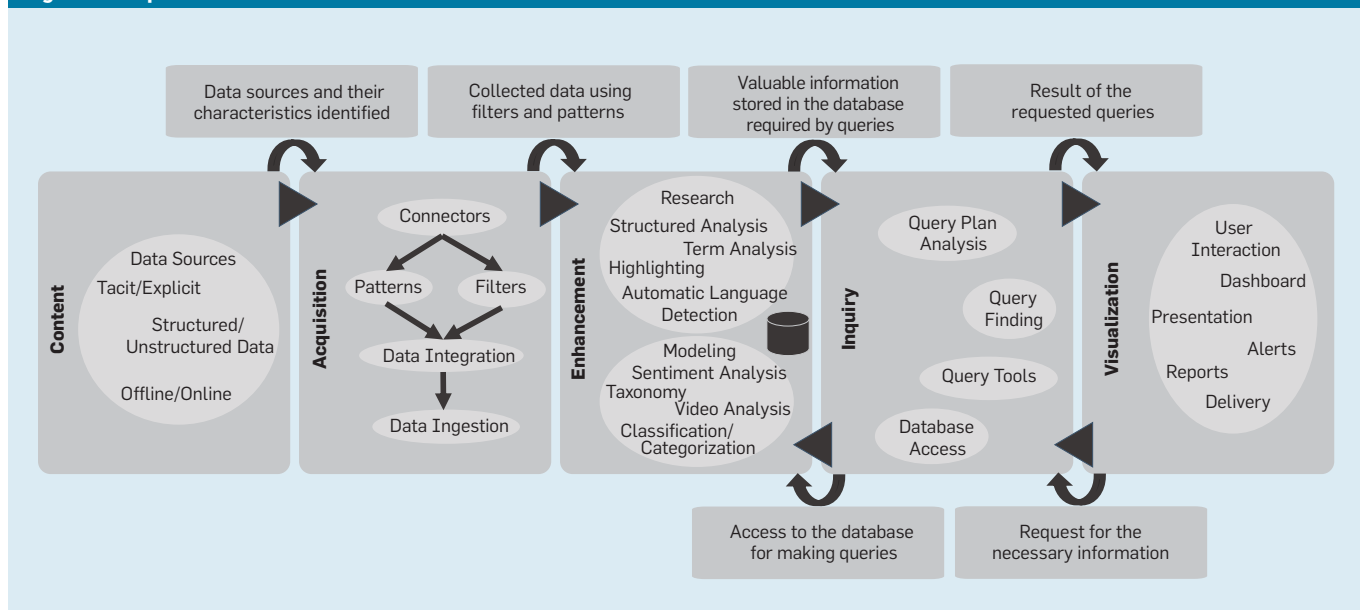
vanced data-analysis techniques are applied, perhaps divided into two main groups: research and modeling. Valuable information is obtained as a result of applying these techniques to the collected data. Metadata is also generated, reducing the complexity and processing of queries or operations that must be performed while endowing the data with meaning. Data and metadata are stored in a database for future queries, processing, generation of new metadata, and/or training and validation of the models.

Inquiry. Here, the system can ac-

Figure 1. BD-IRIS framework dimensions.



Figure 2. Proposed data architecture levels.



cess the data and metadata stored in the system database generated at the enhancement level. The main mode of access is through queries, usually based on the Structured Query Language, that extract the required information as needed.

Visualization. This level addresses presentation and visualization of the results, as well as interpretation of the meaning of the discovered information. Due to the nature of big data and the large amount of data to be processed, clarity and precision are important in the presentation and visualization of the results.

Organizational dimension. This

dimension is related to the characteristics and needs of the organization to provide data and processing and making use of it. It is also related to all the decisions the organization has to make to adapt the system to its needs.

On the one hand, the organization's strategy must be analyzed, since big data projects must align with the organization's business strategy. If not aligned, the results obtained may not be as valuable as they could be for the organization's decision making. To achieve such alignment, the organization must determine the objectives the project is intended to achieve, as well as the organizational challenges involved and the

project's target users, including customers, suppliers, and employees. It is also necessary to define the overall corporate transformation it is willing to make and the new business roles required to exploit big data technology. For example, a big data project could aim to use the knowledge extracted from customer data, products, and operations through the organization's processes to change its business model and create value, optimize business management, and identify new business opportunities. These projects are thus potentially able to increase customer acquisition and satisfaction, as well as increase loyalty and reduce the rate of customer abandonment. They can also improve business efficiency by, say, eliminating overproduction and reducing the launch time of new products or services. In addition, they can help negotiate better prices with suppliers and improve customer service. The project will thus be defined by the organization's business strategy. On the other hand, the resources offered and the knowledge acquired through big data technology allows optimization of existing business processes by improving them as much as possible.

To integrate enterprise strategy, business process, and human resources, the BD-IRIS framework uses the ARDIN (the Spanish acronym for Reference Architecture for INtegrated Development) enterprise reference architecture, allowing project managers to redefine the conceptual aspects of the enterprise (such as mission, vision, strategy, policies, and enterprise values), redesign and implement the new business process map, and reorganize and manage human resources considering in light of the new information and communication technologies—big data in this case—to improve them.⁶

In addition, models of the business processes must be developed so weak points and areas in need of improvement are detected. BD-IRIS uses several modeling languages:

*I**. *I** makes it possible for project engineers to gain a better understanding of organizational environments and business processes, understand the motivations, intentions, goals, and rationales of organizational management, and illustrate the various characteristics seen in the early phases of requirement specification.³⁰

Criteria for selecting appropriate tools.

- What is the price?
- Is it a new product and/or company or well established?
- Is it an open source or commercial tool?
- If commercial, is a trial version available?
- If commercial, is licensing per seat or per core?
- Is it platform independent?
- What is the implementation time?
- What is the implementation cost?
- Does it work in the cloud and use MapReduce and NoSQL features?
- Can real-time features be used or integrated into a real-time system?
- How easy is it to upgrade?
- How scalable is it?
- Can it work in batch and/or programmable mode?
- How easy is it to use? Is a GUI available?
- What learning curve should be expected?
- How compatible is it with other products?
- Does it work with big data?
- Does it offer an API?
- Can it integrate with geospatial data (such as GIS)?
- Does it provide modern techniques for data analysis?
- Can it handle missing data and data cleaning?
- Will it be possible to incorporate new techniques (such as add-ons or modules) different from those already implemented, as user needs evolve?
- What is the speed of computations? Does it use memory efficiently?
- Does it support programming languages (such as C++, Python, Java, and R) rather than just some internal ad hoc language?
- Is it able to fetch data from the Internet or from databases (such as SQL-supported)?
- Does it require connectors for databases? If yes, what do they cost?
- Does it support the SQL language?
- Are visualization capabilities available?
- Does it offer a Web or mobile client?
- Is good technical support, training, and documentation available?
- Is benchmarking available?

Business Process Model and Notation (BPMN). BPMN,²⁰ designed to model an overall map of an enterprise's business processes, includes 11 graphical, or modeling, elements classified into four categories: core elements (the BPD core element set), flow objects, connecting objects, and "swimlanes" and artifacts. BPMN 2.0 extends BPMN.

Unified Modeling Language. UML2.0¹⁹ is also used to model interactions among users and the technological platform in greater detail without ambiguity.


In selecting these modeling languages, we took into account that they are intuitive, well-known by academics and practitioners alike, useful for process modeling and information-system modeling, and proven in real-world enterprise-scale settings.

Support-tools dimension. This dimension consists of information-technology tools that support all dimensions in the framework, facilitating execution of the tasks to be performed in each dimension. Each such task can be supported by tools with certain characteristics; for example, some tools support only certain tasks, and some tasks can be carried out with and without the help of tools.


The tools that can be used in each dimension, except for data architecture, are standard tools that can be used in any software-engineering project. Types of tools include business management, office, case, project management, indicator management, software testing, and quality management. The data architecture dimension requires specific tools for each of its levels; see Table 3 in the online appendix for examples of tools that can be used at each level in the data architecture dimension.

Several tools are able to perform the same tasks, and the choice of appropriate tool for each project depends on the scenario in which it is used. The table here lists criteria to help prompt the questions that project engineers must address when choosing the appropriate tools for the particular needs of each project.

Data sources dimension. Considering that the foundation of big data ecosystems is data, it is essential that such data is reliable and provides value. This dimension refers to the sources of the data processed in big data ecosystems.



Considering that the foundation of big data ecosystems is data, it is essential that such data is reliable and provides value.



Big data technology is able to process both structured data (such as from relational databases, ERPs, CRMs, and open data), as well as data from semi-structured and unstructured data (such as from log files, machine-generated data, social media, transaction records, sensor data, and GPS signals). Objectives depend on the data that is available to the organization. To ensure optimal performance, the organization must define what data is of interest, identify its sources and formats, and perform, as needed, the pre-processing of raw data. Data is transformed into a format that is more readily "processable" by the system. Methods for preprocessing raw data include feature extraction (selecting the most significant specific data for certain contexts), transformation (modifying it to fit a particular type of input), sampling (selecting a representative subset from a large dataset), normalization (organizing it with the aim of allowing more efficient access to it), and "de-noising" (eliminating existing noise in it). Once such operations are performed, data is available to the system for processing.

Data-quality dimension. The aim here is to ensure quality in the acquisition, transformation, manipulation, and analysis of data, as well as in the validity of the results. Quality is the consequence of multiple factors, including complexity (lack of simplicity and uniformity in the data), usability (how readily data can be processed and integrated with existing standards and systems), time (timelines and frequency of data), accuracy (degree of accuracy describing the measured phenomenon), coherence (how the data meets standard conventions and is internally consistent, over time, with other data sources), linkability (how readily the data can be linked or joined with other data), validity (the data reflects what it is supposed to measure), accessibility (ease of access to information), clarity (availability of clear and unambiguous descriptions, together with the data), and relevance (the degree of fidelity of the results with regard to user needs, in terms of measured concepts and represented populations).²⁹

The United Nations Economic Commission for Europe²⁹ has identified the actions software engineers should perform to ensure quality in data input and

output results, thereby minimizing the risk in each of the various factors; see Table 4 in the online appendix.

Privacy/security dimension. Big data ecosystems usually deal with sensitive data, and the knowledge obtained from the data that may be scattered and lacking in value by itself. Due to such scattering, the customers and users who generate the data are often unaware of its value, disclosing it without reflection or compensation. Meanwhile, lack of awareness can lead to unexpected situations where the generated information is personally identifiable and metadata is more important than the data itself. Moreover, big data involves the real-time collection, storage, processing, and analysis of large amounts of data in different formats. Organizations that want to use big data must consider the risks, as well as their legal and ethical obligations, when processing and circulating it.


This dimension considers the privacy and security aspects of data management and communications, included in the same dimension because they are strongly related to each other, as explained in the online appendix.

BD-IRIS Framework Validation


Once the framework is developed, the next task is to validate and improve it, a process consisting of two phases: expert assessment and case studies. The aims are to validate the framework by verifying and confirming its usefulness, accuracy, and quality and improve the framework with the feedback obtained from the organizations involved and the conclusions drawn from the case studies. In such a case study, the framework is applied to a single organization. For example, we applied it to a Spanish “small and medium-size enterprise” from the metal fabrication market with 250 employees, using it to guide development and implementation of a social CRM system supported by a big data ecosystem.²¹ In another case study, we applied it to the Spanish division of a large oil and gas company, using it to guide development and implementation of a knowledge management system 2.0 as supported by a big data ecosystem;²² see the online appendix for results.

Discussion

Big data helps companies increase their competitiveness by improving their



Although proper integration of big data in a company is recognized as a key success factor in all big data projects, only two existing frameworks provide any guidance about the need to consider corporate management implications.



business models or their business processes. Big data has emerged over the past five years in companies, forcing them to deal with multiple business, management, technological, processing, and human resources challenges. Seven big data frameworks have been proposed in the IT literature, as outlined here, to deal with them in a satisfactory way. A framework can be defined as a structure consisting of several dimensions that are fitted and joined together to support or enclose something, in this case development and implementation of a big data ecosystem.

Big data frameworks also have weakness. First, none includes a methodology, understood as a documented approach for performing all activities in a big data project life cycle in a coherent, consistent, accountable, repeatable manner. This lack of a methodology is a big handicap because big data is still a novel area, and only a limited supply of well-trained professionals know what steps to take, in what order to take them, and how they should be performed.¹³ It is thus difficult for IT professionals, even those well trained in big data projects, to successfully take on a project employing the existing frameworks. In addition, in large-scale big data projects employing multiple teams of people, decisions regarding procedures, technologies, methods, and techniques can produce a lack of consistency and poor monitoring procedures. Second, each of the six dimensions of the big data framework—data architecture, organization, sources, quality, support tools, and privacy/security—addresses a different aspect of a project. However, although existing frameworks consider several dimensions, none of the seven frameworks proposed in the IT literature considers all six dimensions. Using only one of these frameworks means some important questions are ignored. Third, the approaches in each dimension are not fitted and joined together and are sometimes too vague and general or do not cover all the activities of the whole project life cycle. For example, although proper integration of big data in a company is recognized as a key success factor in all big data projects,³ only two existing frameworks provide any guidance about the need to consider corporate management implications. Neither do they explain when and

how to improve business strategy or when and how to carry out reengineering of a business process using big data. As a result, opportunities for improving business performance can be lost.


For this reason, the BD-IRIS framework needs to be structured in all seven dimensions. The main innovation is the BD-IRIS methodology dimension, along with the fact that it takes into account all the dimensions a big data framework should have within a single framework. The BD-IRIS methodology represents a guide to producing a big data ecosystem according to a process, covering the big data project life cycle and identifying when and how to use the approaches proposed in the other six dimensions. The utility of the framework and its completeness, level of detail, and accuracy of the relations among the methodology tasks and the approaches to other dimensions were validated in 2016 by five expert professionals from a Spanish consulting company with experience in big data projects, and by managers of the two organizations (not experts in big data projects) participating in our case studies. Lack of validation is a notable weakness of the existing frameworks.

Conclusion

This article has explored a framework for guiding development and implementation of big data ecosystems. We developed its initial design from the existing literature while providing additional knowledge. We then debugged, refined, improved, and validated this initial design through two methods—expert assessment and case studies—in a Spanish metal fabrication company and the Spanish division of an international oil and gas company. The results show the framework is considered valuable by corporate management where the case studies were applied.

The framework is useful for guiding organizations that wish to implement a big data ecosystem, as it includes a methodology that indicates in a clear and detailed way each activity and task that should be carried out in each of its phases. It also offers a comprehensive understanding of the system. Moreover, it provides control over a project and its scope, consequences, opportunities, and needs.

Although the framework has been validated through two different methods—expert evaluation and case studies—it also involves some notable limitations. For example, the methods we used for the analysis and validation in the two case studies are qualitative and not as precise as quantitative ones and based on the perceptions of the people involved in the application of the framework in the case studies and the consultants who evaluated it. Moreover, the evaluation experts were chosen from the same consulting company to avoid potential bias. Finally, we applied the framework in two companies in two different industrial sectors but have not yet tested its validity in other types of organization.

Regarding the scope of future work, we are exploring four areas: apply and assess the framework in companies from different industrial sectors; evaluate the ethical implications of big data systems; refine techniques for converting different input data formats into a common format to optimize the processing and analysis of data in big data systems; and finally, refine the automatic identification of people in different social networks, allowing companies to gather information entered by the same person in a given social network. 

References

- Adams, M.N. Perspectives on data mining. *International Journal of Market Research* 52, 1 (Jan. 2010), 11–19.
- Ahern, M., Clouse, A., and Turner, R. *CMMI Distilled: A Practical Introduction to Integrated Process Improvement, Second Edition*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2003.
- Alfouzan, H.I. Big data in business. *International Journal of Scientific & Engineering Research* 6, 5 (May 2015), 1351–1352.
- Bharadwaj, A., El Sawy, O.A., Pavlou, P.A., and Venkatraman, N. Digital business strategy: Toward a next generation of insights. *MIS Quarterly* 37, 2 (June 2013), 471–482.
- Brown, B., Chui, M., and Manyika, J. Are you ready for the era of 'big data'? *McKinsey Quarterly* 4 (Oct. 2011), 24–35.
- Chalmeta, R., Campos, C., and Grangel, R. Reference architectures for enterprise integration. *Journal of Systems and Software* 57, 3 (July 2001), 175–191.
- Chui, M., Manyika, J., and Bughin, J. Big data's potential for businesses. *Financial Times* (May 13, 2011); <https://www.ft.com/content/64095dba-7cd5-11e0-994d-00144feabdc0>
- Das, T.K. and Kumar, P.M. Big data analytics: A framework for unstructured data analysis. *International Journal of Engineering and Technology* 5, 1 (Feb.–Mar. 2013), 153–156.
- Debartoli, S., Müller, O., and Vom Brocke, J. Comparing business intelligence and big data skills: A text mining study using job advertisements. *Business & Information Systems Engineering* 6, 5 (Oct. 2014), 289–300.
- Demchenko, Y., de Laat, C., and Membrey, P. Defining architecture components of the big data ecosystem. In *Proceedings of the International Conference on Collaboration Technologies and Systems* (Minneapolis, MN, May 19–23). IEEE Press, 2014, 104–112.
- Elgendy, N. and Elragal, A. Big data analytics: A literature review. In *Proceedings of the 14th Industrial Conference on Data Mining* (St. Petersburg, Russia,

- July 16–20). Lecture Notes in Computer Science, 8557. Springer International Publishing, Switzerland, 2014, 214–227.
- Ferguson, M. *Architecting a Big Data Platform for Analytics*. IBM White Paper, Oct. 2012; <http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=IML14333USEN>
- Flouris, I., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Kamp, M., and Mock, M. Issues in complex event processing: Status and prospects in the big data era. *Journal of Systems and Software* 127 (May 2017), 217–236.
- Gëczy, P. Big data management: Relational framework. *Review of Business & Finance Studies* 6, 3 (2015), 21–30.
- Halper, F. and Krishnan, K. *TDWI Big Data Maturity Model Guide*. TDWI Research, Renton, WA, 2013; <https://tdwi.org/whitepapers/2013/10/tdwi-big-data-maturity-model-guide.aspx>
- Hortonworks. *Hortonworks Big Data Maturity Model*, 2016; <http://hortonworks.com/wp-content/uploads/2016/04/Hortonworks-Big-Data-Maturity-Assessment.pdf>
- Jagadish, H.V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J.M., Ramakrishnan, R., and Shahabi, C. Big data and its technical challenges. *Commun. ACM* 57, 7 (July 2014), 86–94.
- Miller, H.G. and Mork, P. From data to decisions: A value chain for big data. *IT Professional* 15, 1 (Jan.–Feb. 2013), 57–59.
- Object Management Group. *Unified Modeling Language*. OMG, 2000; <http://www.uml.org/>
- Object Management Group. *Business Process Model and Notation*. OMG, 2011; <http://www.omg.org/spec/BPMN/2.0>
- Orenga-Roglá, S. and Chalmeta, R. Social customer relationship management: Taking advantage of Web 2.0 and big data technologies. *SpringerPlus* 5, 1462 (Aug. 2016), 1–17.
- Orenga-Roglá, S. and Chalmeta, R. Methodology for the implementation of knowledge management systems 2.0: A case study in an oil and gas company. *Business & Information Systems Engineering* (Dec. 2017), 1–19; <https://doi.org/10.1007/s12599-017-0513-1>
- Pawlowski, J. and Bick, M. The global knowledge management framework: Towards a theory for knowledge management in globally distributed settings. *Electronic Journal of Knowledge Management* 10, 1 (Jan. 2012), 92–108.
- Radcliffe, J. *Leverage a Big Data Maturity Model to Build Your Big Data Roadmap*. Radcliffe Advisory Services, Ltd., Guildford, U.K., 2014.
- Sagiroglu, S. and Sinanc, D. Big data: A review. In *Proceedings of the International Conference on Collaboration Technologies and Systems* (San Diego, CA, May 20–24). IEEE Press, 2013, 42–47.
- Shin, D.H. and Choi, M.J. Ecological views of big data: Perspectives and issues. *Telematics and Informatics* 32, 2 (May 2015), 311–320.
- Sun, H. and Heller, P. *Oracle Information Architecture: An Architect's Guide to Big Data*. Oracle White Paper, Aug. 2012; https://d2jt48ltdp5cjc.cloudfront.net/uploads/test1_3021.pdf
- Tekiner, F. and Keane, J.A. Big data framework. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (Manchester, U.K., Oct. 13–16). IEEE Press, 2013, 1494–1499.
- United Nations Economic Commission for Europe. *A Suggested Framework for the Quality of Big Data*. Deliverables of the UNECE Big Data Quality Task Team. UNECE, Dec. 2014; <http://www.uncece.org/uncece/search?q=A+Suggested+Framework+for+the+Quality+of+Big+Data.+&op=Search>
- Yu, E. Why agent-oriented requirements engineering. In *Proceedings of the Third International Workshop on Requirements Engineering: Foundation of Software Quality* (Barcelona, Spain, June 16–17). Presses Universitaires de Namur, Namur, Belgium, 1997, 171–183.

Sergio Orenga-Roglá (sergio.orenga@uji.es) is a researcher in the Systems Integration and Re-Engineering (IRIS) research group at the Universitat Jaume I, Castellón, Spain.

Ricardo Chalmeta (rchalmet@uji.es) is an assistant professor in the Department of Computer Languages and Systems and Director of the Systems Integration and Re-Engineering (IRIS) research group at the Universitat Jaume I, Castellón, Spain.

DOI:10.1145/3198448

In its original form, the Church-Turing thesis concerned computation as Alan Turing and Alonzo Church used the term in 1936—human computation.

BY B. JACK COPELAND AND ORON SHAGRIR

The Church-Turing Thesis: Logical Limit or Breachable Barrier?

THE CHURCH-TURING THESIS (CTT) underlies tantalizing open questions concerning the fundamental place of computing in the physical universe. For example, is every physical system computable? Is the universe essentially computational in nature? What are the implications for computer science of recent speculation about physical uncomputability? Does CTT place a fundamental logical limit on what can be computed, a computational “barrier” that cannot be broken, no matter how far and in what multitude of ways computers develop? Or could new types of hardware, based perhaps on quantum or relativistic phenomena, lead to radically

new computing paradigms that do breach the Church-Turing barrier, in which the uncomputable becomes computable, in an *upgraded* sense of “computable”? Before addressing these questions, we first look back to the 1930s to consider how Alonzo Church and Alan Turing formulated, and sought to justify, their versions of CTT. With this necessary history under our belts, we then turn to today’s dramatically more powerful versions of CTT.

History of the Thesis

Turing stated what we will call “Turing’s thesis” in various places and with varying degrees of rigor. The following formulation is one of his most accessible.

Turing’s thesis. “L.C.M.s [logical computing machines, Turing’s expression for Turing machines] can do anything that could be described as ... ‘purely mechanical’.”³⁸

Turing also formulated his thesis in terms of numbers. For example, he said, “It is my contention that these operations [the operations of an L.C.M.] include all those which are used in the computation of a number.”³⁶ and “[T]he ‘computable numbers’ include all numbers which would naturally be regarded as computable.”³⁶

Church (who, like Turing, was working on the German mathematician David Hilbert’s *Entscheidungsproblem*) advanced “Church’s thesis,” which he expressed in terms of definability in his lambda calculus.

Church’s thesis. “We now define the notion ... of an effectively calculable

» key insights

- The term “Church-Turing thesis” is used today for numerous theses that diverge significantly from the one Alonzo Church and Alan Turing conceived in 1936.
- The range of algorithmic processes studied in modern computer science far transcends the range of processes a “human computer” could possibly carry out.
- There are at least three forms of the “physical Church-Turing thesis”—modest, bold, and super-bold—though, at the present stage of physical inquiry, it is unknown whether any of them is true.



Is everything in the physical universe computable? Hubble Space Telescope view of the Pillars of Creation in the Eagle Nebula.

function of positive integers by identifying it with the notion of a recursive function of positive integers (or of a λ -definable function of positive integers).³⁵

Church chose to call this a definition. American mathematician Emil Post, on the other hand, referred to Church's thesis as a "working hypothesis" and criticized Church for masking it in the guise of a definition.³³

Upon learning of Church's "defi-

niton," Turing quickly proved that λ -definability and his own concept of computability (over positive integers) are equivalent. Church's thesis and Turing's thesis are thus equivalent, if attention is restricted to functions of positive integers. (Turing's thesis, more general than Church's, also encompassed computable real numbers.) However, it is important for a computer scientist to appreciate that despite this extensional equivalence, Turing's thesis and

Church's thesis have distinct meanings and so are different theses, since they are not intensionally equivalent. A leading difference in their meanings is that Church's thesis contains no reference to computing machinery, whereas Turing's thesis is expressed in terms of the "Turing machine," as Church dubbed it in his 1937 review of Turing's paper.

It is now widely understood that Turing introduced his machines with the intention of providing an idealized

description of a certain human activity—numerical computation; in Turing’s day computation was carried out by rote workers called “computers,” or, sometimes, “computors”; see, for example, Turing.³⁷ The Church-Turing thesis is about computation as the term was used in 1936—human computation. Church’s term “effectively calculable function” was intended to refer to functions that are calculable by an idealized human computer; and, likewise, Turing’s phrase “numbers which would naturally be regarded as computable” was intended to refer to those numbers that could be churned out, digit by digit, by an idealized human computer working ceaselessly.

Here, then, is our formulation of the historical version of the Church-Turing thesis, as informed by Turing’s proof of the equivalence of his and Church’s theses:

CTT-Original (CTT-O). Every function that can be computed by the idealized human computer, which is to say, can be effectively computed, is Turing-computable.

Some mathematical logicians view CTT-O as subject ultimately to either mathematical proof or mathematical refutation, like open mathematical conjectures, as in the Riemann hypothesis, while others regard CTT-O as not amenable to mathematical proof but supported by philosophical arguments and an accumulation of mathematical evidence. Few logicians today follow Church in regarding CTT-O as a definition. We subscribe to Turing’s view of the status of CTT-O, as we outline later.

In computer science today, algorithms and effective procedures are, of course, associated not primarily with humans but with machines. (Note, while some expositors might distinguish between the terms “algorithm” and “effective procedure,” we use the terms interchangeably.) Many computer science textbooks formulate the Church-Turing thesis without mentioning human computers at all; examples include the well-known books by Hopcroft and Ullman²⁴ and Lewis and Papadimitriou.²⁹ This is despite the fact that the concept of human computation was at the heart of both Turing’s and Church’s analysis of computation.

We discuss several important modern forms of the Church-Turing thesis,

each going far beyond CTT-O. First, we look more closely at the algorithmic form of thesis, as stated to a first approximation by Lewis and Papadimitriou²⁹: “[W]e take the Turing machine to be a precise formal equivalent of the intuitive notion of ‘algorithm’.”

What Is an Algorithm?

The range of algorithmic processes studied in modern computer science far transcends the range of processes a Turing machine is able to carry out. The Turing machine is restricted to, say, changing at most one bounded part at each sequential step of a computation. As Yuri Gurevich pointed out, the concept of an algorithm keeps evolving: “We have now parallel, interactive, distributed, real-time, analog, hybrid, quantum, etc. algorithms.”²² There are enzymatic algorithms, bacterial foraging algorithms, slime-mold algorithms, and more. The Turing machine is incapable of performing the atomic steps of algorithms carried out by, say, an enzymatic system (such as selective enzyme binding) or a slime mold (such as pseudopod extension). The Turing machine is similarly unable to duplicate (as opposed to simulate) John Conway’s Game of Life, where—unlike a Turing machine—every cell updates simultaneously.

A thesis aiming to limit the scope of algorithmic computability to Turing computability should thus not state that every possible algorithmic process can be performed by a Turing machine. The way to express the thesis is to say the extensional input-output function $\iota\alpha$ associated with an algorithm α is always Turing-computable; $\iota\alpha$ is simply the extensional mapping of α ’s inputs to α ’s outputs. The algorithm the Turing machine uses to compute $\iota\alpha$ might be very different from α itself. A question then naturally arises: If an algorithmic process need not be one a Turing machine can carry out, save in the weak sense just mentioned, then where do the boundaries of this concept lie? What indeed is an algorithm?

The dominant view in computer science is that, ontologically speaking, algorithms are abstract entities; however, there is debate about what abstract entities algorithms are. Gurevich defined the concept in terms of abstract-state machines, Yiannis Moschovakis in terms of abstract recursion, and Noson

Yanofsky in terms of equivalence classes of programs, while Moshe Vardi has speculated that an algorithm is both abstract-state machine and recursor. It is also debated whether an algorithm must be physically implementable. Moschovakis and Vasilis Paschalis (among others) adopt a concept of algorithm “so wide as to admit ‘non-implementable’ algorithms,”³⁰ while other approaches do impose a requirement of physical implementability, even if only a very mild one. David Harel, for instance, writes: “[A]ny algorithmic problem for which we can find an algorithm that can be programmed in some programming language, any language, running on some computer, any computer, even one that has not been built yet but can be built ... is also solvable by a Turing machine. This statement is one version of the so-called Church/Turing thesis.”²³

Steering between these debates—and following Harel’s suggestion that the algorithms of interest to computer science are always expressible in programming languages—we arrive at the following program-oriented formulation of the algorithmic thesis:

CTT-Algorithm (CTT-A). Every algorithm can be expressed by means of a program in some (not necessarily currently existing) Turing-equivalent programming language.

There is an option to narrow CTT-A by adding “physically implementable” before “program,” although in our view this would be to lump together two distinct computational issues that are better treated separately.

The evolving nature and open-endedness of the concept of an algorithm is matched by a corresponding open-endedness in the concept of a programming language. But this open-endedness notwithstanding, CTT-A requires that all algorithms be bounded by Turing computability.

Later in this article we examine complexity-theoretic and physical versions of the Church-Turing thesis but first turn to the question of the justification of the theses introduced so far. Are CTT-O and CTT-A correct?


What Justifies the Church-Turing Thesis?

Stephen Kleene—who coined the term “Church-Turing thesis”—catalogued four types of argument for CTT-O: First,


the argument from non-refutation points out the thesis has never been refuted, despite sustained (and ongoing) attempts to find a counterexample (such as the attempts by László Kalmár and, more recently, by Doukas Kapan-tais). Second, the argument from confluence points to the fact that the various characterizations of computability, while differing in their approaches and formal details, turn out to encompass the very same class of computable functions. Four such characterizations were presented (independently) in 1936 and immediately proved to be *extensionally* equivalent: Turing computability, Church's λ -definability, Kleene's recursive functions, and Post's finitary combinatory processes.

Third is an argument usually referred to nowadays as "Turing's analysis." Turing called it simply argument "I," stating five very general and intuitive constraints—or axioms—the human computer may be assumed to satisfy: "The behavior of the computer at any moment is determined by the symbols which he is observing, and his 'state of mind' at that moment"; "[T] here is a bound B to the number of symbols or squares which the computer can observe at one moment"; "[E]ach of the new observed squares is within L squares of an immediately previously observed square"; "[I]n a simple operation not more than one symbol is altered"; and "[T]he number of states of mind which need be taken into account is finite." Turing noted that reference to the computer's states of mind can be avoided by talking instead about configurations of symbols, these being "a more definite and physical counterpart" of states of mind.³⁶

The second part of Turing's argument I is a demonstration that each function computed by any human computer subject to these constraints is also computable by a Turing machine; it is not difficult to see that each of the computer's steps can be mimicked by the Turing machine, either in a single step or by means of a series of steps. In short, Turing's five axioms entail CTT-O. (Turing's axiomatic approach to computability was in fact foreshadowed by Kurt Gödel in a suggestion to Church a year or so earlier.¹⁵ Some more recent axiomatic approaches to computability proceed differently; for example, Erwin Engeler



The Turing machine is restricted to, say, changing at most one bounded part at each sequential step of a computation.



employs the Schönfinkel-Curry idea of "combinators" in order to axiomatize the concept of an algorithmic function.)

Fourth in this catalog of considerations supporting CTT-O are arguments from first-order logic. They are typified by a 1936 argument of Church's and by Turing's argument II, from Section 9 of Turing's 1936 paper. In 2013, Saul Kripke²⁸ presented a reconstruction of Turing's argument II, which goes as follows: Computation is a special form of mathematical deduction; and every mathematical deduction—and therefore every computation—can be formalized as a valid deduction in the language of first-order predicate logic with identity (a step Kripke referred to as "Hilbert's thesis"); following Gödel's completeness theorem, each computation is thus formalized by a provable formula of first-order logic; and every computation can therefore be carried out by the universal Turing machine. This last step regarding the universal Turing machine is secured by a theorem proved by Turing: Every provable formula of first-order logic can be proved by the universal Turing machine.

The third and fourth of these arguments provide justification for CTT-O but not for CTT-A. As Robin Gandy²⁰ pointed out, the third argument—Turing's I—contains "crucial steps ... where he [Turing] appeals to the fact that the calculation is being carried out by a human being."²⁰ For example, Turing assumed "a human being can only write one symbol at a time," and Gandy noted this assumption cannot be carried over to a parallel machine that "prints an arbitrary number of symbols simultaneously."²⁰ In Conway's Game of Life, for instance, there is no upper bound on the number of cells that make up the grid, yet the symbols in all the cells are updated simultaneously. Likewise, the fourth argument (Turing's II) involves the claim that computation is a special form of formal proof, but the notion of proof is intrinsically related to what a human mathematician—and not some oracle—can prove.

It is thus perhaps not too surprising that the third and fourth arguments in this catalog seldom if ever appear in logic and computer science textbooks. The two arguments that are always given for the Church-Turing thesis (in, for example, Lewis and Papadimitriou²⁹) are

confluence and non-refutation. Yet both those arguments are merely inductive, whereas the third and fourth arguments are deductive in nature.


However, a number of attempts have sought to extend Turing's axiomatic analysis to machine computation; for example, Gandy²⁰ broadened Turing's analysis in such a way that parallel computation is included, while Dershowitz and Gurevich¹⁶ gave a more general analysis in terms of abstract state machines. We return to the topic of extending the analysis to machine computation later in this article but first address the important question of whether CTT-O is mathematically provable.

Is the Thesis Mathematically Provable?


It used to be thought by mathematical logicians and others that CTT-O is not amenable to formal proof, since it is not a mathematically precise statement. This is because it pairs an informal concept—a “vague intuitive notion,” Church called it⁵—with a precise concept. However, Elliott Mendelson gave a powerful critique of this general argument; and today the view that CTT-O is formally provable seems to be gaining acceptance; see, for example, Dershowitz and Gurevich.¹⁶ Inspired by Gandy,²⁰ Wilfried Sieg³⁵ stated that a tightened form of Turing's argument I proves the thesis; and Kripke²⁸ entertained the same claim for Turing's argument II.

Turing's own view was that, on the contrary, his thesis is not susceptible to mathematical proof. He thought his arguments I and II, and indeed “[a]ll arguments which can be given” for the thesis, are “fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically.”³⁶ Hilbert's thesis is another example of a proposition that can be justified only by appeal to intuition, and so Kripke's²⁸ tightened form of argument II, far from proving CTT-O, merely deduced it from another thesis that is also not amenable to mathematical proof.

Much the same can be said about argument I. If axioms 1–5 are formulated in precise mathematical terms, then it is certainly provable from them that computation is bounded by Turing computability; this is probably what Gandy²⁰ meant when he said Turing's argument I proves a “theorem.” But the real issue



Turing's own view was that, on the contrary, his thesis is not susceptible to mathematical proof.



is whether these axioms completely capture the concept of a computational or algorithmic process, and, so far as we see, no one has ever given a rigorous mathematical justification of that claim. The axioms may be supported by informal arguments, but the whole edifice then falls short of mathematical proof. This is most apparent when the informal arguments offered for the axioms invoke limitations in the cognitive capacities of human computers, as we point out elsewhere.¹³ A justification of the second axiom may, for instance, refer to the limitations of human observation. The axioms most certainly lie beyond the scope of mathematical demonstration if their truth depends on contingent human limitations. Turing himself cheerfully appealed to cognitive limitations in the course of his analysis, saying, for example, “[J]ustification lies in the fact that the human memory is necessarily limited.”³⁶

In summary, our answer to “Is CTT-O mathematically provable?” is: Turing thought not and we have found no reason to disagree with him. The various historical arguments seem more than sufficient to establish CTT-O, but these arguments do indeed fall short of mathematical proof.

We next address complexity theoretic forms of the Church-Turing thesis, then turn to the question of whether CTT-A is justified in the context of physically realistic computations.

Complexity: The Extended Church-Turing Thesis

It is striking that the Turing machine holds a central place not only in computability theory but also in complexity theory, where it is viewed as a universal model for complexity classes.

In complexity theory, the time complexities of any two general and reasonable models of computation are assumed to be polynomially related. But what counts as “reasonable”? Aharonov and Vazirani¹ gloss over “reasonable” as “physically realizable in principle”; see also Bernstein and Vazirani.³ If a computational problem's time complexity is t in some (general and reasonable) model, then its time complexity is assumed to be $\text{poly}(t)$ in the single-tape Turing machine model; see also Goldreich.²¹ This assumption has different names in the literature; Goldreich²¹ called it the

Cobham-Edmonds thesis, while Yao⁴⁰ introduced the term “Extended Church-Turing thesis.” The thesis is of interest only if $P \neq NP$, since otherwise it is trivial.

Quantum-computation researchers also use a variant of this thesis, as expressed in terms of probabilistic Turing machines. Bernstein and Vazirani³ said: “[C]omputational complexity theory rests upon a modern strengthening of [the Church-Turing] thesis, which asserts that any ‘reasonable’ model of computation can be efficiently simulated on a probabilistic Turing machine.”³

Aharonov and Vazirani¹ give the following formulation of this assumption, naming it the “Extended Church-Turing thesis”—though it is not quite the same as Yao’s earlier thesis of the same name, which did not refer to probabilistic Turing machines:

CTT-Extended (CTT-E). “[A]ny reasonable computational model can be simulated efficiently by the standard model of classical computation, namely, a probabilistic Turing machine.”¹

As is well known in computer science, Peter Shor’s quantum algorithm for prime factorization is a potential counterexample to CTT-E; the algorithm runs on a quantum computer in polynomial time and is much faster than the most-efficient known “classical” algorithm for the task. But the counterexample is controversial. Some computer scientists think the quantum computer invoked is not a physically reasonable model of computation, while others think accommodating these results might require further modifications to complexity theory.

We turn now to extensions of the Church-Turing thesis into physics.

Physical Computability

The issue of whether every aspect of the physical world is Turing-computable was broached by several authors in the 1960s and 1970s, and the topic rose to prominence in the mid-1980s.

In 1985, Stephen Wolfram formulated a thesis he described as “a physical form of the Church-Turing hypothesis,” saying, “[U]niversal computers are as powerful in their computational capacities as any physically realizable system can be, so that they can simulate any physical system.”³⁹ In the same year, David Deutsch, who laid the foundations of quantum computation, independently

stated a similar thesis, describing it as “the physical version of the Church-Turing principle.”¹⁷ The thesis is now known as the Church-Turing-Deutsch thesis and the Church-Turing-Deutsch-Wolfram thesis.

Church-Turing-Deutsch-Wolfram thesis (CTDW). Every finite physical system can be simulated to any specified degree of accuracy by a universal Turing machine.

Deutsch pointed out that if “simulated” is understood as “perfectly simulated,” then the thesis is falsified by continuous classical systems, since such classical systems necessarily involve uncomputable real numbers, and went on to introduce the concept of a universal quantum computer, saying such a computer is “capable of perfectly simulating every finite, realizable physical system.” Other physical formulations were advanced by Lenore Blum et al., John Earman, Itamar Pitowsky, Marian Pour-El, and Ian Richards, among others.

We next formulate a strong version of the physical Church-Turing thesis we call the “total physical computability thesis.” (We consider some weaker versions later in the article.) By “physical system” we mean any system whose behavior is in accordance with the actual laws of physics, including non-actual and idealized systems.

Total physical computability thesis (CTT-P). Every physical aspect of the behavior of any physical system can be calculated (to any specified degree of accuracy) by a universal Turing machine.

As with CTT-E, there is also a probabilistic version of CTT-P, formulated in terms of a probabilistic Turing machine.

Arguably, the phrase “physical version of the Church-Turing thesis” is an inappropriate name for this and related theses, since CTT-O concerns a form of effective or algorithmic activity and asserts the activity is always bounded by Turing computability, while CTT-P and CTDW, on the other hand, entail that the activity of every physical system is bounded by Turing computability; the system’s activity need not be algorithmic/effective at all. Nevertheless, in our “CTT-” nomenclature, we follow the Deutsch-Wolfram tradition throughout this article.

Is CTT-P true? Not if physical systems include systems capable of producing unboundedly many digits of a random

binary sequence; Church showed such sequences are uncomputable, as we discussed elsewhere.⁸ Moreover, speculation that there may be deterministic physical processes whose behavior cannot be calculated by the universal Turing machine stretches back over several decades; for a review, see Copeland.⁹ In 1981, Pour-El and Richards³⁴ showed that a system evolving from computable initial conditions in accordance with the familiar three-dimensional wave equation is capable of exhibiting behavior that falsifies CTT-P; even today, however, it is an open question whether these initial conditions are physically possible. Earlier papers, from the 1960s, by Bruno Scarpellini, Arthur Komar, and Georg Kreisel, in effect questioned CTT-P, with Kreisel stating: “There is no evidence that even present-day quantum theory is a mechanistic, i.e., recursive theory in the sense that a recursively described system has recursive behavior.”²⁷ Other potential counterexamples to CTT-P have been described by a number of authors, including what are called “relativistic” machines. First introduced by Pitowsky,³² they will be examined in the section called “Relativistic Computation.”

CTT-P and Quantum Mechanics

There are a number of theoretical countermodels to CTT-P arising from quantum mechanics. For example, in 1964, Komar²⁶ raised “the issue of the macroscopic distinguishability of quantum states,” asserting there is no effective procedure “for determining whether two arbitrarily given physical states can be superposed to show interference effects.” In 2012, Eisert et al.¹⁹ showed “[T]he very natural physical problem of determining whether certain outcome sequences cannot occur in repeated quantum measurements is undecidable, even though the same problem for classical measurements is readily decidable.” This is an example of a problem that refers unboundedly to the future but not to any specific time. Other typical physical problems take the same form; Pitowsky gave as examples “Is the solar system stable?” and “Is the motion of a given system, in a known initial state, periodic?”

Cubitt et al.¹⁴ described another such undecidability result in a 2015 *Nature* article, outlining their proof that “[T]he

spectral gap problem is algorithmically undecidable: There cannot exist any algorithm that, given a description of the local interactions, determines whether the resultant model is gapped or gapless.” Cubitt et al. also said this is the “first undecidability result for a major physics problem that people would really try to solve.”

The spectral gap, an important determinant of a material’s properties, refers to the energy spectrum immediately above the ground-energy level of a quantum many-body system, assuming a well-defined least-energy level of the system exists; the system is said to be “gapless” if this spectrum is continuous and “gapped” if there is a well-defined next-least energy level. The spectral gap problem for a quantum many-body system is the problem of determining whether the system is gapped or gapless, given the finite matrices (at most three) describing the local interactions of the system.

In their proof, Cubitt et al.¹⁴ encoded the halting problem in the spectral gap problem, showing the latter is at least as hard as the former. The proof involves an infinite family of two-dimensional lattices of atoms. But they pointed out their result also applies to finite systems whose size increases, saying, “Not only can the lattice size at which the system switches from gapless to gapped be arbitrarily large, the threshold at which this transition occurs is uncomputable.” Their proof offers an interesting countermodel to CTT-P, involving a physically relevant example of a finite system of increasing size. There exists no effective method for extrapolating the system’s future behavior from (complete descriptions of) its current and past states.

It is debatable whether any of these quantum models correspond to real-world quantum systems. Cubitt et al.¹⁴

admitted the model invoked in their proof is highly artificial, saying, “Whether the results can be extended to more natural models is yet to be determined.” There is also the question of whether the spectral gap problem becomes computable when only local Hilbert spaces of realistically low dimensionality are considered. Nevertheless, these results are certainly suggestive: CTT-P cannot be taken for granted, even in a finite quantum universe.

Summarizing the current situation with respect to CTT-P, we can say, although theoretical countermodels in which CTT-P is false have been described, there is at present—so far as we know—not a shred of evidence that CTT-P is false in the actual universe. Yet it would seem most premature to assert that CTT-P is true.

Weaker Physical Computability Theses

Piccinini³¹ has distinguished between two different types of physical versions of the Church-Turing thesis, both commonly found in the literature, describing them as “bold” and “modest” versions of the thesis, respectively. The bold and modest versions are weaker than our “super-bold” version just discussed (CTT-P). Bold versions of the thesis state, roughly, that “Any physical process can be simulated by some Turing machine.”³¹ The Church-Turing-Deutsch-Wolfram thesis (CTDW) is an example, though Piccinini emphasized that the bold versions proposed by different researchers are often “logically independent of one another” and that, unlike the different formulations of CTT-O, which exhibit confluence, the different bold formulations in fact exhibit “lack of confluence.”³¹

CTDW and other bold forms are too

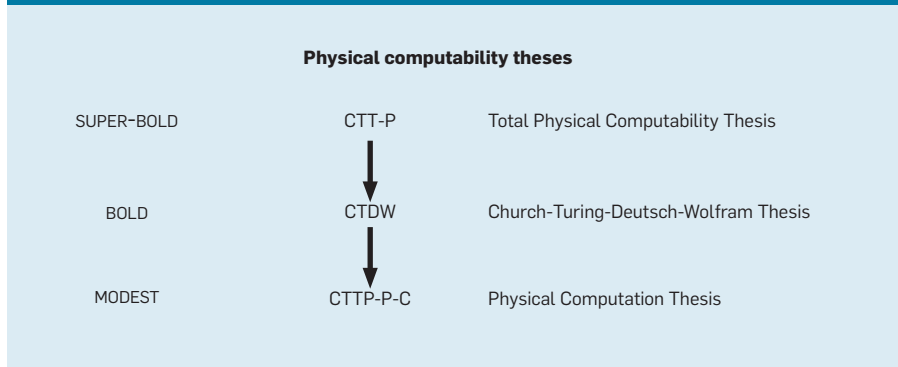
weak to rule out the uncomputability scenarios described by Cubitt et al.¹⁴ and by Eisert et al.¹⁹ This is because the physical processes involved in these scenarios may, so far as we know, be Turing-computable; it is possible that each process can be simulated by a Turing machine, to any required degree of accuracy, and yet the answers to certain physical questions about the processes are, in general, uncomputable. The situation is similar in the case of the universal Turing machine itself. The machine’s behavior (consisting of the physical actions of the read/write head) is always Turing-computable since it is produced by the Turing machine’s program, yet the answers to some questions about the behavior (such as whether or not the machine halts given certain inputs) are not computable.

Nevertheless, bold forms (such as CTDW) are interesting empirical hypotheses in their own right and the world might confute them. For instance, CTDW fails in the wave-equation countermodel due to Pour-El and Richards³⁴ where the mapping between the wave equation’s “inputs” and “outputs” is not a Turing-computable (real) function; although, as noted earlier, the physicality of this countermodel can readily be challenged. We discuss some other potential countermodels later in the article, but turn first to what Piccinini termed “modest” versions of the thesis.

Modest versions maintain in essence that every physical computing process is Turing-computable; for two detailed formulations, see Gandy²⁰ and Copeland.⁸ Even if CTT-P and CTDW are in general false, the behavior of the subset of physical systems that are appropriately described as computing systems may nevertheless be bounded by Turing-computability. An illustration of the difference between modest versions on the one hand and CTT-P and CTDW on the other is given by the fact that the wave-equation example is not a countermodel to the modest thesis, assuming, as seems reasonable, that the physical dynamics described by the equation do not constitute a computing process.

Here, we formulate a modest version of the physical Church-Turing thesis we call the “Physical Computation” thesis, then turn to the question of whether it is true.

Relationships between the three physical computability theses: CTT-P, CTDW, and CTT-P-C.



Physical Computation Thesis

This form of the thesis maintains that physical computation is bounded by Turing-computability.

Physical computation thesis (CTT-P-C). Every function computed by any physical computing system is Turing-computable.

Is CTT-P-C true? As with the stronger physical computability theses, it seems too early to say. CTT-P-C could be false only if CTT-P and CTDW turn out to be false, since each of them entails CTT-P-C (see the figure here, which outlines the relationships among CTT-P, CTDW, and CTT-P-C). If all physical computation is effective in the 1930s sense of Turing and Church, then CTT-P-C is certainly true. If, however, the door is open to a broadened sense of computation, where physical computation is not necessarily effective in the sense of being bounded by Turing-computability, then CTT-P-C makes a substantive claim.

There is, in fact, heated debate among computer scientists and philosophers about what counts as physical computation. Moreover, a number of attempts have sought to describe a broadened sense of computation in which computation is not bounded by Turing-computability; see, for example, Copeland.⁶ Computing machines that compute “beyond the Turing limit” are known collectively as “hypercomputers,” a term introduced in Copeland and Proudfoot.¹¹ Some of the most thought-provoking examples of notional machines that compute in the broad sense are called “supertask” machines. These “Zeno computers” squeeze infinitely many computational steps into a finite span of time. Examples include accelerating machines,^{7,12} shrinking machines, and the intriguing relativistic computers described in the next section.

Notional machines all constitute rather theoretical countermodels to CTT-P-C, so long as it is agreed that they compute in a broadened sense, but none has been shown to be physically realistic, although, as we explain, relativistic computers come close. In short, the truth or falsity of CTT-P-C remains unsettled.

Relativistic Computation

Relativistic machines operate in space-time structures with the property that

the entire endless lifetime of one component of the machine is included in the finite chronological past of another component, called “the observer.” The first component could thus carry out an infinite computation (such as calculating every digit of π) in what is, from the observer’s point of view, a finite timespan of, say, one hour. (Such machines are in accord with Einstein’s general theory of relativity, hence the term “relativistic.”) Examples of relativistic computation have been detailed by Pitowsky, Mark Hogarth, and Istvan Németi.

In this section we outline a relativistic machine RM consisting of a pair of communicating Turing machines, T_E and T_O , in relative motion. T_E is a universal machine, and T_O is the observer. RM is able to compute the halting function, in a broad sense of computation. Speaking of computation here seems appropriate, since RM consists of nothing but two communicating Turing machines.

Here is how RM works. When the input (m,n) , asking whether the m^{th} Turing machine (in some enumeration of the Turing machines) halts or not when started on input n , enters T_O , T_O first prints 0 (meaning “never halts”) in its designated output cell and then transmits (m,n) to T_E . T_E simulates the computation performed by the m^{th} Turing machine when started on input n and sends a signal back to T_O if and only if the simulation terminates. If T_O receives a signal from T_E , T_O deletes the 0 it previously wrote in its output cell and writes 1 in its place (meaning “halts”). After one hour, T_O ’s output cell shows 1 if the m^{th} Turing machine halts on input n and shows 0 if the m^{th} machine does not halt on n .

The most physically realistic version of this setup to date is due to Németi and his collaborators in Budapest. T_E , an ordinary computer, remains on Earth, while the observer T_O travels toward and enters a slowly rotating Kerr black hole. T_O approaches the outer event horizon, a bubble-like hypersurface surrounding the black hole. Németi theorized that the closer T_O gets to the event horizon, the faster T_E ’s clock runs relative to T_O due to Einsteinian gravitational time dilation, and this speeding up continues with no upper limit. T_O motion proceeds until, relative to a time t on T_O clock, the entire span of T_E ’s computing is over. If any signal was emitted by T_E , the sig-

nal will have been received by T_O before time t . So T_O will fall into the black hole with 1 in its output cell if T_E halted and 0 if T_E never halted. Fortunately, T_O can escape annihilation if its trajectory is carefully chosen in advance, says Németi; the rotational forces of the Kerr hole counterbalance the gravitational forces that would otherwise “spaghettify” T_O . T_O thus emerges unscathed from the hole and goes on to use the computed value of the halting function in further computations.

Németi and colleagues emphasize their machine is physical in the sense it is “not in conflict with presently accepted scientific principles” and, in particular, “the principles of quantum mechanics are not violated.”² They suggest humans might “even build” a relativistic computer “sometime in the future.”² This is, of course, highly controversial. However, our point is that Németi’s theoretical countermodel, which counters not only CTT-P-C but also CTT-P and CTDW, helps underscore that the “physical version of the Church-Turing thesis” is quite independent of CTT-O, since the countermodel stands whether or not CTT-O is endorsed. We next reconsider CTT-A.

CTT-A and Computation in the Broad

The continuing expansion of the concept of an algorithm is akin to the extension of the concept of number from integers to signed integers to rational, real, and complex numbers. Even the concept of human computation underwent an expansion; before 1936, computation was conceived of in terms of total functions, and it was Kleene in 1938 who explicitly extended the conception to also cover partial functions.

Gurevich argued in 2012 that formal methods cannot capture the algorithm concept in its full generality due to the concept’s open-ended nature; at best, formal methods provide treatments of “strata of algorithms” that “have matured enough to support rigorous definitions.”²² An important question for computer science is whether CTT-A is a reasonable constraint on the growth of new strata. Perhaps not. In 1982, Jon Doyle¹⁸ suggested equilibrating systems with discrete spectra (such as molecules and other quantum many-body systems) illustrate a concept of effectiveness that is broader than the

classical concept, saying, “[E]quilibrating can be so easily, reproducibly, and mindlessly accomplished” that we may “take the operation of equilibrating as an effective one,” even if “the functions computable in principle given Turing’s operations and equilibrating include non-recursive functions.”

Over the years, there have been several departures from Turing’s 1936 analysis, as the needs of computer science led to a broadening of the algorithm concept. For example, Turing’s fourth axiom, which bounds the number of parts of a system that can be changed simultaneously, became irrelevant when the algorithm concept broadened to cover parallel computations. The future computational landscape might conceivably include more extensive revisions of the concept, if, for example, physicists were to discover that hardware effective in Doyle’s extended sense is a realistic possibility.

If such hardware were to be developed—hardware in which operations are effective in the sense of being “easily, reproducibly, and mindlessly accomplished” but not bounded by Turing computability—then would the appropriate response by computer scientists be to free the algorithm concept from CTT-A? Or should CTT-A remain as a constraint on algorithms, with instead two different species of computation being recognized, called, say, algorithmic computation and non-algorithmic computation? Not much rides on a word, but we note we prefer “effective computation” for computation that is bounded by Turing computability and “neo-effective computation” for computation that is effective in Doyle’s sense and *not* bounded by Turing computability, with “neo” indicating a new concept related to an older one.

The numerous examples of notional “hypercomputers” (see Copeland⁹ for a review) prompt similar questions. Interestingly, a study of the expanding literature about the concept of an infinite-time Turing machine, introduced by Joel Hamkins and Andy Lewis in 2000, shows that a number of computer scientists are prepared to describe the infinite-time machine as computing the halting function. Perhaps this indicates the concept of computation is already in the process of bifurcating into “effective” and “neo-effective” computation.

Conclusion

In the computational literature the term “Church-Turing thesis” is applied to a variety of different propositions usually not equivalent to the original thesis—CTT-O; some even go far beyond anything either Church or Turing wrote. Several but not all are fundamental assumptions of computer science. Others (such as the various physical computability theses we have discussed) are important in the philosophy of computing and the philosophy of physics but are highly contentious; indeed, the label “Church-Turing thesis” should not mislead computer scientists or anyone else into thinking they are established fact or even that Church or Turing endorsed them. C

References

1. Aharonov, D. and Vazirani, U.V. Is quantum mechanics falsifiable? A computational perspective on the foundations of quantum mechanics. Chapter in *Computability: Gödel, Turing, Church and Beyond*, B.J. Copeland, C.J. Posy, and O. Shagrir, Eds. MIT Press, Cambridge, MA, 2013.
2. Andréka, H., Németi, I., and Németi, P. General relativistic hypercomputing and foundation of mathematics. *Natural Computing* 8, 3 (Sept. 2009), 499–516.
3. Bernstein, E. and Vazirani, U. Quantum complexity theory. *SIAM Journal on Computing* 26, 5 (Oct. 1997), 1411–1473.
4. Castelvecchi, D. Paradox at the heart of mathematics makes physics problem unanswerable. *Nature* 528 (Dec. 9, 2015), 207.
5. Church, A. An unsolvable problem of elementary number theory. *American Journal of Mathematics* 58, 2 (Apr. 1936), 345–363.
6. Copeland, B.J. The broad conception of computation. *American Behavioral Scientist* 40, 6 (May 1997), 690–716.
7. Copeland, B.J. Even Turing machines can compute uncomputable functions. Chapter in *Unconventional Models of Computation*, C. Calude, J. Casti, and M. Dinneen, Eds. Springer, Singapore, 1998.
8. Copeland, B.J. Narrow versus wide mechanism: Including a re-examination of Turing’s views on the mind-machine issue. *The Journal of Philosophy* 97, 1 (Jan. 2000), 5–32.
9. Copeland, B.J. Hypercomputation. *Minds and Machines* 12, 4 (Nov. 2002), 461–502.
10. Copeland, B.J. *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life, Plus the Secrets of Enigma*. Oxford University Press, Oxford, U.K., 2004.
11. Copeland, B.J. and Proudfoot, D. Alan Turing’s forgotten ideas in computer science. *Scientific American* 280, 4 (Apr. 1999), 98–103.
12. Copeland, B.J. and Shagrir, O. Do accelerating Turing machines compute the uncomputable? *Minds and Machines* 21, 2 (May 2011), 221–239.
13. Copeland, B.J. and Shagrir, O. Turing versus Gödel on computability and the mind. Chapter in *Computability: Gödel, Turing, Church, and Beyond*, B.J. Copeland, C.J. Posy, and O. Shagrir, Eds. MIT Press, Cambridge, MA, 2013.
14. Cubitt, T.S., Perez-Garcia, D., and Wolf, M.M. Undecidability of the spectral gap. *Nature* 528, 7581 (Dec. 2015), 207–211.
15. Davis, M. Why Gödel didn’t have Church’s thesis. *Information and Control* 54, 1-2 (July 1982), 3–24.
16. Dershowitz, N. and Gurevich, Y. A natural axiomatization of computability and proof of Church’s thesis. *Bulletin of Symbolic Logic* 14, 3 (Sept. 2008), 299–350.
17. Deutsch, D. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 400, 1818 (July 1985), 97–117.

18. Doyle, J. What is Church’s thesis? An outline. *Minds and Machines* 12, 4 (Nov. 2002), 519–520.
19. Eisert, J., Müller, M.P., and Gogolin, C. Quantum measurement occurrence is undecidable. *Physical Review Letters* 108, 26 (June 2012), 1–5.
20. Gandy, R.O. Church’s thesis and principles for mechanisms. In *Proceedings of the Kleene Symposium*, J. Barwise, H.J. Keisler, and K. Kunen, Eds. (Madison, WI, June 1978). North-Holland, Amsterdam, Netherlands, 1980.
21. Goldreich, O. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, New York, 2008.
22. Gurevich, Y. What is an algorithm? In *Proceedings of the 38th Conference on Current Trends in the Theory and Practice of Computer Science* (Špindléuv Mlýn, Czech Republic, Jan. 21–27), M. Bieliková, G. Friedrich, G. Gottlob, S. Katzenbeisser, and G. Turán, Eds. Springer, Berlin, Heidelberg, Germany, 2012.
23. Harel, D. *Algorithmics: The Spirit of Computing, Second Edition*. Addison-Wesley, Reading, MA, 1992.
24. Hopcroft, J.E. and Ullman, J.D. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA, 1979.
25. Kleene, S.C. *Introduction to Metamathematics*. Van Nostrand, New York, 1952.
26. Komar, A. Undecidability of macroscopically distinguishable states in quantum field theory. *Physical Review* 133, 2B (Jan. 1964), 542–544.
27. Kreisel, G. Mathematical logic: What has it done for the philosophy of mathematics? Chapter in *Bertrand Russell: Philosopher of the Century*, R. Schoenman, Ed. Allen and Unwin, London, U.K., 1967.
28. Kripke, S.A. Another approach: The Church-Turing ‘thesis’ as a special corollary of Gödel’s completeness theorem. Chapter in *Computability: Gödel, Turing, Church, and Beyond*, B.J. Copeland, C.J. Posy, and O. Shagrir, Eds. MIT Press, Cambridge, MA, 2013.
29. Lewis, H.R. and Papadimitriou, C.H. *Elements of the Theory of Computation*. Prentice Hall, Upper Saddle River, NJ, 1981.
30. Moschovakis, Y.N. and Paschalis, V. Elementary algorithms and their implementations. Chapter in *New Computational Paradigms: Changing Conceptions of What Is Computable*, S.B. Cooper, B. Lowe, and A. Sorbi, Eds. Springer, New York, 2008.
31. Piccinini, G. The physical Church-Turing thesis: Modest or bold? *The British Journal for the Philosophy of Science* 62, 4 (Aug. 2011), 733–769.
32. Pitowsky, I. The physical Church thesis and physical computational complexity. *Iyyun* 39, 1 (Jan. 1990), 81–99.
33. Post, E.L. Finite combinatory processes: Formulation I. *The Journal of Symbolic Logic* 1, 3 (Sept. 1936), 103–105.
34. Pour-El, M.B. and Richards, I.J. The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics* 39, 3 (Mar. 1981), 215–239.
35. Sieg, W. Mechanical procedures and mathematical experience. Chapter in *Mathematics and Mind*, A. George, Ed. Oxford University Press, New York, 1994.
36. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem (1936); in Copeland.¹⁰
37. Turing, A.M. *Lecture on the Automatic Computing Engine* (1947); in Copeland.¹⁰
38. Turing, A.M. *Intelligent Machinery* (1948); in Copeland.¹⁰
39. Wolfram, S. Undecidability and intractability in theoretical physics. *Physical Review Letters* 54, 8 (Feb. 1985), 735–738.
40. Yao, A.C.C. Classical physics and the Church-Turing thesis. *Journal of the ACM* 50, 1 (Jan. 2003), 100–105.

B. Jack Copeland (jack.copeland@canterbury.ac.nz) is Distinguished Professor of Philosophy at the University of Canterbury in Christchurch, New Zealand, and Director of the Turing Archive for the History of Computing, also at the University of Canterbury.

Oron Shagrir (oron.shagrir@gmail.com) is Schulman Professor of Philosophy and Cognitive Science at the Hebrew University of Jerusalem, Jerusalem, Israel.



AWARD NOMINATIONS SOLICITED

As part of its mission, ACM brings broad recognition to outstanding technical and professional achievements in computing and information technology.

ACM welcomes nominations for those who deserve recognition for their accomplishments. Please refer to the ACM Awards website at <https://awards.acm.org> for guidelines on how to nominate, lists of the members of the 2018 Award Committees, and listings of past award recipients and their citations.

Nominations are due **January 15, 2019** with the exceptions of the Doctoral Dissertation Award (due **October 31, 2018**) and the ACM – IEEE CS George Michael Memorial HPC Fellowship (due **May 1, 2019**).

A.M. Turing Award: ACM's most prestigious award recognizes contributions of a technical nature which are of lasting and major technical importance to the computing community. The award is accompanied by a prize of \$1,000,000 with financial support provided by Google.

ACM Prize in Computing (previously known as the ACM-Infosys Foundation Award in the Computing Sciences): recognizes an early- to mid-career fundamental, innovative contribution in computing that, through its depth, impact and broad implications, exemplifies the greatest achievements in the discipline. The award carries a prize of \$250,000. Financial support is provided by Infosys Ltd.

Distinguished Service Award: recognizes outstanding service contributions to the computing community as a whole.

Doctoral Dissertation Award: presented annually to the author(s) of the best doctoral dissertation(s) in computer science and engineering, and is accompanied by a prize of \$20,000. The Honorable Mention Award is accompanied by a prize totaling \$10,000. Winning dissertations are published in the ACM Digital Library and the ACM Books Series.

ACM – IEEE CS George Michael Memorial HPC Fellowships: honors exceptional PhD students throughout the world whose research focus is on high-performance computing applications, networking, storage, or large-scale data analysis using the most powerful computers that are currently available. The Fellowships includes a \$5,000 honorarium.

Grace Murray Hopper Award: presented to the outstanding young computer professional of the year, selected on the basis of a single recent major technical or service contribution. The candidate must have been 35 years of age or less at the time the qualifying contribution was made. A prize of \$35,000 accompanies the award. Financial support is provided by Microsoft.

Paris Kanellakis Theory and Practice Award: honors specific theoretical accomplishments that have had a significant and demonstrable effect on the practice of computing. This award is accompanied by a prize of \$10,000 and is endowed by contributions from the Kanellakis family, and financial support by ACM's SIGACT, SIGDA, SIGMOD, SIGPLAN, and the ACM SIG Project Fund, and individual contributions.

Karl V. Karlstrom Outstanding Educator Award: presented to an outstanding educator who is appointed to a recognized educational baccalaureate institution, recognized for advancing new teaching methodologies, effecting new curriculum development or expansion in computer science and engineering, or making a significant contribution to ACM's educational mission. The Karlstrom Award is accompanied by a prize of \$10,000. Financial support is provided by Pearson Education.

Eugene L. Lawler Award for Humanitarian Contributions within Computer Science and Informatics: recognizes an individual or a group who have made a significant contribution through the use of computing technology; the award is intentionally defined broadly. This biennial, endowed award is accompanied by a prize of \$5,000, and alternates with the ACM Policy Award.

ACM – AAAI Allen Newell Award: presented to individuals selected for career contributions that have breadth within computer science, or that bridge computer science and other disciplines. The \$10,000 prize is provided by ACM and AAAI, and by individual contributions.

Outstanding Contribution to ACM Award: recognizes outstanding service contributions to the Association. Candidates are selected based on the value and degree of service overall.

ACM Policy Award: recognizes an individual or small group that had a significant positive impact on the formation or execution of public policy affecting computing or the computing community. The biennial award is accompanied by a \$10,000 prize. The next award will be the 2019 award.

Software System Award: presented to an institution or individuals recognized for developing a software system that has had a lasting influence, reflected in contributions to concepts, in commercial acceptance, or both. A prize of \$35,000 accompanies the award with financial support provided by IBM.

ACM Athena Lecturer Award: celebrates women researchers who have made fundamental contributions to computer science. The award includes a \$25,000 honorarium.

For SIG-specific Awards, please visit <https://awards.acm.org/sig-awards>.

Vinton G. Cerf, ACM Awards Committee Co-Chair

Insup Lee, SIG Governing Board Awards Committee Liaison

John R. White, ACM Awards Committee Co-Chair

Rosemary McGuinness, ACM Awards Committee Liaison

YOLANDA GIL
University of Southern California

SUZANNE A. PIERCE
The University of Texas Austin

HASSAN BABAIE
Georgia State University

ARINDAM BANERJEE
University of Minnesota

KIRK BORNE
Booz Allen Hamilton

GARY BUST
Johns Hopkins University

MICHELLE CHEATHAM
Wright State University

IMME EBERT-UPHOFF
Colorado State University

CARLA GOMES
Cornell University

MARY HILL
University of Kansas

JOHN HOREL
University of Utah

LESLIE HSU
Columbia University

JIM KINTER
George Mason University

CRAIG KNOBLOCK
University of Southern California

DAVID KRUM
University of Southern California

VIPIN KUMAR
University of Minnesota

PIERRE LERMUSIAUX
Massachusetts Institute of Technology

YAN LIU
University of Southern California

CHRIS NORTH
Virginia Tech

VICTOR PANKRATIUS
Massachusetts Institute of Technology

SHANAN PETERS
University of Wisconsin-Madison

BETH PLALE
Indiana University Bloomington

ALLEN POPE
University of Colorado Boulder

SAI RAVELA
Massachusetts Institute of Technology

JUAN RESTREPO
Oregon State University

AARON RIDLEY
University of Michigan

HANAN SAMET
University of Maryland

SHASHI SHEKHAR
University of Minnesota

A research agenda for intelligent systems that will result in fundamental new capabilities for understanding the Earth system.

Intelligent Systems for Geosciences: An Essential Research Agenda

MANY ASPECTS OF geosciences pose novel problems for intelligent systems research. Geoscience data is challenging because it tends to be uncertain, intermittent, sparse, multiresolution, and multi-scale. Geosciences processes and objects often have amorphous spatiotemporal boundaries. The lack of ground truth makes model evaluation, testing, and comparison difficult. Overcoming these challenges requires breakthroughs that would significantly transform intelligent systems, while greatly benefitting the geosciences in turn. Although there have been significant and beneficial interactions between the intelligent systems and geosciences communities,^{4,12} the potential for synergistic research in intelligent



systems for geosciences is largely untapped. A recently launched Research Coordination Network on Intelligent Systems for Geosciences followed a workshop at the National Science Foundation on this topic.¹ This expanding network builds on the momentum of the NSF EarthCube initiative for geosciences, and is driven by practical problems in Earth, ocean, atmospheric, polar, and geospace sciences.¹¹ Based on discussions and activities within this network, this article presents a research agenda for intelligent systems inspired by geosciences challenges.

Geosciences research aims to understand the Earth as a system of complex highly interactive natural processes and their interactions with human activities. Current approaches have fundamental shortcomings given the complexity of geosciences data. First, using data alone is insufficient to create models of the very complex phenomena under study so prior theories need to be taken into account. Second, data collection can be most effective if steered using knowledge about existing models to focus on data that will make a difference. Third, to combine disparate data and models across disciplines requires capturing and reasoning about extensive qualifications and context to enable their integration. These are all illustrations of the need for knowledge-rich intelligent systems that incorporate significant amounts of geosciences knowledge.

The article begins with an overview of research challenges in geosciences. It then presents a research agenda and vision for intelligent system to address those challenges. It concludes with an overview of ongoing activities in the newly formed research network of intelligent systems for geosciences that is fostering a community to pursue this interdisciplinary research agenda.

The pace of geosciences investigations today can hardly keep up with the urgency presented by societal needs to manage natural resources, respond to geohazards, and understand the long-term effects of human activities on the planet.⁶⁻¹¹ In addition, recent unprecedented increases in data availability together with a stronger emphasis on societal drivers emphasize the need for research that crosses over traditional

knowledge boundaries. Different disciplines in geosciences are facing these challenges from different motivations and perspectives:

► **Forecasting rates of sea level change in polar ice shelves:** Polar scientists, along with atmospheric and ocean scientists, face an urgent need to understand sea level rise around the globe. Ice-shelf environments represent extreme environments for sampling and sensing. Current efforts to collect sensed data are limited and use tethered robots with traditional sampling frequency and collection limitations. The ability to collect extensive data about conditions at or near the ice shelves will inform our understanding about changes in ocean circulation patterns, as well as feedbacks with wind circulation. New research on intelligent sensors would support selective data collection, on-board data analysis, and adaptive sensor steering. New submersible robotic platforms could detect and respond to interesting situations while adjusting sensing frequencies that could be triggered depending on the data being collected in real time.

► **Unlock deep Earth time:** Earth scientists focus on understanding the dynamics of the Earth, including the interior of the Earth or *deep Earth* (such as tectonics, seismology, magnetic or gravity fields, and volcanic activity) and the near-surface Earth (such as the hydrologic cycle, the carbon cycle, the food production cycle, and the energy cycle). While collecting data from the field is done by individuals in select

locations, the problems under consideration cover spatially vast regions of the planet. Moreover, scientists have been collecting data at different times in different places and reporting results in separate repositories and often unconnected publications. This has resulted in a poorly connected collection of information that makes wide-area analyses extremely difficult and is impossible to reproduce. Earth systems are integrated, but current geoscience data and models are not. To unravel significant questions about topics, such as Deep Earth Time, geoscientists need intelligent systems to efficiently integrate data from disparate locations, data types, and collection efforts within a wide area.

► **Predict critical atmosphere and geospace events:** Atmospheric and geospace science research aims to improve understanding of the Earth's atmosphere and its interdependencies with all of the other Earth components, and to understand the important physical dynamics, relationships, and coupling between the incident solar wind stream, and the magnetosphere, ionosphere, and thermosphere of the Earth. Atmospheric research investigates phenomena operating from planetary to micro spatial scales and from millennia to microseconds. Although the data collected is very large, it is miniscule given the complexity of the phenomena under study. Therefore, the data available must be augmented with knowledge about physical laws underlying the phenomena in order to generate effective models.

► **Detect ocean-land-atmosphere-ice interactions:** Our ability to understand the Earth system is heavily dependent on our ability to integrate geoscience models across time, space, and discipline. This requires sophisticated approaches that support composition and discover structure, diagnose, and compensate for compound model errors and uncertainties, and generate rich visualizations of multidimensional information that take into account a scientist's context.

The accompanying figure illustrates intelligent systems research directions inspired by these geoscience challenges, organized at various scales. Studying the Earth as a system requires fundamentally new capabilities to collect

» key insights

- **Advances in artificial intelligence are needed to collect data where and when it matters, to integrate isolated observations into broader studies, to create models in the absence of comprehensive data, and to synthesize models from multiple disciplines and scales.**
- **Intelligent systems need to incorporate extensive knowledge about the physical, geological, chemical, biological, ecological, and anthropomorphic factors that affect the Earth system while leveraging recent advances in data-driven research.**
- **A new generation of knowledge-rich intelligent systems have the potential to significantly transform geosciences research practices.**

data where and when it matters, to integrate isolated observations into broader studies, to create models in the absence of comprehensive data, and to synthesize models from multiple disciplines and scales. Advances in intelligent systems to develop more robust sensor platforms, more effective information integration, more capable machine learning algorithms, and intelligent interactive environments have the potential to significantly transform geosciences research practices and expand the nature of the problems under study.

A Roadmap for Intelligent Systems Research with Benefits to Geosciences

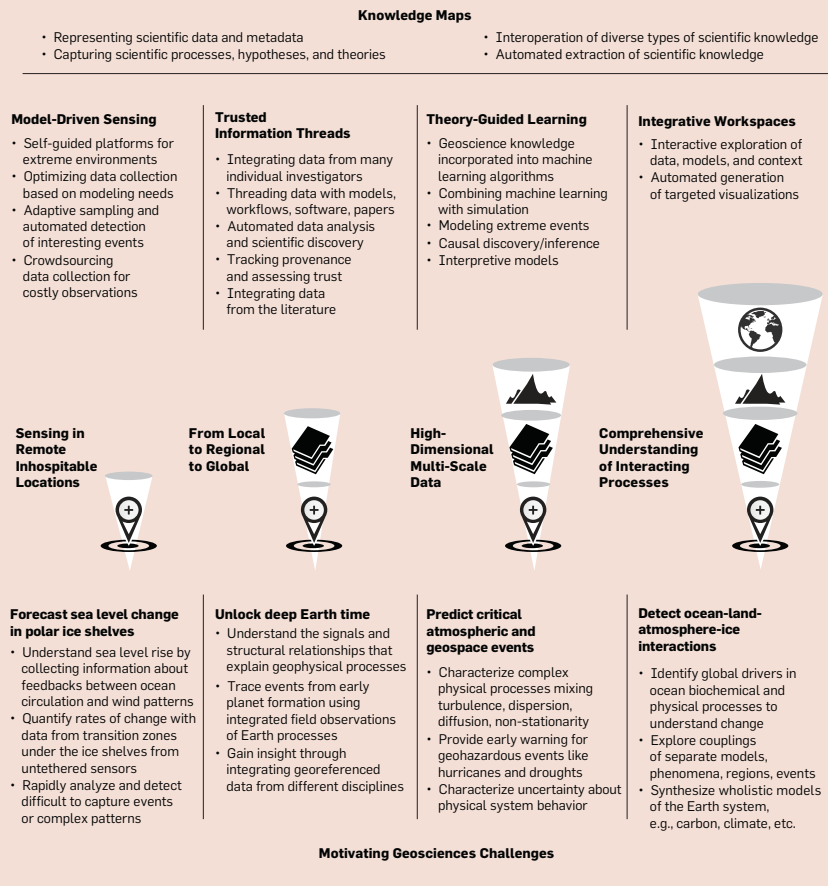
Earth systems phenomena are characterized by nonlinear, multiresolution, multi-scale, heterogeneous, and highly dynamic processes. Geosciences research is also challenged by extreme events and long-term shifts in Earth systems. The data available is intermittent, has significant sources of uncertainty, and is very sparse given the complexity and rich phenomena under study. Therefore, the small sample size of the datasets must be supplemented with the scientific principles underlying geosciences processes in order to guide knowledge discovery. For example, encapsulating knowledge about the physical processes governing Earth system datasets can help constrain the learning of complex nonlinear relationships in geoscience applications, ensuring theoretically consistent results. We need approaches that leverage the advances in data-driven research with methods that exploit the domain knowledge and scientific principles that govern the phenomena under study. These geoscience-aware systems will need to incorporate extensive knowledge about phenomena that combine physical, geological, chemical, biological, ecological, and anthropomorphic factors.

This body of research will lead to a new generation of *knowledge-rich intelligent systems* that contain rich knowledge and context in addition to data, enabling fundamentally new forms of reasoning, autonomy, learning, and interaction. The research challenges for creating knowledge-rich intelligent systems center on five major areas:

AI research.

New research in artificial intelligence (top) will result in a new generation of knowledge-rich intelligent systems that could address the significant challenges faced by geosciences (bottom). Knowledge-rich intelligent systems will exploit knowledge maps containing models and pre-existing knowledge in order to drive sensor data collection, create trusted information threads, power theory-guided learning, and enable integrative analytics.

A Research Agenda for Knowledge-Rich Intelligent Systems



- Knowledge representation and capture:** Capturing scientific knowledge about processes, models, and hypotheses.
- Sensing and robotics:** Prioritizing data collection based on the scientific knowledge available.
- Information integration:** Representing data and models as a “system of systems” where all knowledge is interconnected.
- Machine learning:** Enriching algorithms with knowledge and models of the relevant underlying processes.
- Interfaces and interactive systems:** Exploring and understanding user context using interconnected knowledge.

We describe these five areas in turn. For each area, we introduce major research directions followed by an overarching vision for that area.

Knowledge representation and capture. In order to create knowledge-rich

intelligent systems, scientific knowledge relevant to geoscience processes must be explicitly represented, captured, and shared.

Research directions:

- Representing scientific data and metadata.** Geoscientists are collecting more data than ever before, but raw data sitting on isolated servers is of little utility. Recent work on semantic and Linked Open Data standards enables publishing datasets in Web standard formats with open access licenses, creating links among datasets to further interoperability.² This leads to Web-embedded semantic networks and knowledge graphs that provide vast amounts of open interconnected knowledge about geosciences. Semantics, ontological representations, scientifically accurate concept mappings across domains, knowledge graphs,

and the application of Linked Open Data are all areas of active research to facilitate search and integration of data without a great deal of manual effort.⁵

2. *Capturing scientific processes, hypo-theses, and theories.* To complement the ontologies and data representations just discussed, a great challenge is representing the ever-evolving, uncertain, complex, and dynamic scientific knowledge and information. Important challenges will arise in representing dynamic processes, uncertainty, theories and models, hypotheses and claims, and many other aspects of a constantly growing scientific knowledge base. These representations need to be expressive enough to capture complex scientific knowledge, but they also need to support scalable reasoning that integrates disparate knowledge at different scales. In addition, scientists will need to understand the representations and trust the outcomes.

3. *Interoperation of diverse scientific knowledge.* Scientific knowledge comes in many forms that use different tacit and explicit representations: hypotheses, models, theories, equations, assumptions, data characterizations, and others. These representations are all interrelated, and it should be possible to translate knowledge fluidly as needed from one representation to another. A major research challenge is the seamless interoperation of alternative representations of scientific knowledge, from descriptive to taxonomic to mathematical, from facts to interpretation and alternative hypotheses, from smaller to larger scales, and from isolated processes to complex integrated phenomena.

4. *Authoring scientific knowledge collaboratively.* Formal knowledge representation languages, especially if they are expressive and complex, are not easily accessible to scientists for encoding understanding. A major challenge will be creating authoring tools that enable scientists to create, interlink, reuse, and disseminate knowledge. Scientific knowledge needs to be updated continuously, allow for alternative models, and separate facts from interpretation and hypotheses. These are new challenges for knowledge capture and authoring research. Finally, scientific knowledge should be created

collaboratively, allowing different contributors to weigh in based on their diverse expertise and perspectives.

5. *Automated extraction of scientific knowledge.* Not all scientific knowledge needs to be authored manually. Much of the data known to geoscientists is stored in semi-structured formats, such as spreadsheets or text, and is inaccessible to structured search mechanisms. Automated techniques are needed to identify and import these kinds of data into structured knowledge bases.

Research vision: Knowledge maps. We envision rich knowledge graphs that will contain explicit interconnected representations of scientific knowledge linked to time and space to form multidimensional *knowledge maps*. Interpretations and assumptions will be well documented and linked to observational data and models. Today's semantic networks and knowledge graphs link together distributed facts on the Web, but they contain simple facts that lack the depth and grounding needed for scientific research. Knowledge maps will have deeper spatiotemporal representations of processes, hypotheses, and theories and will be grounded in the physical world, interconnecting the myriad models of geoscience systems.

Robotics and sensing. Knowledge-informed sensing and data collection has great potential to do more cost-effective data gathering across the geosciences.

Research directions:

1. *Optimizing data collection.* Geoscience data is needed across many scales, both spatial and temporal. Since it is not possible to monitor every measurement at all scales all of the time, there is a crucial need for intelligent methods for sensing. New research is needed to estimate the cost of data collection prior to sensor deployment, whether that means storage size, energy expenditure, or monetary cost. A related research challenge is trade-off analysis of the cost of data collection versus the utility of the data to be collected.

2. *Active sampling.* Geoscience knowledge can be exploited to inform autonomous sensing systems to not only enable long-term data collection, but to also increase the effectiveness of sensing through adaptive sam-

pling, resulting in richer datasets at lower costs. Interpreting sensor data onboard allows autonomous vehicles to make decisions guided by real-time variations in data, or to react to unexpected deviations from the current physical model.

3. *Crowdsourcing data collection for costly observations.* Citizen scientists can contribute useful data (for example, collected through geolocated mobile devices) that would otherwise be very costly to acquire. One challenge in data collection through crowdsourcing is in ensuring high quality of data required by geoscience research. A potential area of research is to improve methods of evaluating crowdsourced data collection empirically, and to gain an understanding of the biases involved in the collection process.

Research vision: Model-driven sensing. New research on sensors will create a new generation of devices that will contain more knowledge of the scientific context for the data being collected. These devices will use that knowledge to optimize their performance and improve their effectiveness. This will result in new *model-driven sensors* that will have more autonomy and exploratory capabilities.

Information integration. Data, models, information, and knowledge are scattered across different communities and disciplines, causing great limitations to current geosciences research. Their integration presents major research challenges that will require the use of scientific knowledge for information integration.

Research directions:


1. *Integrating data from distributed repositories.* The geosciences have phenomenal data integration challenges. Most of the hard geoscience problems require that scientists work across sub-disciplinary boundaries and share very large amounts of data. Another facet of this issue is that the data spans a wide variety of modalities and greatly varying temporal and spatial scales. Distributed data discovery tools, meta-data translators, and more descriptive standards are emerging in this context. Open issues include cross-domain concept mapping, entity resolution and scientifically valid data linking, and effective tools for finding, integrating, and reusing data.

2. *Threading scientific information and resources.* Scientific information and digital resources (data, software, models, workflows, papers, and so on) should be interconnected and interrelated according to their authors and use. Research challenges include developing new knowledge networks that accurately and usefully link together people, data, models, and workflows. This research will deepen our understanding of Earth science information interoperability and composition, and of how collaborative expertise and shared conceptual models develop.


3. *Automated data analysis and scientific discovery.* Capturing complex integrative data analysis processes as workflows facilitates reuse, scalable execution, and reproducibility. The pace of research could be significantly accelerated with intelligent workflow systems that automatically select data from separate repositories and carry out integrated analyses of data from different experiments. Through workflows that integrate large amounts of diverse data and interdisciplinary models, intelligent systems will lead to new discoveries.

4. *Tracking provenance and assessing trust.* Incoming data to the integration process must be analyzed for its fit and trustworthiness. The original sources must be documented, as well as the integration processes in order for the information to be understood and trusted. The challenges are in developing appropriate models and automating provenance/metadata generation throughout the integration and scientific discovery processes.

5. *Integrating data from the published literature.* Important historical data in geosciences is often only available in the published literature, requiring significant effort to integrate with new data. Text mining and natural language processing tools can already extract scientific evidence from articles.⁵ Important research challenges in this area include improving the quality of existing information extraction systems, minimizing the effort required to set up and train these systems, and making them scalable through the vast amounts of the published record. Another area of research is georeferencing extracted facts and integrating



This body of research will lead to a new generation of knowledge-rich intelligent systems that contain rich-knowledge and context in addition to data, enabling fundamentally new forms of reasoning, autonomy, learning, and interaction.



newly extracted information with existing data repositories.

Research vision: Trusted information threads. The proposed research will result in a scientifically accurate, useful, and trusted knowledge-rich landscape of data, models, and information that will include integrated broad-scale by-products derived from raw measurements. These products will be described to explain the derivations and assumptions to increase understanding and trust of other scientists. These *trusted information threads* will be easily navigated, queried, and visualized.

Machine learning. In order to address the challenges of analyzing sparse geosciences data given the complexity of the phenomena under study, new machine learning approaches that incorporate scientific knowledge will be needed so that inferences will be obtained better than from data alone.

Research directions:


1. *Incorporation of geoscience knowledge into machine learning algorithms.* Geoscience processes are very complex and high dimensional, and the sample size of the data is typically small given the space of possible observations. For those reasons, current machine learning methods are not very effective for many geoscience problems. A promising approach is to supplement the data with knowledge of the dominant geoscience processes.³ Examples from current work include the use of graphical models, the incorporation of priors, and the application of regularizers. Novel research is needed to develop new machine learning approaches that incorporate knowledge about geoscience processes and use it effectively to supplement the small sample size of the data. Prior knowledge reduces model complexity and makes it possible to learn from smaller amounts of data. Incorporating geoscience process knowledge can also address the high dimensionality that is typical of geoscience data. Prior knowledge constrains the possible relationships among the variables, reducing the complexity of the learning task.

2. *Combining machine learning and simulation approaches.* Machine learning offers data-driven methods to derive models from observational data. In contrast, geoscientists often use simulation models that are built.


Process-based simulation approaches impose conservation principals such as conservations of mass, energy, and momentum. Each approach has different advantages. Data-driven models are generally easier to develop. Process-based simulation models arguably provide reasonable prediction results for situations not represented in the model calibration period, while data-driven models are thought to be unable to extrapolate as well. Yet difficulties in the development of process-based simulation models, such as parameterization and the paucity of clear test results, can draw this claim into question. Intelligent Systems hold the promise of producing the evaluations needed to make the complex approaches used in data-driven and process-model simulation approaches more transparent and refutable. Such efforts will help to use these methods more effectively and efficiently. Novel approaches are needed that combine the advantages of machine learning and simulation models.

3. *Modeling of extreme values.* There are important problems in geosciences that are concerned with extreme events, such as understanding changes in the frequency and spatial distribution of extremely high temperature or extremely low precipitation in response to increase in greenhouse gas emissions. However, existing climate simulation models are often unable to reproduce realistic extreme values and therefore the results are not reliable. Although data science models offer an alternative approach, the heavy-tail property of the extreme values and its spatiotemporal nature poses important challenges to machine learning algorithms. A major challenge is presented by the spatiotemporal nature of the data.

4. *Evaluation methodologies.* Machine learning evaluation methodology relies heavily on gold standards and benchmark datasets with ground-truth labels. In geosciences there are no gold standard datasets for many problems, and in those cases it is unclear how to demonstrate the value of machine learning models. One possible approach involves making predictions, collecting observations, and then adjusting the models to account for differences between prediction and observations. Holding data mining competitions using such data would be a



Novel research is needed to develop new machine learning approaches that incorporate knowledge about geoscience processes and use it effectively to supplement the small sample size of the data.



very effective attractor for the machine learning community. Another alternative could be the creation of training datasets from simulations. Training datasets could be generated that would mimic real data but also have ground truth available, providing opportunity to rigorously train, test and evaluate machine learning algorithms.

5. *Causal discovery and inference for large-scale applications.* Many geoscience problems involve fundamental questions around causal inference. For example, what are the causes of more frequent occurrences of heat waves? What could be the causes for the change of ocean salinity? While it may be very hard to prove causal connections, it is possible to generate new (likely) hypotheses for causal connections that can be tested by a domain expert using methods such as generalization analysis of causal inference, causal inference in presence of hidden components, domain adaption and subsample data, Granger graphical models and causal discovery with probabilistic graphical models. Given the large amount of data available, we are in a unique position to use these advances to answer fundamental questions around causal inference in the geosciences.

6. *Novel machine learning methods motivated by geosciences problems.* A wide range of advanced machine learning methods could be effectively applied to geoscience problems. Moreover, geosciences problems drive researchers to develop entirely new machine learning algorithms. For example, attempts to build a machine learning model to predict forest fires in the tropics using multispectral data from earth observing satellites led to a novel methodology for building predictive models for rare phenomena¹ that can be applied in any setting where it is not possible to get high-quality labeled data even for a small set of samples, but poor-quality labels (perhaps in the form of heuristics) are available for all samples. Machine learning methods have already shown great potential in a few specific geoscience applications, but significant research challenges remain in order for those methods to be widely and easily applicable for other areas of geoscience.

7. *Active learning, adaptive sampling, and adaptive observations.* Many geoscience applications involve learn-

ing highly complex nonlinear models from data, which usually requires large amounts of labeled data. However, in most cases, obtaining labels can be extremely costly and demand significant effort from domain experts, costly experiments, or long time periods. Therefore, a significant research challenge is to effectively utilize a limited labeling effort for better prediction models. In machine learning, this area of research is known as active learning. Many relevant active sampling algorithms, such as clustering-based active learning, have been developed. New challenges emerge when existing active learning algorithms are applied in geosciences, due to issues such as high dimensionality, extreme events, and missing data. In addition, in some cases, we may have abundant labeled data for some sites while being interested in building models for other locations (for example, remote areas). Transfer active learning aims to solve the problem with algorithms that can significantly reduce the number of labeling requests and build an effective model by transferring the knowledge from areas with large amount of labeled data. Transfer active learning is still in the early stages and many opportunities exist for novel machine learning research.

8. *Interpretive models.* In the past few decades, we have witnessed many successes of powerful but complex machine learning algorithms, exemplified by the recent peak of deep learning models. They are usually treated as a black box in practical applications, but have been accepted by more communities given the rise of big data and their modeling power. However, in applications such as geosciences, we are interested in both predictive modeling and scientific understanding, which requires explanatory and interpretive modeling. A significant research area for machine learning is the incorporation of domain knowledge and causal inference to enable the design of interpretive machine learning approaches that can be understood by scientists and related to existing geosciences theories and models.

Research vision: Theory-guided learning. Geosciences data presents new challenges to machine learning approaches due to the small sample sizes relative to the complexity and non-linearity of

the phenomena under study, the lack of ground truth, and the high degree of noise and uncertainty. New approaches for *theory-guided learning* will need to be developed, where knowledge about underlying geosciences processes will guide the machine learning algorithms in modeling complex phenomena.

Intelligent user interaction. Scientific research requires well-integrated user interfaces where data can easily flow from one to another, and that include and exploit the user's context to guide the interaction. New forms of interaction, including virtual reality and haptic interfaces, should be explored to facilitate understanding and synthesis.

Research directions:

1. *Knowledge-rich context-aware recommender systems.* Scientists would benefit from proactive systems that understand the task at hand and make recommendations for potential next steps, suggest datasets and analytical methods, and generate perceptually effective visualizations. A major research challenge is to design recommender systems that appropriately take into account the complex science context of a geoscientist's investigation.

2. *Embedding visualizations throughout the science process.* Pervasive use of visualizations and direct manipulation interfaces throughout the science process would need to link data to hypotheses and allow scientists to experience models from completely new perspectives. These visualization-based interactive systems require research on the design and validation of novel visual representations that effectively integrate diverse data in 2D, 3D, multidimensional, multiscale, and multispectral views, as well as how to link models to the relevant data used to derive them.

3. *Intelligent design of rich interactive visualizations.* In order to be more ubiquitous throughout the research process, visualizations must be automatically generated and be interactive. One research challenge is to design visualizations. Another challenge is the design of visualizations that fit a scientist's problem. An important area of future research is the interactive visualizations and direct manipulation interfaces would enable scientists to explore data and

gain a better understanding of the underlying phenomena.

4. *Immersive visualizations and virtual reality.* There are new opportunities for low-cost usable immersive visualizations and physical interaction techniques that virtually put geoscientists into the physical space under investigation, while also providing access to other related forms of data. This research agenda requires bridging prior distinctions in scientific visualization, information visualization, and immersive virtual environments.

5. *Interactive model building and refinement through visualizations that combine models and data.* Interactive environments for model building and refinement would enable scientists to gain improved understanding on how models are affected by changes in initial data and assumptions, how model changes affect results, and how data availability affects model calibration. Developing such interactive modeling environments requires visualizations that integrate data with models, ensembles of models, model parameters, model results, and hypothesis specifications. These integrated environments would be particularly useful for developing machine learning approaches to geosciences problems, for example in assisting with parameter tuning and selecting training data. A major challenge is the heterogeneity and complexity of these different kinds of information that needs to be represented.

6. *Interfaces for spatiotemporal information.* The vast majority of geosciences research products is geospatially localized and with temporal references. Geospatial information requires specialized interfaces and data management approaches. New research is needed in intelligent interfaces for spatiotemporal information that exploit the user's context and goals to identify implicit location, to disambiguate textual location specification, or to decide what subset of information to present. The small form factor of mobile devices is also constraint in developing applications that involve spatial data.

7. *Collaboration and assistance for data analysis and scientific discovery processes.* Intelligent workflow systems could help scientists by automating routine aspects of their work.

Because each scientist has a unique workflow of activities, and because their workflow changes over time, a research challenge is that these systems need to be highly flexible and customizable. Another research challenge is to support a range of workflows and processes, from common ones that can be reused to those that are highly exploratory in nature. Such workflows systems must enable collaborative design and analysis and be able to coordinate the work of teams of scientists. Finally, workflow systems must also support emerging science processes, including crowdsourcing for problems such as data collection and labeling.

Research vision: Integrative workspaces. New research is required to allow scientists to interact with all forms of knowledge relevant to the phenomenon at hand, to understand uncertainties and assumptions, and to provide many alternative views of integrated information. This will result in user interfaces focused on *integrative workspaces*, where visualizations and manipulations will be embedded throughout the analytic process. These new intelligent user interfaces and interaction modalities will support the exploration not only of data but of the relevant models and knowledge that provide context to the data. Research activities will flow seamlessly from one user interface to another, each appropriate to the task at hand and rich in user context.

Conclusion

This article presented research opportunities in knowledge-rich intelligent systems inspired by geosciences challenges. Crucial capabilities are needed that require major research in knowledge representation, selective sensing, information integration, machine learning, and interactive analytics.

Enabling these advances requires intelligent systems and geosciences researchers work together to formulate knowledge-rich frameworks, algorithms, and user interfaces. Recognizing that these interactions are not likely to occur without significant facilitation, a new Research Coordination Network on Intelligent Systems for Geosciences has been created to enable sustained communication

across these fields that do not typically cross paths. This network focuses on three major goals. First, the organization of joint workshops and other forums will foster synergistic discussions and collaborative projects. Second, repositories of challenge problems and datasets with crisp problem statements will lower the barriers to getting involved. Third, a curated repository of learning materials to educate researchers and students alike will reduce the steep learning curve involved in understanding advanced topics in the other discipline. Additionally, members of the Research Coordination Network are engaging other synergistic efforts, programs, and communities, such as artificial intelligence for sustainability, climate informatics, science gateways, and the U.S. NSF Big Data Hubs.

A strong research community in this area has the potential to have transformative impact in artificial intelligence research with significant concomitant advances in geosciences as well as in other science disciplines, accelerating discoveries and innovating how science is done.

Acknowledgments

This work was sponsored in part by the Directorate for Computer and Information Science and Engineering (CISE) and the Directorate for Geosciences (GEO) of the U.S. National Science Foundation under awards IIS-1533930 and ICER-1632211. We thank NSF CISE and GEO program directors for their guidance and suggestions, in particular Hector Muñoz-Avila and Eva Zanzerkia for their guidance, and Todd Leen, Frank Olken, Sylvia Spengler, Amy Walton, and Maria Zemankova for suggestions and feedback. We also thank all the participants in the Research Coordination Network on Intelligent Systems for Geosciences for creating the intellectual space for productive discussions across these disciplines. □

References

1. Gil, Y. and Pierce, S. (Eds). Final Report of the 2015 NSF Workshop on Information and Intelligent Systems for Geosciences. National Science Foundation Workshop Report, October 2015; <http://dl.acm.org/collection.cfm?id=C13> and <http://is-geo.org/>
2. Berners-Lee, T. Linked data. *Design Issues* (retrieved Nov. 11, 2017); <https://www.w3.org/DesignIssues/LinkedData.html>
3. Karpatne, A. et al. Theory-guided data science: A new

- paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* 29, 10 (2017) 2318–2331.
4. Mithal, V., Nayak, G., Khandelwal, A., Kumar, V., Oza, N.C. and Nemani, R. RAPT: Rare class prediction in absence of true labels. *IEEE Transactions on Knowledge and Data Engineering*, 2017; DOI: 10.1109/TKDE.2017.2739739.
 5. Narock, T. and Fox, P. The Semantic Web in Earth and space science. Current status and future directions. *Studies in the Semantic Web*. IOS Press, 2015.
 6. National Research Council, Committee on Challenges and Opportunities in the Hydrologic Sciences, Water Science and Technology Board, Division on Earth and Life Studies. Challenges and Opportunities in the Hydrologic Sciences. National Academies Press, Washington, D.C., 2012, 188. ISBN: 978-0-309-22283-9.
 7. National Research Council, Committee on a Decadal Strategy for Solar and Space Physics (Heliophysics); Space Studies Board; Aeronautics and Space Engineering Board; Division of Earth and Physical Sciences. Solar and Space Physics: A Science for a Technological Society. National Academies Press, Washington, D.C., 2013, 466. ISBN 978-0-309-16428-3.
 8. National Research Council, Committee on Guidance for NSF on National Ocean Science Research Priorities: Decadal Survey of Ocean Sciences, Ocean Studies Board; Division on Earth and Life Studies. Sea Change: 2015–2025 Decadal Survey of Ocean Sciences. National Academies Press, Washington, D.C., 2014, 98. ISBN 978-0-309-36688-5.
 9. National Research Council, Committee on New Research Opportunities in the Earth Sciences. New Research Opportunities in the Earth Sciences at the National Science Foundation. National Academies Press, Washington, D.C., 2012, 216. ISBN 978-0-309-21924-2.
 10. National Research Council, Committee to Review the NSF AGS Science Goals and Objectives. Review of the National Science Foundation's Division on Atmospheric and Geospace Sciences Goals and Objectives Document. National Academies Press, Washington, D.C., 2014, 36. ISBN 978-0-309-31048-2.
 11. National Science Foundation. Dynamic Earth: GEO Imperatives and Frontiers 2015–2020. Advisory Committee for Geosciences, 2014.
 12. Peters, S.E., Zhang, C., Livny, M. and Ré, C. A machine reading system for assembling synthetic paleontological databases. *PLoS ONE* 9, 12 (2014).

Yolanda Gil, University of Southern California; **Suzanne A. Pierce**, The University of Texas Austin; **Hassan Babaie**, Georgia State University; **Arindam Banerjee**, University of Minnesota; **Kirk Borne**, Booz Allen Hamilton; **Gary Bust**, Johns Hopkins University; **Michelle Cheatham**, Wright State University; **Imme Ebert-Uphoff**, Colorado State University; **Carla Gomes**, Cornell University; **Mary Hill**, University of Kansas; **John Horel**, University of Utah; **Leslie Hsu**, Columbia University; **Jim Kinter**, George Mason University; **Craig Knoblock**, University of Southern California; **David Krum**, University of Southern California; **Vipin Kumar**, University of Minnesota; **Pierre Lermusiaux**, Massachusetts Institute of Technology; **Yan Liu**, University of Southern California; **Chris North**, Virginia Tech; **Victor Pankratius**, Massachusetts Institute of Technology; **Shanan Peters**, University of Wisconsin-Madison; **Beth Plale**, Indiana University Bloomington; **Allen Pope**, University of Colorado Boulder; **Sai Ravela**, Massachusetts Institute of Technology; **Juan Restrepo**, Oregon State University; **Aaron Ridley**, University of Michigan; **Hanan Samet**, University of Maryland; **Shashi Shekhar**, University of Minnesota

Correspondence regarding this article should be directed to **Yolanda Gil** (gil@isi.edu).

Copyright held by authors/owners.



Watch the authors discuss this work in the exclusive *Communications* video. <https://cacm.acm.org/videos/intelligent-systems-for-geosciences>

Classical mathematical game theory helps to evolve the emerging logic of identity in the cyber world.

BY WILLIAM CASEY, ANSGAR KELLNER, PARISA MEMARMOSHREFI, JOSE ANDRE MORALES, AND BUD MISHRA

Deception, Identity, and Security: The Game Theory of Sybil Attacks

“When the world is destroyed, it will be destroyed not by its madmen but by the sanity of its experts and the superior ignorance of its bureaucrats.”

— John le Carré

DECADES BEFORE THE advent of the Internet, Fernando António Nogueira Pessoa assumed a variety of identities with the ease that has become common in cyber-social platforms—those where cyber technologies play a part in human activity (for example, online banking, and social networks). Pessoa, a Portuguese poet, writer, literary critic, translator, publisher, and philosopher, wrote under his own name as well as 75 imaginary identities. He would write poetry or prose using one identity, then criticize that writing using another identity, then defend the original writing using yet another identity. Described by author Carmela Ciuraru as “the loving ringmaster, director, and traffic cop of his literary crew,” Pessoa is one

» key insights

- **Cyber systems have reshaped the role of identity. The low cost to mint cyber identities facilitates greater identity fluidity. This simplicity provides a form of privacy via anonymity or pseudonymity by disguising identity, but also hazards proliferation of deceptive, multiple and stolen identities. With growing connectivity, designing the verification/management algorithms for cyber identity has become complex, and requires examining what motivates such deception.**
- **Signaling games provide a formal mathematical way to analyze how identity and deception are coupled in cyber-social systems. The game theoretic framework can be extended to reason about dynamical system properties and behavior traces.**

WANETS and Hastily Formed Networks

Wireless ad hoc networks (WANETs) consist of spatially distributed autonomous devices (network nodes) that can exchange data without direct physical connections. The nodes do not rely on an existing infrastructure but can form an on-demand network without any manual configuration. WANETs are used in a variety of application areas and are likely to play an important role in the upcoming Internet of Things (IoT) application areas such as smart cities, environmental monitoring, health care monitoring, industrial monitoring, and hastily formed networks (HFNs).

Given the multi-hop nature of WANETs, the risks of any single node's non-cooperative behavior are apparent: information leakage, disinformation, denial of service, among others. Particularly, the use of Sybil nodes is a serious problem.

WANETs provide an elegant framework for many novel applications of ad hoc communication, most of which can be abstracted in terms of information-asymmetric games like those we describe.

Case study: Haiti earthquake HFN: One particularly relevant example took place in the aftermath of Haiti's devastating 2010 earthquakes. There, a hastily assembled information-sharing network to coordinate emergency responders and relief efforts was viewed as a pivotal moment in humanitarian relief efforts.²¹

Notwithstanding the new role and possible benefits of cyber-social systems in this context, the after-action report indicated security and privacy concerns that hampered information sharing. Rapid consignment among responders and relief workers, whose identities, reputations, and individual utilities are not necessarily known to the others a priori (see sidebar "Ant Colonies") is clearly necessary for pooling information to save lives and relieve suffering. Within this context a dishonest identity motivated by conflict or personal gain could easily damage the relief effort.

of the foremost Portuguese poets and a contributor to the Western canon. The story of Pessoa illustrates a key insight that holds true for the cyber-social systems of today: Identity costs little in the way of minting, forming, and maintaining yet demands a high price for its timely and accurate attribution to physical agency.

Along with the low cost of minting and maintaining identities, a lack of constraints on using identities is a primary factor that facilitates adversarial innovations that rely on deception. With these factors in mind, we study the following problem: Will it be possible to engineer a decentralized system that can enforce honest usage of identity via mutual challenges and costly consequences when challenges fail? The success of such an approach will remedy currently deteriorating situations without requiring new infrastructure. For example, such a system should be able to reduce fake personas in social engineering attacks, malware that mimics the attributes of trusted software, and Sybil attacks that use fake identities to penetrate ad hoc networks.

Note that many cyber-physical facilities—those where a physical mechanism is controlled or monitored by computer algorithms and tied closely

to the internet and its users (for example, autonomous cars, medical monitoring)—also aim to enable users to remain anonymous and carry out certain tasks with only a persistent but pseudonymous identity. This form of short-term identity (especially in the networks that are ad hoc, hastily formed, and short lived) can remain uncoupled from a user's physical identity and allow them to maintain a strong form of privacy control. How can this dichotomy, namely trading off privacy for transparency in identity, be reconciled? The emerging logic underlying identity (what types of behaviors are expected, stable, possible) will also be central to avoiding many novel and hitherto unseen, unanticipated, and unanalyzed security problems.

Our approach is founded upon traditional *mathematical game theory*, but is also inspired by several mechanisms that have evolved in biology. Here, we analyze a game theoretic model that evolves cooperative social behavior, learning, and verification to express the strategy of costly signaling. We further suggest this could scale within cyber-social systems.

Road map. Our approach starts with mathematical game theory to analyze decisions concerning identity. Central

to the dilemma are privacy and intent, and these notions are captured with information asymmetry (for example, an agent's true identity vs. the agent's purported identity) and utility (that is, the agent's preference of identity use). We argue this scenario is best captured with a *classical signaling game*, a dynamic Bayesian two-player game, involving a Sender who (using a chosen identity) signals a Receiver to act appropriately. With the identity signaling game defined, the communication among agent identities is a repeated signaling game played among peers. Throughout communications, agents remain uncertain of both the strategies implemented by other identities and the true physical agent controlling those identities. We treat the population of agents as *dynamic* (that is, allowing agents to be removed from the population and be replaced by mutants who use modified strategies) and *rational* (allowing them to preferentially seek greater payoff). By specifying the procedures of direct and vicarious learning we construct a dynamical system familiar to evolutionary game theory. However, we control the parameters in this system associated with evolution rates. Using these building blocks we synthesize models in order to create population simulations and empirically evaluate Nash and weaker equilibria. We present experiments that focus on how ad hoc information flows within networks and examine mechanisms that further stabilize cooperative equilibria. Results are presented and conclusions drawn by outlining the design of cooperativity-enhancing technologies and how such mechanisms could operate in the open and among deceptive types.

Motivation. Novel ad hoc network-communication techniques (for example, formed hastily in the wake of a disaster or dynamically among a set of nearby vehicles) blur the boundaries between cyber and physical purposefully for the benefits of their cohesion. Within these innovations security concerns have centered on identity deception.²⁶ Here, we motivate our game theoretic models via illustrative examples from wireless ad hoc networks (WANETs) and hastily formed networks (HFNs) for humanitarian assistance (see the sidebar "WANETS

and Hastily Formed Networks”) under Sybil attacks. A Sybil attack involves forging identities in peer-to-peer networks to subvert a reputation system and is named after the character Sybil Dorsett, a person diagnosed with dissociative identity disorder. Within the framework of game theory, the Sybil attack is viewed by how agents reason and deliberate under uncertainty as well as control deception in an information asymmetric setting (see “Defining Deception” for a definition of game theoretic deception). Looking to the future, as the distinction between the cyber and physical fades, attacks such as these will very likely pose existential threats to our rapidly growing cyber-physical infrastructure. Hence, there is urgency to the problem.

Conceptual Building Blocks

Here, we construct the signaling game theory of identity within cyber-social systems. The effects of repeated play and evolutionary dynamics provide the conditions under which the theory admits equilibria.

Agency, identity, and signaling. An agent is the notion of a decision maker informed by various faculties. In our setting, an agent’s utility models preferences related to the possible use of pseudonymous identity and actions upon receiving information from other pseudonymous identities.

For example, in the WANET setting the network nodes act as identities, themselves a proxy to the root physical agent controlling them. Thus, a physical agent who constructs a deception via a screen manages a Sybil node: the node’s physical agent appears unknown, murky, or rooted elsewhere.

To create convincing fake identities, a root agent must maintain the act when challenged. One approach to design costly signaling within cyber-social systems is to add risks into the required decisions for maintaining fake identities. We use the term *M-coin* to represent assets held at risk when an agent’s identity is challenged. For example, the bio-inspired protocol detailed in Casey et al.⁶ and simplified in the sidebars “*Costly Signaling*” and “*Ant Colonies*” imposes costly signaling with a digital form of the ant’s Cuticular Hydrocarbon Chemicals (CHCs). Analogously M-coins, encoded digi-

Costly Signaling

The signaling games do have certain kinds of Nash equilibria: the trivial ones being *pooling* and *babbling equilibria*, but more interesting ones being the so-called *separating equilibria*—or their combinations. In a pooling equilibrium, senders with different types all choose the same message. However, in a semi-separating equilibrium some types of senders choose the same message and other types choose different messages; in contrast, in a fully separating equilibrium senders with different types always choose different messages. A fully separating equilibrium requires more message types than sender types. Since the sender types are private, there is ample room for deception in such equilibria.

Various mechanisms can be introduced to tame the deception in the system: adding non-strategic players (for example, recommenders and verifiers); imposing credible and non-credible threats; or making signaling costly. In eusocial species, such as ant colonies, Cuticular Hydrocarbon Chemicals (CHCs)—chemicals impossible to fabricate without engaging the queen ant—provide an example of costly signaling and its use in taming identity and other forms of deception. Other natural examples in the context of mate selection in a biological species were brought to light by the mathematical biologist Ronald Fisher. Fisher argued that although genotypes are private and phenotypes are signaled, possible deceptions are overcome if there are “preference genes” pleiotropically correlated with the “signal genes” governing display traits in males; choosier females discover more preferred mates by selecting showier males, since showy signaling is naturally costly.

An elegant technological example of costly signaling may be found in the technology of crypto-coins such as bitcoins and in the recording and verification of signals by non-strategic agents such as bitcoin miners who maintain a record of signals in a block-chain but engage in solving a computationally costly combinatorial problem. Here, a sender with a bitcoin wallet and a persistent identity (associated with a private signing key and public verification key) sends a signed signal encoding a payment, which the intended receiver can verify for non-repudiability and other financial constraints and act accordingly to update their own bitcoin wallet. Since the receiver is unable to protect against deceptions that involve other global temporal properties (for example, double-spending), this system requires a coupled data structure (a distributed ledger) as well as verifiers to create, maintain, and check the ledger. However, this system is subject to a costly computational investment. For example, we proposed an M-coin to devise cyber-secure systems, where the M-coins are obtained by repeatedly proving properties of one’s attack surface, depending on how M-coins expire and diffuse. The similarities between M-coins and CHCs can be further exploited in designing other bio-inspired technologies.

Ant Colonies

In certain eusocial species such as ant colonies, one encounters rather sophisticated strategies involving costly signaling and credible and non-credible threats (see “Costly Signaling”). In ant colonies, each ant has a CHC profile in which diverse information about the ant itself and its environment can be encoded.^{15,28} For example, ants make use of pheromones to reinforce good paths between the nest and a food source and communicate via chemical substances to inform nestmates about these good paths. In addition to the use of pheromones for marking routes, auxiliary information is stored in an ant’s CHC profile; this information includes diet, genetics, and common nesting materials. Thus, ants from a colony where members share a certain diet have a similar CHC profile that enables them to identify non-nest members. Since CHC profiles are thought to be impossible to fabricate (without active participation by the queen ant), their use in communication by ants is an example of costly signaling (see “Defining Deception”).

Nature, and its more fluid notion of identity, has evolved highly robust solutions to identity management, allowing colonies to survive even in dynamic and contested environments. Therefore, the CHC profile also suggests that a bio-inspired generalization could protect technological systems. To achieve this, several challenging problems must be worked out to ensure the essential physical properties of CHC profiles are retained in their synthesized digital counterparts. A combination of design techniques like crypto-coins (for example, M-coins) can be used to share identity information and to subject data to a variety of cryptographically guaranteed constraints; however, some work remains to ensure physically accurate constraints analogous to those involved in chemical signaling.

Figure 1. Identity: Trust or verify.

The outcomes (as links) of an information-asymmetric game played by two types of signaling agents: informed senders (left) and uninformed receivers (right) who may either trust or verify. A system that can promote the peer interaction types represented by parallel links and minimize the peer interaction types represented by the crossing links will more robustly control deceptions and verification expenses within a system. The fundamental tools impose costly signaling (for example, using M-coins). There are analogous systems used by eusocial organisms to maintain identity and reputation by diffusing costly hard-to-produce Cuticular Hydrocarbon Chemicals (CHCs): in ant colonies only the queen ants produce and distribute such chemicals.

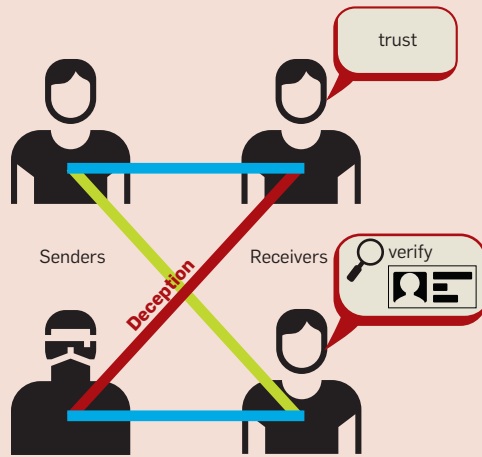
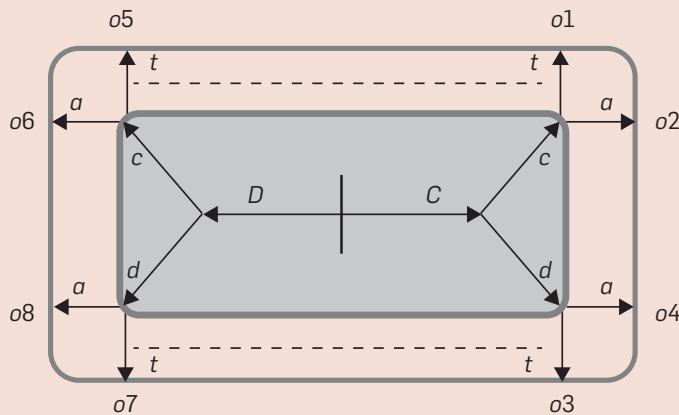


Figure 2. Extensive form games.

The game below is played between a sender identity and a receiver, where the senders, endowed with invisible types by nature: *C* (Cooperative) and *D* (Defective), signal the receivers by sending messages, *c* or *d*, either honestly or deceptively. The game starts in the center of the figure with the sender being assigned a type, which is only known to the sender, and the sender branches to the left or right. The sender then signals *c* (branching up) or *d* (branching down). The receiver, who knows the persistent pseudo-identity of the sender, but not the type, may trust the sender or verify (audit) the sender. The challenge results in different utilities for the senders and the receivers, which they rationally optimize. The inner box encapsulates the selections of the agent utilizing the identity. The audit report may also be made visible to the recommenders and verifiers, thus affecting the reputation (and other credible threats) assigned to the sender's identity.



tally but constrained like CHCs, aim to have similar effects for the utility and identity of nodes within a WANET.

The game. Traditional mathematical game theory^{23,35} models scenarios where outcomes depend on multiple agent preferences. Not all outcomes are alike; under various conditions some outcomes feature greater stability (that is, non-deviation)^{24,25} and are computable.^{16,17,27} Interesting game scenarios yield differing rewards to agents depending on outcome. Thus, agents evaluate scenarios insofar as common

and private knowledge allows, and they act rationally (that is, to optimize utility) by selecting their own action in the context of how other agents act. As the case of Pessoa's creative use of identities suggests, private knowledge is important in shaping outcomes.

To accommodate these types of scenarios, game theory has developed a branch of models known as incomplete/partial information games,^{22,30} of which the Lewis signaling game is one example.^{4,14,19,31,34} Signaling games have been studied in diverse

contexts, including economics and biology,^{1,13,18,20,29,33,36} particularly for evaluating the stability of honest signaling when agents have a partially common interest and where the role of costly signaling and credible deterrence is widely recognized. Applications to cybersecurity are addressed in these references.⁶⁻¹² The simplest such signaling game involving identity focuses on the possibility that during an encounter, a sender node *S* may use a strategic deception by claiming either a fabricated identity or making a malicious attempt to impersonate another's identity. Within a WANET we will consider two natural types of nodes \mathcal{T}_C and \mathcal{T}_D to indicate respectively a *cooperative node* that employs no deceptions (preserving the desired systemwide properties of identity management), and a *deceptive node* that directly employs a deception. In either case, the node will communicate a signal to a receiver node *R* including a status of *c* to indicate it is cooperative with respect to system security, or a status of *d* to indicate anomalous behavior (such as compromised status). A receiver node *R*, given the signal of a sender node *S* but unaware of the sender node's true type, must select an action to take.

One option for the receiver is to simply trust the sender node, denoted as *t*; alternatively, the receiver node may pose a challenge action, denoted as *a*, which creates an attempt to reveal the sender's nature and leads to costly outcomes for deception. While any individual challenge may not reveal completely the nature of a sender, repeated challenges may eventually expose Sybil identities, as senders who are frequently challenged are under pressure to manage their resources for verifying their identity.

We sketch the outcomes of an encounter scenario graphically with an extensive-form game tree illustrated in Figure 2. Starting in the center, the sender *S* has type \mathcal{T}_C (cooperative) or \mathcal{T}_D (deceptive). Next, the sender selects a signal *c* (cooperative) or *d* (otherwise); the receiver selects an action *t* (trust) or *a* (challenge). We explore the outcomes and payoffs for identity as illustrated in the accompanying table.

Outcomes. Outcome o_1 describes a

sender S that is cooperative by nature and offers a nominal proof of identity to the receiver R . The receiver R then trusts S and acts upon the information provided, for example, relaying the communicated message.

Outcome o_2 describes a scenario like o_1 , except the receiver R challenges S to provide a more rigorous proof of identity. In this case, given the cooperative nature of the sender, the challenge is unnecessary, netting cost burdens to maintaining a trusted network.

Outcome o_3 describes a cooperative sender S not willing (or able) to offer a nominal proof of identity (for example, after being repeatedly but maliciously challenged by “suspicious” receivers to the point of insolvency).^a The receiver R nonetheless trusts S , and in this case the exchange is altruistic, helping to recover a trustworthy node in distress.

For brevity, we describe only one more outcome here. Outcome o_5 describes a sender S that is deceptive but offers a nominal proof of identity. The receiver R trusts S and acts upon the information and the receiver’s misguided trust of the deceptive identity is costly.

Signaling games involve asymmetric information constraints for the receiver; without the sender’s type, the receiver cannot distinguish outcome o_1 from o_6 , nor o_3 from o_8 . By selecting the challenge action, the receiver exchanges additional resource cost to partially distinguish among these outcomes. From the point of view of a trustworthy network, we summarize outcomes $\{o_1, o_3\}$ as naturally supporting, while $\{o_5, o_7\}$ are the most destructive; outcomes $\{o_2, o_4\}$ add unnecessary cost, and $\{o_6, o_8\}$, although they add cost, are necessary and effective recourse given deceptive types.

The payoff structure of the table depends on four parameters. We let A be the reward extracted by the deceptive sender at the loss of the trusting receiver;^b let B be the benefit enjoyed by both sender and receiver nodes acting cooperatively in message passing; let C be the cost of challenging a node for additional proof concerning its identity without knowing sender’s

type; and let D be the imputed cost to the sender for being deceptive (identified by a receiver’s challenge).

Repeated games and strategy. Repeated interactions occur as a sequence of plays between two identities. While in classical signaling games there is little need for a distinction to be made between identity and agent, here we highlight identity fluidity with which an identity or cyber asset can be usurped by another agent. Games are played between two

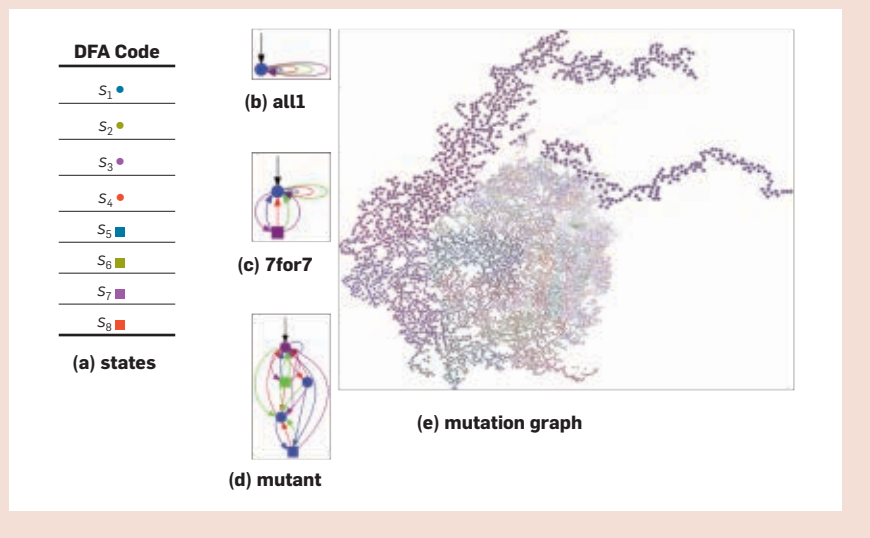
identities, and identities are bound to physical agents (the resident decision control at the time of play). Agent types will remain fixed by nature but note that in subsequent plays the control of an identity can pass from one agent to another, consequently the type changes accordingly. This type of perturbation is intended to be explored by our model, in order that cybersecurity issues such as Sybil attacks (where identities are stolen or fabricated) can be adequately ex-

Outcome labels, payoff, transaction costs, and DFA codes for identity management signal game.

Outcome labels, payoff (S, R), transaction cost, and encoding						
Sender S		Receiver R	Outcomes			DFA Code
Type	Signal	action	Label	Payoff	tcost	
\mathcal{T}_C	c	trust	o_1	(B, B)	1	S_1 ●
		challenge	o_2	$(0, -C)$	1	S_2 ●
	d	trust	o_3	(B, B)	1	S_3 ●
		challenge	o_4	$(0, -C)$	1	S_4 ●
\mathcal{T}_D	c	trust	o_5	$(A, -A)$	0.5	S_5 ■
		challenge	o_6	$(-D, -C)$	0.5	S_6 ■
	d	trust	o_7	$(A, -A)$	0.5	S_7 ■
		challenge	o_8	$(-D, -C)$	0.5	S_8 ■

Figure 3. Evolutionary games and dynamics.

Evolutionary games involve a population of players who play repeated games with encountered peers. To play these games, each player selects a strategy to implement. Each state (a) prescribes send/receive actions according to accompanying table. Strategies, such as those in (b), (c), and (d) are composed of states and conditional logic to facilitate Markovian memory. Formally, strategies are encoded as a labeled transition system of states (for example, DFA). The population of players will select and implement strategies. Players, being rational, optimize utilities. All players commonly understand that information available is partial and fragmented by nature, but they nonetheless dynamically update strategies with reselection and mutation to find niches of high utility. These dynamics of individual players contribute to a population’s exploration of strategy space (e) displays a network of activated strategies within a population).



a Not dissimilar to traditional media being accused of producing “fake news.”
 b The zero-sum equity establishes the conflict and incentivizes a Sybil attack.


pressed and tested for their ability to destabilize a desired equilibrium.

To accommodate this, we encode change to the population over time (for example, by invasion of mutants) over repeated games by using deterministic finite automata (DFA). The DFA strategy space offers a vastly reachable space of dynamic strategic structures. This provides the means to explore the uses of identity in repeated signaling interactions.


The DFA state codes noted in the table determine the (type, signal) of a sender's controlling agent, or the action as receiver. Each DFA encounter determines a sequence of outcomes as illustrated in the example that follows. Consider the strategy of Figure 3(c) as sender matched against strategy of (d) as receiver with a transaction budget of two units. The sender starts in state s_1 , and the receiver starts in state s_3 ; they play at the cost of one unit against the transaction budget. Note that the discount for deception will entail additional communication efforts. Next, the sender transitions to state s_7 by following the s_3 labeled transition, and the receiver loops back to state s_3 ; they both play at the cost of a half unit since state s_7 uses deception. Next, the sender transitions to state s_1 while the receiver transitions to state s_6 to exhaust the transaction budget and complete the game. The computed outcome sequence is o_1, o_7, o_2 , resulting in a sender aggregate utility of $(A + B)$ and receiver aggregate utility of $(B - (A + C))$.

Evolutionary strategy. Evolutionary game theory models a dynamic population of agents capable of modifying their strategy and predicts population-level effects.^{2,3,5,19,32} Formally, evolutionary games are a dynamic system with stochastic variables. The agents in evolutionary games may (both individually and collectively) explore strategy structures directly (via mutation and peer-informed reselection), and they may exploit strategies where and when competitive advantages are found.

To implement this system, the time domain is divided into intervals called *generations*. The system is initialized by fixing a finite set of agents and assigning each agent a strategy determined with a *seeding probability distribution*.



When a deceptive identity succeeds, it will be used numerous times as there is no reason to abandon it after one interaction. Moreover, it is precisely the repeated interactions that are needed to develop trust.



During a generation, pairs of agents will encounter one another to play repeated signaling games; the encounters are determined by an *encounter distribution*. At the completion of a generation, agents evaluate rewards obtained from their implemented strategies. This evaluation results in their *performance measure*. Next, performance measures are compared within a set of peer agents that cooperate to inform each agents' reselection stage. During the reselection stage, agents determine a strategy to use in the next generation, as achieved by a *boosting probability distribution* that preferentially selects strategies based on performance. After reselection, some agents are mutated with a *mutation probability distribution*. This step completes the generation and establishes the strategies implemented during the next generation.

The agents evolve discrete strategic forms (DFA); a strategic mutation network is graphed in Figure 3(e) to provide a sense of scale. The dynamic system thus evolves a population measure over strategies. Within the WANET, nodes freely mutate, forming deceptive strategies as often as they augment cooperative ones. Evolutionary games allow us to elucidate the stability and resilience of various strategies arising from mutations and a selection process ruled by non-cooperation and rationality.

We augment the basic structure of reselection by considering carefully how strategic information is shared. Upon noticing that deceptive and cooperative strategies differ fundamentally in their information asymmetric requirements, we introduce a technique referred to as split-boosting, which modulates the information flow components of the network.

Recreate by split-boosting. During the *Recreate* phase, agents select strategies preferentially by comparing performance measured only among a set of agents that share this pooled information.

Splitting the set of agents into components we limit the boosting to include only strategies available from the component. Within a component (subset) S , let v_i be the performance measure for strategy used by agents $i \in S$. Letting $v_* = \min_{i \in S} \{v_i\}$ and $v^* = \max_{i \in S} \{v_i\}$ we can safely transfer the performance measures to the

interval $[0, 1]$ as the limit of fractional transformation:

$$V_i^\xi = \lim_{\eta \rightarrow 0^+} \frac{v_i + (\xi - v_*)}{v^* - (v_* - \eta)}$$

The term η simply prevents division by zero, and the term ξ is a *statistical shrinkage* term used as a model parameter that helps to distort global information available to agents when they reselect a strategy.

We describe the probability that agent $i \in S$ switches over to use the strategy that agent $j \in S$ previously implemented as $\phi_j := V_j^\xi / \sum_{k \in S} V_k^\xi$.

Results

Under the signaling game theoretic model, we evaluate equilibrium concepts and their stability under evolutionary dynamics including mutant Sybil identities. We further specify the WANET case and its parameters to perform computer simulations yielding empirical measures of its behavior. Here, we focus on how validated and shared security information can ballast the desired equilibrium of honest signaling.

Models and simulations. To demonstrate simulation scalability, we used a laptop (with a 2GHz Intel core i7 processor and 8GB of RAM) to measure a simulation history (with 800 nodes and 1,000 generations). In eight minutes of user time over 16M rounds of play, 160K strategic mutations were explored; 125K of those mutations were found to be unique DFA strategy structures, and 36K employed deceptive identities. It was possible to discover a stable equilibrium where all agents reveal their identity honestly and act with the common knowledge of others revealing their identities honestly. Since mutating into a Sybil behavior is detectable by others and credibly punishable, the equilibrium is stable. Note also that the nature of cyber-social systems makes these systems amenable to empirical evolutionary studies in that model checking or other formal approaches would require “an intelligent designer” who could specify various global properties of the system. However, we do not rule out a role for statistical model checking in this and other similar mechanism design studies.

Experiments and empirical analysis. Our experiments consider a simple

Defining Deception

To avoid undesirable outcomes arising from deception, we call upon a theory of information-asymmetric signaling games to unify many of the adversarial use cases under a single framework, in particular when adversarial actions may be viewed mathematically as rational (that is, utility-optimizing agents possessing common knowledge of rationality).

The simplest model of signaling games involves two players. They are asymmetric in information and are called S , sender (informed), and R , receiver (uninformed). A key notion in this game is that of type, a random variable whose support is given by T (known to sender S). Also, we use $\pi_T(\cdot)$ to denote probability distribution over T as a prior belief of R about the sender's type. A round of game play proceeds as follows: Player S learns $t \in T$; S sends to R a signal $s \in M$; and R takes an action $a \in A$. Their payoff/utility functions are known and depend on the type, signal, and action:

$$u^i : T \times M \times A \rightarrow \mathbb{R} : i \in \{S, R\}. \quad (1)$$

In this structure, the players' behavior strategies can be described by the following two sets of probability distributions: (1) $\mu(\cdot|t)$, $t \in T$, on M and (2) $\alpha(\cdot|s)$, $s \in M$, on A . For S , the sender strategy μ is a probability distribution on signals given types; namely, $\mu(s|t)$ describes the probability that S with type t sends signal s . For R , the receiver strategy α is a probability distribution on actions given signals; namely, $\alpha(a|s)$ describes the probability that R takes action a following signal s . A pair of strategies μ and α is in Nash equilibrium if (and only if) they are mutually best responses (that is, if each maximizes the expected utility given the other):

$$\begin{aligned} \sum_{t \in T, s \in M, a \in A} u^S(t, s, a) \pi_T(t) \mu^*(s|t) \alpha(a|s) \\ \geq \sum_{t \in T, s \in M, a \in A} u^S(t, s, a) \pi_T(t) \mu(s|t) \alpha(a|s) \end{aligned} \quad (2)$$

and

$$\begin{aligned} \sum_{t \in T, s \in M, a \in A} u^R(t, s, a) \pi_T(t) \mu(s|t) \alpha^*(a|s) \\ \geq \sum_{t \in T, s \in M, a \in A} u^R(t, s, a) \pi_T(t) \mu(s|t) \alpha(a|s) \end{aligned} \quad (3)$$

for any μ, α . It is straightforward to show that such a strategy profile (α^*, μ^*) exists. We conjecture that the natural models for sender-receiver utility functions could be based on functions that combine information rates with distortion, as in rate distortion theory (RDT). For instance, assume there are certain natural connections between the types and actions, as modeled by the functions f_S and f_R for the sender and receiver respectively:

$$f_S : T \rightarrow A; \quad f_R : A \rightarrow T. \quad (4)$$

Then the utility function for each consists of two weighted-additive terms, one measuring the mutual information with respect to the signals and the other measuring the undesirable distortion, where the weights are suitably chosen Lagrange constants

$$u^S = I(T, M) - \lambda_S d^S(f_S(t), a), \quad \& \quad (5)$$

$$u^R = I(A, M) - \lambda_R d^R(t, f_R(a)),$$

where I denotes mutual information and d^R, d^S denote measures of distortion.

This definition also captures the notion of deception as follows. Thus the distribution of signals received by R is given by the probability distribution π_M , where

$$\pi_M(s) = \sum_{t \in T} \pi_T(t) \mu(s|t), \quad (6)$$

and the distribution of actions produced by R is given by the probability distribution π_A , where

$$\pi_A(a) = \sum_{s \in M} \pi_M(s) \alpha(a|s). \quad (7)$$

Clearly π_T and π_A are probability distributions on T and A respectively.

If $\hat{\pi}_T$ is the probability distribution on T induced by π_A under the function f_R , then

$$\hat{\pi}_T(\cdot) := \pi_A(f_R^{-1}(\cdot)). \quad (8)$$

A natural choice of measure for deception is given by the relative entropy between the probability distributions π_T and $\hat{\pi}_T$:

$$\begin{aligned} \text{Deception} &:= \text{Rel. Entropy}(\hat{\pi}_T | \pi_T) \\ &= \sum_{t \in T} \hat{\pi}_T(t) \log_2 \frac{\hat{\pi}_T(t)}{\pi_T(t)}. \end{aligned} \quad (9)$$

This definition describes deception from the point of view of the receiver. To get the notion of deception from the point of view of the sender, one needs to play the game several rounds.

setting to illustrate the intuition that costly signaling and verified information flows among cooperative types can stabilize behavior in WANETs. More generally simulations (as a computational technique) can evaluate a variety of mechanisms and how they influence system behaviors.

Our major control in experiments examines how differing information pooling for cooperative vs. deceptive types leads to differing qualitative behavior outcomes. We consider a reference system S_0 and reengineer it with a device to express improved information pooling among cooperative types to create alternate system S_1 . The systems feature the same competitive pressures and are identical in every way except in their implementation of the reselection step. Game parameters are $A, B, C, D = 4, 0.5, 0.5, 4.0$, with 800 network nodes and 400 generations. In both systems, the same seeding distribution initializes the simulations from a state where no nodes employ (immediately) deceptive or Sybil identities. From these initial conditions, mutations allow nodes to quickly use deceptive strategies and test their efficacy.

In the first system S_0 , all agents select strategies using common and

identical information pooling. Therefore, both cooperative and deceptive types are treated alike, specifically with the same awareness to and distortions of pooled information guiding strategic exploration.

In the second system S_1 , agents select strategy with boosting split by type. Strategic information, once verified as cooperative, is offered to all agents with an openly shared common database of clean strategies. This modification enhances information for cooperative types while conversely imposing isolating effects for deceptive types. Also, in our simulations, the deceptive types maintain rationality, so when a deceptive strategy is found to be performing poorly (less than the cooperative group average), the agents abandon the deceptive strategy as being non-productive, thereby coming clean and reselecting strategies from the shared database as the best survival option.

In Figure 4 we show typical simulated traces for systems S_0 and S_1 plotting the proportion of population employing deceptive strategies (a crude estimation of deception as defined in the sidebar “Defining Deception”). The differing properties for information flows affecting reselection

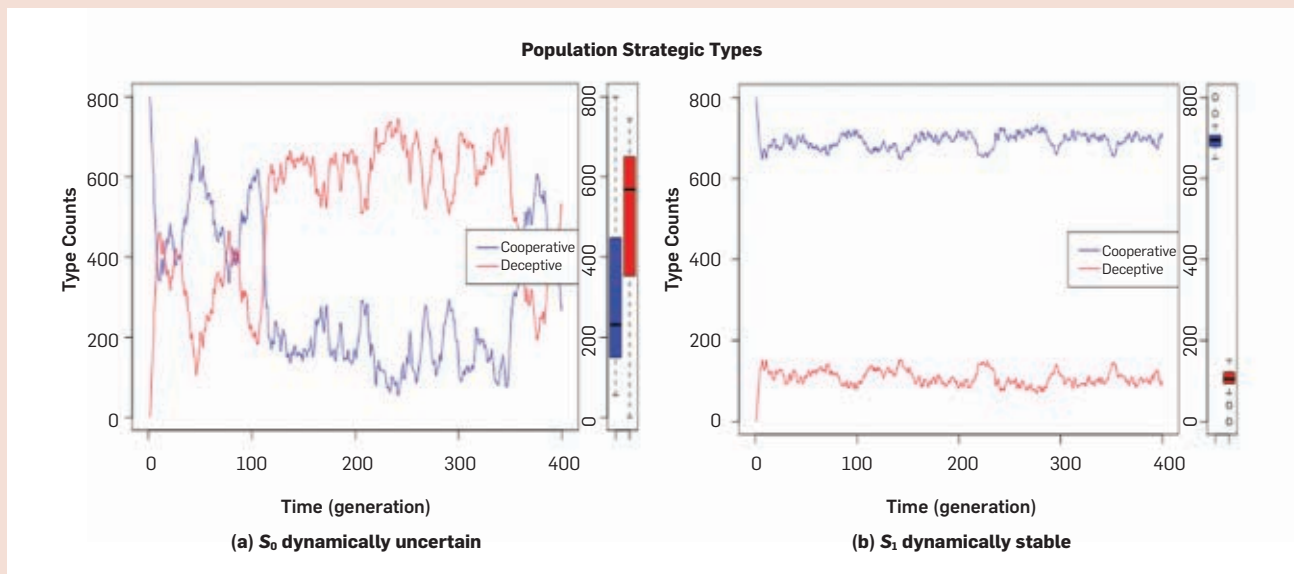
offer strong controls to stabilize the dynamic equilibrium favorable to cooperators. In S_1 the advantages of deception are short-lived, and cooperative behaviors are promoted even when agents remain free to explore for niche use of deception.

Conclusion and Future Work

Several insights and contributions emerge from our experiments. One key insight is that challenging an agent in such a way that deceptive agents either fail the challenge or face greater risk can deter deception. Another key insight is that many instances where agents use deceptive identities in cyber-social systems are repeated games. When a deceptive identity succeeds, it will be used numerous times as there is no reason to abandon it after one interaction. Moreover, it is precisely the repeated interactions that are needed to develop trust. Thus, formalizing these insights we devised a mathematical game to model strategic interactions, while recognizing a possibility of permissive and malleable identities. With the dilemma between privacy and intent clarified formally in signaling games, we computationally considered various strategies such as those based in behavior learn-

Figure 4. Results.

For cyber-social systems, we can use simulation to study a variety of equilibria (or lack thereof) affected by various mechanisms. Here, and with few additional assumptions concerning Sybil attackers, the effects of using a shared database of verified cooperative strategic forms is shown to deter deceptive types (b) in contrast to instances where no such advantage is given to cooperative strategic forms (a). The x-axis represents the temporal dimension (generations), the blue graph and quantile figure represents the proportion of population using honest identity signaling, the red otherwise.




ing and costly signaling. Our computational simulations uncovered several interesting information flow properties that may be leveraged to deter deception, specifically by enhancing the flow of information regarding cooperative strategies while reinforcing the cooperative group's identity. Interestingly, this result indicates an identity management system, typically thought to hinge on the precision of true positives and astronomical unlikeliness of false-positive recognition, may rather critically depend on how learned behavior and strategic information can be shared.

Our computational experiment offers new insights for achieving strong deterrence of identity deception within ad hoc networks such as WANETs, however much is left as future work. Our larger practical goal is M-coin, a design strategy and system for cooperation enhancing technologies. M-coin may be thought of as an abstract currency guiding an open recommender-verification system that incorporates new agent types (to verify identities, behavior histories, and cooperative strategies as well as the consistency of distrusted information); the new types promote efficiencies supporting cooperative coalitions. The main step forward, as demonstrated here, is recognizing the effects of pooled and verified strategic information, and its flow constraints (as well as its capabilities to operate in the open). Vetted strategic information assists cooperators to rapidly adapt to and out-compete deceptive strategies.

Still, many challenges remain outstanding. The possibility of an agent not compelled by utility presents a problem, as that agent may persist within the network indefinitely to form effective attacks. Future work may focus on how the expression of rationality could be fortified for identities/nodes. Critically, deceptively minded actors will need to prefer a base level of utility, and this remains an open challenge (although the solution could lie in the many possibilities suggested by biological systems). Additionally, technologies supporting the tedious aspects of information gathering and validation must be aligned to user incentives.

Properly constructed recommender-verifier architectures could be used in

WANETs, HFNs, and other fluid-identity cyber-social and cyber-physical systems to reliably verify private but trustworthy identities and limit the damage of deceptive attack strategies. Starting with WANETs, we motivate an elegant solution using formalisms we originally developed for signaling games. Nonetheless, we are encouraged by analogous biological solutions derived naturally under Darwinian evolution.

Acknowledgments. We thank the anonymous reviewers for their insightful comments. This material is based upon work funded and supported by U.S. Department of Defense Contract No. FA8702-15-D-0002 with Carnegie Mellon University Software Engineering Institute and New York University and ARO grant A18-0613-00 (B.M.). This material has been approved for public release and unlimited distribution, ref DM17-0409. 

References

- Argiento R., Pemantle, R., Skyrms, B. and Volkov, S. Learning to signal: Analysis of a micro-level reinforcement model. *Stochastic Processes and their Applications* 119, 2 (2009), 373–390.
- Axelrod, R. An evolutionary approach to norms. *American Political Science Review* 80, 4 (1986), 1095–1111.
- Axelrod, R. *The Evolution of Cooperation*. Basic books, 2006.
- Banks, J. and Sobel, J. Equilibrium selection in signaling games. *Econometrica: J. Econometric Society*, (1987), 647–661.
- Binmore, K. and Samuelson, L. Evolutionary stability in repeated games played by finite automata. *J. Economic Theory* 57, 2 (1992), 278–305.
- Casey, W., Memarmoshrefi, P., Kellner, A., Morales, J.A. and Mishra, B. Identity deception and game deterrence via signaling games. In *Proceedings of the 9th EAI Intern. Conf. Bio-inspired Information and Communications Technologies*, 73–82.
- Casey, W., Morales, J.A. and Mishra, B. Threats from inside: Dynamic utility (mis) alignments in an agent-based model. *J. Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* 7 (2016), 97–117.
- Casey, W., Morales, J.A., Nguyen, T., Spring, J., Weaver, R., Wright, E., Metcalfe, L. and Mishra, B. Cyber security via signaling games: Toward a science of cyber security. In *Proceedings of the Intern. Conf. Distributed Computing and Internet Technology*, 34–42.
- Casey, W., Morales, J.A., Wright, E., Zhu, Q. and Mishra, B. Compliance signaling games: Toward modeling the deterrence of insider threats. *Computational and Mathematical Organization Theory* 22, 3 (2016), 318–349.
- Casey, W., Weaver, R., Morales, J.A., Wright, E. and Mishra, B. Epistatic signaling and minority games, the adversarial dynamics in social technological systems. *Mobile Networks and Applications* 21, 1 (2016), 161–174.
- Casey, W., Wright, E., Morales, J.A., Appel, M., Gennari, J. and Mishra, B. Agent-based trace learning in a recommendation verification system for cybersecurity. In *Proceedings of the 9th IEEE Intern. Conf. on Malicious and Unwanted Software: The Americas*, (2014), 135–143.
- Casey, W., Zhu, Q., Morales, J.A. and Mishra, B. Compliance control: Managed vulnerability surface in social-technological systems via signaling games. In *Proceedings of the 7th ACM CCS Intern. Workshop on Managing Insider Security Threats*, (2015), 53–62.
- Catteeuw, D., Manderick, B. et al. Evolution of honest signaling by social punishment. In *Proceedings of the 2014 Annual Conf. Genetic and Evolutionary Computation*, (2014), 153–160.
- Cho, I.-K. and Sobel, J. Strategic stability and uniqueness in signaling games. *J. Economic Theory* 50, 2 (1990), 381–413.
- Chung, H. and Carroll, S.B. Wax, sex and the origin of species: Dual roles of insect cuticular hydrocarbons in adaptation and mating. *BioEssays*, (2015).
- Daskalakis, C., Goldberg, P.W. and Papadimitriou, C.H. The complexity of computing a Nash equilibrium. *SIAM J. Computing* 39, 1 (2009), 195–259.
- Fabrikant, A., Papadimitriou, C. and Talwar, K. The complexity of pure Nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, (2004), 604–612.
- Hamblin, S. and Hurd, P.L. When will evolution lead to deceptive signaling in the Sir Philip Sidney game? *Theoretical Population Biology* 75, 2 (2009), 176–182.
- Huttenberger, S.M., Skyrms, B., Smead, R. and Zollman, K.J.S. Evolutionary dynamics of Lewis signaling games: Signaling systems vs. partial pooling. *Synthese* 172, 1 (2010), 177–191.
- Jee, J., Sundstrom, A., Massey, S.E. and Mishra, B. What can information-asymmetric games tell us about the context of Crick's 'frozen accident'? *J. the Royal Society Interface* 10, 88 (2013).
- King, D. The Haiti earthquake: Breaking new ground in the humanitarian information landscape. *Humanitarian Exchange Magazine* 48, (2010).
- Lewis, D. *Convention: A Philosophical Study*. John Wiley & Sons, 2008.
- Nash, J. Non-cooperative games. *Annals of Mathematics*, (1951), 286–295.
- Nash, J. et al. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences* 36, 1 (1950), 48–49.
- Nash, J.F. Jr. The bargaining problem. *Econometrica: J. Econometric Society*, (1950), 155–162.
- Newsome, J., Shi, E., Song, D. and Perrig, A. The Sybil attack in sensor networks: Analysis & defenses. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, (2004), 259–268.
- Papadimitriou, C. Algorithms, Games, and the Internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, (2001), 749–753.
- Sharma, K.R., Enzmann, B.L. et al. Cuticular Hydrocarbon pheromones for social behavior and their coding in the ant antenna. *Cell Reports* 12, 8 (2015), 1261–1271.
- Silk, J.B., Kaldor, E., and Boyd, R. Cheap talk when interests conflict. *Animal Behavior* 59, 2 (2000), 423–432.
- Skyrms, B. *The Stag Hunt and the Evolution of Social Structure*. Cambridge University Press, 2004.
- Skyrms, B. *Signals: Evolution, Learning, and Information*. Oxford University Press, 2010.
- Smith, J.M. *Evolution and the Theory of Games*. Cambridge University Press, 1982.
- Smith, J.M. Honest signaling: The Philip Sidney game. *Animal Behaviour* 42, 6 (1991), 1034–1035.
- Sobel, M.J. et al. Non-cooperative stochastic games. *The Annals of Mathematical Statistics* 42, 6 (1971), 1930–1935.
- Neumann, J.V. and Morgenstern, O. *Theory of Games and Economic Behavior*. Princeton University Press, 2007.
- Zollman, K.J.S., Bergstrom, C.T., and Huttenberger, S.M. Between cheap and costly signals: The evolution of partially honest communication. In *Proceedings of the Royal Society of London B: Biological Sciences*, (2012).

William Casey (wcasey@cmu.edu) is a senior member of Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, USA.

Ansgar Kellner is a research fellow at the Institute of System Security at Technische Universität Braunschweig, Germany.

Parisa Memarmoshrefi is a research staff member at University of Göttingen, Germany.

Jose Andre Morales is a researcher at the Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Bud Mishra (mishra@nyu.edu) is a professor at New York University Courant Institute, Tandon School of Engineering and School of Medicine, New York, NY, USA.

research highlights

P. 95

Technical Perspective **Photorealistic Facial Digitization and Manipulation**

By Hao Li

P. 96

Face2Face: Real-Time Face Capture and Reenactment of RGB Videos

By Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner

P. 105

Technical Perspective **Attacking Cryptographic Key Exchange with Precomputation**

By Dan Boneh

P. 106

Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice

By David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann

Technical Perspective

Photorealistic Facial Digitization and Manipulation

By Hao Li

FOR MORE THAN a decade, computer graphics (CG) researchers and visual effects experts have been fascinated with bringing photorealistic digital actors to the screen. Crossing the well-known “uncanny valley” in CG humans has been one of the most difficult and crucial challenges, due to hypersensitivity to synthetic humans lacking even the slightest and most subtle features of genuine human faces. Given sufficient resources and time, photorealistic renderings of digital characters have been achieved in recent years. Some of the most memorable cases are seen in blockbuster movies, such as *The Curious Case of Benjamin Button*, *Furious 7*, and *Rogue One: A Star Wars Story*, in which large teams of highly skilled digital artists use cutting-edge digitization technologies. Despite the progress of 3D-scanning solutions, facial animation systems, and advanced rendering techniques, weeks of manual work are still needed to produce even just a few seconds of animation.

When depth cameras, such as structured light systems or time-of-flight sensors, were introduced, the 3D acquisition of highly deformable surfaces became possible. Graphics and vision researchers started to investigate the possibility of directly capturing complex facial performances, instead of manually key-framing them or applying complex simulations. While marker-based motion capture technologies are already widely adopted in industry, massive amounts of hand-tweaking and post-processing are still needed to generate lifelike facial movements. On the other hand, markerless solutions based on real-time RGB-D sensors provide dense and accurate facial shape measurements and were poised to automate and scale animation production.


The release of the mainstream Kinect depth sensor from Microsoft sparked a great deal of interest in real-time facial animation in the con-

sumer space, most notably through several seminal SIGGRAPH publications between 2010 and 2013, as well as the popular facial animation software, Faceshift, later acquired by Apple. While computer vision-based facial landmark detectors are suitable for puppeteering CG faces using conventional RGB cameras, they do not capture nuanced facial expressions, as only sparse features are tracked. However, when dense depth measurements are available, an accurate 3D face model can be computed by refining the shape of a statistical face model to fit a dense input depth map. Not only can this face-fitting problem be solved in real time using efficient numerical optimization, but the shape and expression parameters of the face can be fully recovered and used for retargeting purposes. If facial performance capture is possible for conventional RGB videos in real time, then believable facial expressions can be transferred effortlessly from one person to another in a live-action scenario. This capability is demonstrated by the Face2Face system of Thies et al. detailed in the following paper.

As opposed to animating a CG character in a virtual environment, the key challenge is to produce a photorealistic video of a target subject whose facial performance matches the source actor. In addition to being able to track and transfer dense facial movements at the pixel level, the facial albedo and lighting environment also must be estimated on the target video, in order to ensure a consistent shading with the original footage. The solution consists of a real-time GPU implementation of a photometric consistency optimization that solves for parameters of a morphable face model originally introduced by Blanz and Vetter, extended with linear facial expression blendshapes. The authors also introduce an important data-driven technique to handle the non-lin-

ear appearance deformations of the mouth, in which plausible textures are retrieved instead of being rendered using a parametric model. Such an approach is particularly effective in producing a photorealistic output, as it bypasses the traditional and more complex rendering pipeline. While some limitations remain, such as the inability to control the head pose in the target video sequence, very convincing photorealistic facial reenactments are demonstrated on footages of celebrities and politicians obtained from YouTube.

While the original intent of performance-driven video was to advance immersive communication, teleconferencing, and visual effects, the ease and speed with which believable manipulations can be created with such technology has garnered widespread media attention, and raised concerns about the authenticity and ethical aspects of artificially generated videos.

Recent progress in artificial intelligence, such as deep generative models, is further accelerating these capabilities and making them even easier for ordinary people to use. For instance, Pinscreen’s photorealistic avatar creation technology requires only a single input picture and can be used to create compelling video game characters at scale, but face replacement technologies, such as DeepFake, have been exploited to create inappropriate and misleading video content. I highly recommend the following paper, as it is one of the first that promotes awareness of modern technology’s capability to manipulate videos, at a time in which social media is susceptible to the spread of doctored videos and fake news. 

Hao Li (hao@hao-li.com) is assistant professor of computer science at the University of Southern California, director of the Vision and Graphics Lab of the USC Institute for Creative technologies, and CEO of Pinscreen.

Copyright held by author/owner.

Face2Face: Real-Time Face Capture and Reenactment of RGB Videos

By Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner

Abstract

Face2Face is an approach for real-time facial reenactment of a monocular target video sequence (e.g., Youtube video). The source sequence is also a monocular video stream, captured live with a commodity webcam. Our goal is to animate the facial expressions of the target video by a source actor and re-render the manipulated output video in a photo-realistic fashion. To this end, we first address the under-constrained problem of facial identity recovery from monocular video by non-rigid model-based bundling. At run time, we track facial expressions of both source and target video using a dense photometric consistency measure. Reenactment is then achieved by fast and efficient deformation transfer between source and target. The mouth interior that best matches the re-targeted expression is retrieved from the target sequence and warped to produce an accurate fit. Finally, we convincingly re-render the synthesized target face on top of the corresponding video stream such that it seamlessly blends with the real-world illumination. We demonstrate our method in a live setup, where Youtube videos are reenacted in real time. This live setup has also been shown at SIGGRAPH Emerging Technologies 2016, by Thies et al.²⁰ where it won the Best in Show Award.

1. INTRODUCTION

In recent years, real-time markerless facial performance capture based on commodity sensors has been demonstrated. Impressive results have been achieved, both based on Red-Green-Blue (RGB) as well as RGB-D data. These techniques have become increasingly popular for the animation of virtual Computer Graphics (CG) avatars in video games and movies. It is now feasible to run these face capture and tracking algorithms from home, which is the foundation for many Virtual Reality (VR) and Augmented Reality (AR) applications, such as teleconferencing.

In this paper, we employ a new dense markerless facial performance capture method based on monocular RGB data, similar to state-of-the-art methods. However, instead of transferring facial expressions to virtual CG characters, our main contribution is monocular *facial reenactment* in real-time. In contrast to previous reenactment approaches that run offline, our goal is the *online* transfer of facial expressions of a source actor captured by an RGB sensor to a target actor. The target sequence can be any monocular video; for example, legacy video footage downloaded from Youtube with a facial performance. We aim to modify the target video in a

photo-realistic fashion, such that it is virtually impossible to notice the manipulations. Faithful photo-realistic facial reenactment is the foundation for a variety of applications; for instance, in video conferencing, the video feed can be adapted to match the face motion of a translator, or face videos can be convincingly dubbed to a foreign language.

In our method, we first reconstruct the shape identity of the target actor using a new global non-rigid model-based bundling approach based on a prerecorded training sequence. As this preprocess is performed globally on a set of training frames, we can resolve geometric ambiguities common to monocular reconstruction. At run-time, we track both the expressions of the source and target actor's video by a dense analysis-by-synthesis approach based on a statistical facial prior. We demonstrate that our RGB tracking accuracy is on par with the state of the art, even with online tracking methods relying on depth data. In order to transfer expressions from the source to the target actor in real-time, we propose a novel transfer functions that efficiently applies deformation transfer¹⁸ directly in the used low-dimensional expression space. For final image synthesis, we re-render the target's face with transferred expression coefficients and composite it with the target video's background under consideration of the estimated environment lighting. Finally, we introduce a new image-based mouth synthesis approach that generates a realistic mouth interior by retrieving and warping best matching mouth shapes from the offline sample sequence. It is important to note that we maintain the appearance of the target mouth shape; in contrast, existing methods either copy the source mouth region onto the target²³ or a generic teeth proxy is rendered,^{8, 19} both of which leads to inconsistent results. Figure 2 shows an overview of our method.

We demonstrate highly convincing transfer of facial expressions from a source to a target video in real time. We show results with a live setup where a source video stream, which is captured by a webcam, is used to manipulate a target Youtube video (see Figure 1). In addition, we compare against state-of-the-art reenactment methods, which we outperform both in terms of resulting video quality and

The original version of this paper was published in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2016, IEEE.

Figure 1. Proposed online reenactment setup: A monocular target video sequence (e.g., from Youtube) is reenacted based on the expressions of a source actor who is recorded live with a commodity webcam.

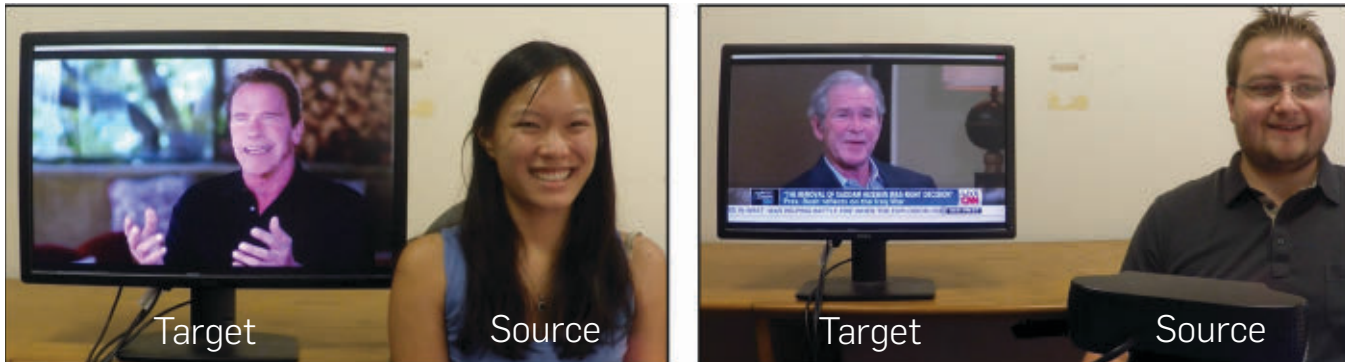
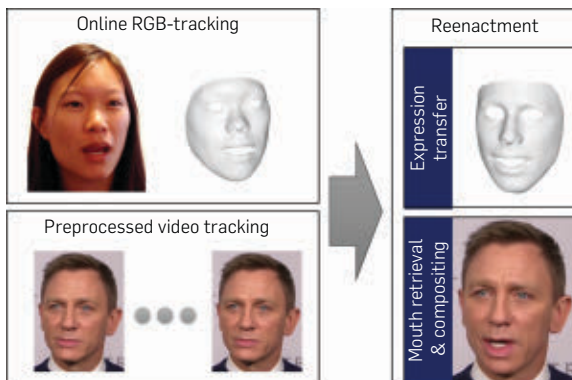


Figure 2. An overview of our reenactment approach: In a preprocessing step we analyze and reconstruct the face of the target actor. During live reenactment, we track the expression of the source actor and transfer them to the reconstructed target face. Finally, we composite a novel image of the target person using a mouth interior of the target sequence that best matches the new expression.



runtime (we are the first real-time RGB reenactment method). In summary, our key contributions are:

- dense, global non-rigid model-based bundling,
- accurate tracking, appearance, and lighting estimation in unconstrained live RGB video,
- person-dependent expression transfer using subspace deformations,
- and a novel mouth synthesis approach.

2. RELATED WORK

2.1. Offline RGB performance capture

Recent offline performance capture techniques approach the hard monocular reconstruction problem by fitting a blendshape or a multilinear face model to the input video sequence. Even geometric fine-scale surface detail is extracted via inverse shading-based surface refinement. Shi et al.¹⁶ achieve impressive results based on global energy optimization of a set of selected keyframes. Our model-based bundling formulation to recover actor identities is similar to their approach; however, we use robust and dense global photometric alignment, which we enforce with an

efficient data-parallel optimization strategy on the Graphics Processing Unit (GPU).

2.2. Online RGB-D performance capture

Weise et al.²⁵ capture facial performances in real-time by fitting a parametric blendshape model to RGB-D data, but they require a professional, custom capture setup. The first real-time facial performance capture system based on a commodity depth sensor has been demonstrated by Weise et al.²⁴ Follow up work focused on corrective shapes,² dynamically adapting the blend-shape basis,¹¹ non-rigid mesh deformation.⁶ These works achieve impressive results, but rely on depth data which is typically unavailable in most video footage.

2.3. Online RGB performance capture

While many sparse real-time face trackers exist, for example, Saragih et al.,¹⁵ real-time dense monocular tracking is the basis of realistic online facial reenactment. Cao et al.⁵ propose a real-time regression-based approach to infer 3D positions of facial landmarks which constrain a user-specific blendshape model. Follow-up work⁴ also regresses fine-scale face wrinkles. These methods achieve impressive results, but are not directly applicable as a component in facial reenactment, since they do not facilitate dense, pixel-accurate tracking.

2.4. Offline reenactment

Vlasic et al.²³ perform facial reenactment by tracking a face template, which is re-rendered under different expression parameters on top of the target; the mouth interior is directly copied from the source video. Image-based offline mouth re-animation was shown in Bregler et al.³ Garrido et al.⁷ propose an automatic purely image-based approach to replace the entire face. These approaches merely enable self-reenactment; that is, when source and target are the same person; in contrast, we perform reenactment of a different target actor. Recent work presents virtual dubbing,⁸ a problem similar to ours; however, the method runs at slow offline rates and relies on a generic teeth proxy for the mouth interior. Li et al.¹² retrieve frames from a database based on a similarity metric. They use optical flow as appearance and

velocity measure and search for the k -nearest neighbors based on time stamps and flow distance. Saragih et al.¹⁵ present a real-time avatar animation system from a single image. Their approach is based on sparse landmark tracking, and the mouth of the source is copied to the target using texture warping.

2.5. Online reenactment

Recently, first online facial reenactment approaches based on RGB-(D) data have been proposed. Kemelmacher-Shlizerman et al.¹⁰ enable image-based puppetry by querying similar images from a database. They employ an appearance cost metric and consider rotation angular distance. While they achieve impressive results, the retrieved stream of faces is not temporally coherent. Thies et al.¹⁹ show the first online reenactment system; however, they rely on depth data and use a generic teeth proxy for the mouth region. In this paper, we address both shortcomings: (1) our method is the first real-time RGB-only reenactment technique; (2) we synthesize the mouth regions exclusively from the target sequence (no need for a teeth proxy or direct source-to-target copy).

2.6. Follow-up work

The core component of the proposed approach is the dense face reconstruction algorithm. It has already been adapted for several applications, such as head mounted display removal,²² facial projection mapping,¹⁷ and avatar digitization.⁹ FaceVR²² demonstrates self-reenactment for head mounted display removal, which is particularly useful for enabling natural teleconferences in virtual reality. The FaceForge¹⁷ system enables real-time facial projection mapping to dynamically alter the appearance of a person in the real world. The avatar digitization approach of Hu et al.⁹ reconstructs a stylized 3D avatar that includes hair and teeth, from just a single image. The resulting 3D avatars can for example be used in computer games.

3. USE CASES

The proposed facial tracking and reenactment has several use-cases that we want to highlight in this section. In movie productions the idea of facial reenactment can be used as a video editing tool to change for example the expression of an actor in a particular shot. Using the estimated geometry of an actor, it can also be used to modify the appearance of a face in a post-process, for example, changing the illumination. Another field in post-production is the synchronization of an audio channel to the video. If a movie is translated to another language, the movements of the mouth do not match the audio of the so called dubber. Nowadays, to match the video, the audio including the spoken text is adapted, which might result in a loss of information. Using facial reenactment instead, the expressions of the dubber can be transferred to the actor in the movie and thus the audio and video is synchronized. Since our reenactment approach runs in real time, it is also possible to setup a teleconferencing system with a live interpreter that simultaneously translates the speech of a person to another language.

In contrast to state-of-the-art movie production setups that work with markers and complex camera setups, our system presented in this paper only requires commodity hardware without the need for markers. Our tracking results can also be used to animate virtual characters. These virtual characters can be part of animation movies, but can also be used in computer games. With the introduction of virtual reality glasses, also called Head Mounted Displays (HMDs), the realistic animation of such virtual avatars, becomes more and more important for an immersive game-play. FaceVR²² demonstrates that facial tracking is also possible if the face is almost completely occluded by such an HMD. The project also paves the way to new applications like teleconferencing in VR based on HMD removal.

Besides these consumer applications, you can also think of numerous medical applications. For example, one can build a training system that helps patients to train expressions after a stroke.

4. METHOD OVERVIEW

In the following, we describe our real-time facial reenactment pipeline (see Figure 2). Input to our method is a monocular target video sequence and a live video stream captured by a commodity webcam. First, we describe how we synthesize facial imagery using a statistical prior and an image formation model (see Section 5). We find optimal parameters that best explain the input observations by solving a variational energy minimization problem (see Section 6). We minimize this energy with a tailored, data-parallel GPU-based Iteratively Reweighted Least Squares (IRLS) solver (see Section 7). We employ IRLS for off-line non-rigid model-based bundling (see Section 8) on a set of selected keyframes to obtain the facial identity of the source as well as of the target actor. This step jointly recovers the facial identity, expression, skin reflectance, and illumination from monocular input data. At runtime, both source and target animations are reconstructed based on a model-to-frame tracking strategy with a similar energy formulation. For reenactment, we propose a fast and efficient deformation transfer approach that directly operates in the subspace spanned by the used statistical prior (see Section 9). The mouth interior that best matches the re-targeted expression is retrieved from the input target sequence (see Section 10) and is warped to produce an accurate fit. We demonstrate our complete pipeline in a live reenactment setup that enables the modification of arbitrary video footage and perform a comparison to state-of-the-art tracking as well as reenactment approaches (see Section 11). In Section 12, we show the limitations of our proposed method.

Since we are aware of the implications of a video editing tool like Face2Face, we included a section in this paper that discusses the potential misuse of the presented technology (see Section 13). Finally, we conclude with an outlook on future work (see Section 14).

5. SYNTHESIS OF FACIAL IMAGERY

The synthesis of facial imagery is based on a multi-linear face model (see the original Face2Face paper for more details). The first two dimensions represent facial identity—that is,

geometric shape and skin reflectance—and the third dimension controls the facial expression. Hence, we parametrize a face as:

$$\mathcal{M}_{\text{geo}}(\alpha, \delta) = \mathbf{a}_{\text{id}} + E_{\text{id}} \cdot \alpha + E_{\text{exp}} \quad (1)$$

$$\mathcal{M}_{\text{alb}}(\beta) = \mathbf{a}_{\text{alb}} + E_{\text{alb}} \cdot \beta. \quad (2)$$

This prior assumes a multivariate normal probability distribution of shape and reflectance around the average shape $\mathbf{a}_{\text{id}} \in \mathbb{R}^{3n}$ and reflectance $\mathbf{a}_{\text{alb}} \in \mathbb{R}^{3n}$. The shape $E_{\text{id}} \in \mathbb{R}^{3n \times 80}$, reflectance $E_{\text{alb}} \in \mathbb{R}^{3n \times 80}$, and expression $E_{\text{exp}} \in \mathbb{R}^{3n \times 76}$ basis and the corresponding standard deviations $\sigma_{\text{id}} \in \mathbb{R}^{80}$, $\sigma_{\text{alb}} \in \mathbb{R}^{80}$, and $\sigma_{\text{exp}} \in \mathbb{R}^{76}$ are given. The model has 53K vertices and 106K faces. A synthesized image C_S is generated through rasterization of the model under a rigid model transformation $\Phi(\mathbf{v})$ and the full perspective transformation $\Pi(\mathbf{v})$. Illumination is approximated by the first three bands of Spherical Harmonics (SH)¹³ basis functions, assuming Lambertian surfaces and smooth distant illumination, neglecting self-shadowing.

Synthesis is dependent on the face model parameters α, β, δ , the illumination parameters γ , the rigid transformation \mathbf{R}, \mathbf{t} , and the camera parameters κ defining Π . The vector of unknowns \mathcal{P} is the union of these parameters.

6. ENERGY FORMULATION

Given a monocular input sequence, we reconstruct all unknown parameters \mathcal{P} jointly with a robust variational optimization. The proposed objective is highly non-linear in the unknowns and has the following components:

$$E(\mathcal{P}) = \underbrace{w_{\text{col}} E_{\text{col}}(\mathcal{P})}_{\text{data}} + \underbrace{w_{\text{lan}} E_{\text{lan}}(\mathcal{P})}_{\text{data}} + \underbrace{w_{\text{reg}} E_{\text{reg}}(\mathcal{P})}_{\text{prior}} \quad (3)$$

The data term measures the similarity between the synthesized imagery and the input data in terms of photo-consistency E_{col} and facial feature alignment E_{lan} . The likelihood of a given parameter vector \mathcal{P} is taken into account by the statistical regularizer E_{reg} . The weights w_{col} , w_{lan} , and w_{reg} balance the three different sub-objectives. In all of our experiments, we set $w_{\text{col}} = 1$, $w_{\text{lan}} = 10$, and $w_{\text{reg}} = 2.5 \cdot 10^{-5}$. In the following, we introduce the different sub-objectives.

Photo-Consistency. In order to quantify how well the input data is explained by a synthesized image, we measure the photometric alignment error on pixel level:

$$E_{\text{col}}(\mathcal{P}) = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{p} \in \mathcal{V}} \|C_S(\mathbf{p}) - C_I(\mathbf{p})\|_2, \quad (4)$$

where C_S is the synthesized image, C_I is the input RGB image, and $\mathbf{p} \in \mathcal{V}$ denote all visible pixel positions in C_S . We use the $\ell_{2,1}$ -norm instead of a least-squares formulation to be robust against outliers. In our scenario, distance in color space is based on ℓ_2 , while in the summation over all pixels an ℓ_1 -norm is used to enforce sparsity.

Feature Alignment. In addition, we enforce feature similarity between a set of salient facial feature point pairs detected in the RGB stream:

$$\ln(\mathcal{P}) = \frac{1}{|\mathcal{F}|} \sum_{\mathbf{f}_j \in \mathcal{F}} w_{\text{conf},j} \|\mathbf{f}_j - \Pi(\Phi(\mathbf{v}_j))\|_2 \quad (5)$$

To this end, we employ a state-of-the-art facial landmark tracking algorithm by Saragih et al.¹⁴ Each feature point $\mathbf{f}_j \in \mathcal{F} \subset \mathbb{R}^2$ comes with a detection confidence $w_{\text{conf},j}$ and corresponds to a unique vertex $\mathbf{v}_j = \mathcal{M}_{\text{geo}}(\alpha, \delta) \in \mathbb{R}^3$ of our face prior. This helps avoiding local minima in the highly complex energy landscape of $E_{\text{col}}(\mathcal{P})$.

Statistical Regularization. We enforce plausibility of the synthesized faces based on the assumption of a normal distributed population. To this end, we enforce the parameters to stay statistically close to the mean:

$$E_{\text{reg}}(\mathcal{P}) = \sum_{i=1}^{80} \left[\left(\frac{\alpha_i}{\sigma_{\text{id},i}} \right)^2 + \left(\frac{\beta_i}{\sigma_{\text{alb},i}} \right)^2 \right] + \sum_{i=1}^{76} \left(\frac{\delta_i}{\sigma_{\text{exp},i}} \right)^2 \quad (6)$$

This commonly used regularization strategy prevents degenerations of the facial geometry and reflectance, and guides the optimization strategy out of local minima.¹

7. DATA-PARALLEL OPTIMIZATION

The proposed robust tracking objective is a general unconstrained non-linear optimization problem. We use IRLS to minimize this objective in real-time using a novel data-parallel GPU-based solver. The key idea of IRLS is to transform the problem, in each iteration, to a non-linear least-squares problem by splitting the norm in two components:

$$\|r(\mathcal{P})\|_2 = \underbrace{\left(\|r(\mathcal{P}_{\text{old}})\|_2 \right)^{-1}}_{\text{constant}} \cdot \|r(\mathcal{P})\|_2$$

Here, $r(\cdot)$ is a general residual and \mathcal{P}_{old} is the solution computed in the last iteration. Thus, the first part is kept constant during one iteration and updated afterwards. Close in spirit to Thies et al.,¹⁹ each single iteration step is implemented using the Gauss-Newton approach. We take a single GN step in every IRLS iteration and solve the corresponding system of normal equations $\mathbf{J}^T \mathbf{J} \delta^* = -\mathbf{J}^T \mathbf{F}$ based on PCG (Preconditioned Conjugate Gradient) to obtain an optimal linear parameter update δ^* . The Jacobian \mathbf{J} and the systems' right hand side $-\mathbf{J}^T \mathbf{F}$ are precomputed and stored in device memory for later processing as proposed by Thies et al.¹⁹ For more details we refer to the original paper.²¹ Note that our complete framework is implemented using DirectX for rendering and DirectCompute for optimization. The joint graphics and compute capability of DirectX11 enables us to execute the analysis-by-synthesis loop without any resource mapping overhead between these two stages. In the case of an analysis-by-synthesis approach, this is essential for runtime performance, since many rendering-to-compute switches are required. To compute the Jacobian \mathbf{J} we developed a differential renderer that is based on the standard rasterizer of the graphics pipeline. To this end, during the synthesis stage, we additionally store the vertex and triangle attributes that are required for computing the partial derivatives to dedicated rendertargets. Using

this information a compute shader calculates the final derivatives that are needed for the optimization.

8. NON-RIGID MODEL-BASED BUNDLING

To estimate the identity of the actors in the heavily underconstrained scenario of monocular reconstruction, we introduce a non-rigid model-based bundling approach. Based on the proposed objective, we jointly estimate all parameters over k key-frames of the input video sequence. The estimated unknowns are the global identity $\{\alpha, \beta\}$ and intrinsics κ as well as the unknown per-frame pose $\{\delta^k, \mathbf{R}^k, \mathbf{t}^k\}_k$ and illumination parameters $\{\gamma^k\}_k$. We use a similar data-parallel optimization strategy as proposed for model-to-frame tracking, but jointly solve the normal equations for the entire key-frame set. For our non-rigid model-based bundling problem, the non-zero structure of the corresponding Jacobian is block dense. Our PCG solver exploits the non-zero structure for increased performance (see original paper). Since all keyframes observe the same face identity under potentially varying illumination, expression, and viewing angle, we can robustly separate identity from all other problem dimensions. Note that we also solve for the intrinsic camera parameters of Π , thus being able to process uncalibrated video footage. The employed Gauss-Newton framework is embedded in a hierarchical solution strategy (see Figure 3). The underlying hierarchy enables faster convergence and avoids getting stuck in local minima of the optimized energy function. We start optimizing on a coarse level and lift the solution to the next finer level using the parametric face model. In our experiments we used three levels with 25, 5, and 1 Gauss-Newton iterations for the coarsest, the medium, and the finest level, respectively. In each Gauss-Newton iteration, we employ 4 PCG steps to efficiently solve the underlying normal equations. Our implementation is not restricted

Figure 3. Non-rigid model-based bundling hierarchy: The top row shows the hierarchy of the input video and the second row the overlaid face model.



to the number k of used keyframes, but the processing time increases linearly with k . In our experiments we used $k = 6$ keyframes for the estimation of the identity parameters, which results in a processing time of only a few seconds ($\sim 20s$).

9. EXPRESSION TRANSFER

To transfer the expression changes from the source to the target actor while preserving person-specificness in each actor's expressions, we propose a sub-space deformation transfer technique. We are inspired by the deformation transfer energy of Sumner et al.,¹⁸ but operate directly in the space spanned by the expression blend-shapes. This not only allows for the precomputation of the pseudo-inverse of the system matrix, but also drastically reduces the dimensionality of the optimization problem allowing for fast real-time transfer rates. Assuming source identity α^s and target identity α^t fixed, transfer takes as input the neutral δ^i , deformed source δ^s , and the neutral target δ^t expression. Output is the transferred facial expression δ^t directly in the reduced sub-space of the parametric prior.

As proposed by Sumner and Popović,¹⁸ we first compute the source deformation gradients $\mathbf{A}_i \in \mathbb{R}^{3 \times 3}$ that transform the source triangles from neutral to deformed. The deformed target $i = \mathbf{M}_i(\alpha^t, \delta^i)$ is then found based on the undeformed state $i = \mathbf{M}_i(\alpha^t, \delta^i)$ by solving a linear least-squares problem. Let (i_0, i_1, i_2) be the vertex indices of the i -th triangle, $\mathbf{v} = [\mathbf{v}_{i_0}, \mathbf{v}_{i_1}, \mathbf{v}_{i_2}]$ and $\hat{\mathbf{v}} = [\hat{\mathbf{v}}_{i_0}, \hat{\mathbf{v}}_{i_1}, \hat{\mathbf{v}}_{i_2}]$, then the optimal unknown target deformation δ^t is the minimizer of:

$$\delta^t = \sum_i^{|F|} \|\mathbf{A}_i \mathbf{v} - \hat{\mathbf{v}}\|^2 \quad (7)$$

This problem can be rewritten in the canonical least-squares form by substitution:

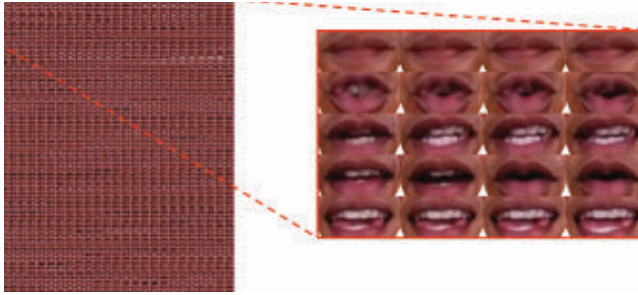
$$\delta^t = \|\mathbf{A} \delta^t - \mathbf{b}\|_2^2 \quad (8)$$

The matrix $\mathbf{A} \in \mathbb{R}^{6|F| \times 76}$ is constant and contains the edge information of the template mesh projected to the expression sub-space. Edge information of the target in neutral expression is included in the right-hand side $\mathbf{b} \in \mathbb{R}^{6|F|}$. \mathbf{b} varies with δ^s and is computed on the GPU for each new input frame. The minimizer of the quadratic energy can be computed by solving the corresponding normal equations. Since the system matrix is constant, we can precompute its *Pseudo Inverse* using a Singular Value Decomposition (SVD). Later, the small 76×76 linear system is solved in real-time. No additional smoothness term as in Bouaziz et al.² and Sumner and Popović¹⁸, is needed, since the blendshape model implicitly restricts the result to plausible shapes and guarantees smoothness.

10. MOUTH RETRIEVAL

For a given transferred facial expression, we need to synthesize a realistic target mouth region. To this end, we retrieve and warp the best matching mouth image from the target actor sequence (see Figure 4). We assume that

Figure 4. Mouth Database: We use the appearance of the mouth of a person that has been captured in the target video sequence.



sufficient mouth variation is available in the target video, that is, we assume that the entire target video is known or at least a short part of it. It is also important to note that we maintain the appearance of the target mouth. This leads to much more realistic results than either copying the source mouth region²³ or using a generic 3D teeth proxy.^{8,19} For detailed information on the mouth retrieval process, we refer to the original paper.

11. RESULTS

11.1. Live reenactment setup

Our live reenactment setup consists of standard consumer-level hardware. We capture a live video with a commodity webcam (source), and download monocular video clips from Youtube (target). In our experiments, we use a *Logitech HD Pro C920* camera running at 30Hz in a resolution of 640×480 ; although our approach is applicable to any consumer RGB camera. Overall, we show highly realistic reenactment examples of our algorithm on a variety of target Youtube videos at a resolution of 1280×720 . The videos show different subjects in different scenes filmed from varying camera angles; each video is reenacted by several volunteers as source actors. Reenactment results are generated at a resolution of 1280×720 . We show real-time reenactment results in Figure 5 and in the accompanying video.

11.2. Runtime

For all experiments, we use three hierarchy levels for tracking (source and target). In pose optimization, we only consider the second and third level, where we run one and seven Gauss-Newton steps, respectively. Within a Gauss-Newton step, we always run four PCG steps. In addition to tracking, our reenactment pipeline has additional stages whose timings are listed in Table 1. Our method runs in real time on a commodity desktop computer with an NVIDIA Titan X and an Intel Core i7-4770.

11.3. Tracking comparison to previous work

Face tracking alone is not the main focus of our work, but the following comparisons show that our tracking is on par with or exceeds the state of the art. Here we show some of the comparisons that we conducted in the original paper.

Cao et al. 2014:⁵ They capture face performance from monocular RGB in real time. In most cases, our and their

method produce similar high-quality results (see Figure 6); our identity and expression estimates are slightly more accurate though.

Thies et al. 2015:¹⁹ Their approach captures face performance in real-time from RGB-D, Figure 6. While we do not require depth data, results of both approaches are similarly accurate.

11.4. Reenactment evaluation

In Figure 7, we compare our approach against state-of-the-art reenactment by Garrido et al.⁸ Both methods provide highly realistic reenactment results; however, their method is fundamentally offline, as they require all frames of a sequence to be present at any time. In addition, they rely on a generic geometric teeth proxy which in some frames makes reenactment less convincing. In Figure 8, we compare against the work by Thies et al.¹⁹ Runtime and visual quality are similar for both approaches; however, their geometric teeth proxy leads to an undesired appearance of the reenacted mouth. Thies et al. use an RGB-D camera, which limits the application range; they cannot reenact Youtube videos.

12. LIMITATIONS

The assumption of Lambertian surfaces and smooth illumination is limiting, and may lead to artifacts in the presence of hard shadows or specular highlights; a limitation shared by most state-of-the-art methods. Scenes with face occlusions by long hair and a beard are challenging. Furthermore, we only reconstruct and track a low-dimensional blend-shape model (76 coefficients), which omits fine-scale static and transient surface details. Our retrieval-based mouth synthesis assumes sufficient visible expression variation in the target sequence. On a too short sequence, or when the target remains static, we cannot learn the person-specific mouth behavior. In this case, temporal aliasing can be observed, as the target space of the retrieved mouth samples is too sparse. Another limitation is caused by our commodity hardware setup (webcam, USB, and PCI), which introduces a small delay of ≈ 3 frames.

13. DISCUSSION

Our face reconstruction and photo-realistic re-rendering approach enables the manipulation of videos at real-time frame rates. In addition, the combination of the proposed approach with a voice impersonator or a voice synthesis system, would enable the generation of made-up video content that could potentially be used to defame people or to spread so-called “fake-news.” We want to emphasize that computer-generated content has been a big part of feature-film movies for over 30 years. Virtually every high-end movie production contains a significant percentage of synthetically generated content (from *Lord of the Rings* to *Benjamin Button*). These results are already hard to distinguish from reality and it often goes unnoticed that the content is not real. Thus, the synthetic modification of video clips was already possible for a long time, but it was a time consuming process and required domain

Figure 5. Results of our reenactment system. Corresponding run times are listed in Table 1. The length of the source and resulting output sequences is 965, 1436, and 1791 frames, respectively; the length of the input target sequences is 431, 286, and 392 frames, respectively.



Table 1. Avg. run times for the three sequences of Figure 5, from top to bottom.^a

CPU		GPU			FPS
SparseFT	MouthRT	DenseFT	DefTF	Synth	(Hz)
5.97ms	1.90ms	22.06ms	3.98ms	10.19ms	27.6
4.85ms	1.50ms	21.27ms	4.01ms	10.31ms	28.1
5.57ms	1.78ms	20.97ms	3.95ms	10.32ms	28.4

Figure 6. Comparison of our RGB tracking to Cao et al.⁵ and to RGB-D tracking by Thies et al.¹⁹

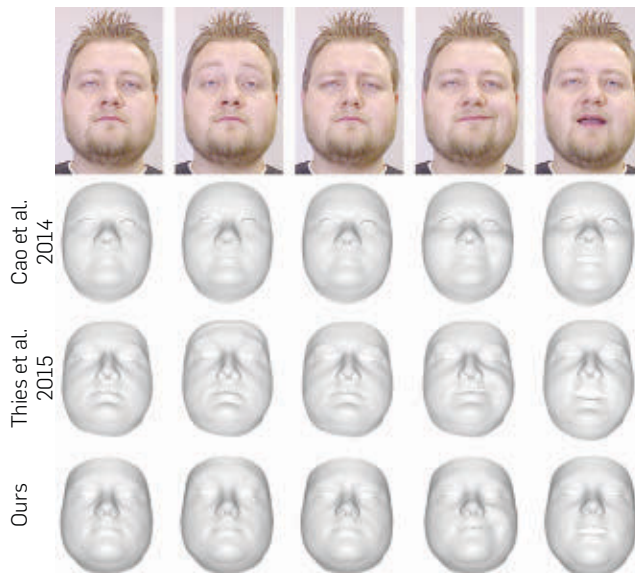


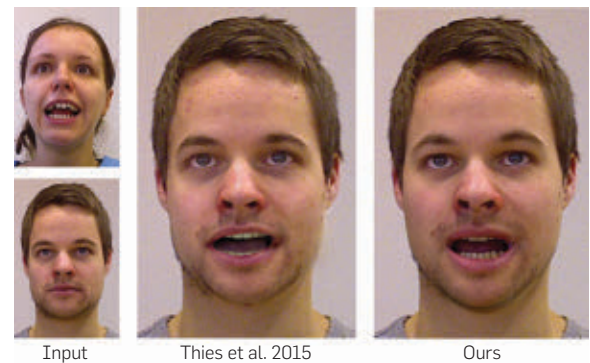
Figure 7. Dubbing: Comparison to Garrido et al.⁸



experts. Our approach is a game changer, since it enables editing of videos in real time on a commodity PC, which makes this technology accessible to non-experts. We hope that the numerous demonstrations of our reenactment systems will teach people to think more critical about the video content they

^a Standard deviations w.r.t. the final frame rate are 0:51, 0:56, and 0:59 fps, respectively. Note that CPU and GPU stages run in parallel.

Figure 8. Comparison of the proposed RGB reenactment to the RGB-D reenactment of Thies et al.¹⁹



consume every day, especially if there is no proof of origin. The presented system also demonstrates the need for sophisticated fraud detection and watermarking algorithms. We believe that the field of digital forensics will receive a lot of attention in the future.

14. CONCLUSION

The presented approach is the first real-time facial reenactment system that requires just monocular RGB input. Our live setup enables the animation of legacy video footage—for example, from Youtube—in real time. Overall, we believe our system will pave the way for many new and exciting applications in the fields of VR/AR, teleconferencing, or on-the-fly dubbing of videos with translated audio. One direction for future work is to provide full control over the target head. A properly rigged mouth and tongue model reconstructed from monocular input data will provide control over the mouth cavity, a wrinkle formation model will provide more realistic results by adding fine-scale surface detail and eye-tracking will enable control over the target's eye movement.

Acknowledgments

We would like to thank Chen Cao and Kun Zhou for the blendshape models and comparison data, as well as Volker Blanz, Thomas Vetter, and Oleg Alexander for the provided face data. The facial landmark tracker was kindly provided by TrueVisionSolution. We thank Angela Dai for the video voice over and Daniel Ritchie for video reenactment. This research is funded by the German Research Foundation (DFG), grant GRK-1773 Heterogeneous Image Systems, the ERC Starting Grant 335545 CapReal, and the Max Planck Center for Visual Computing and Communications (MPC-VCC). We also gratefully acknowledge the support from NVIDIA Corporation for hardware donations. □

References

1. Blanz, V., Vetter, T. A morphable model for the synthesis of 3d faces. *Proc. SIGGRAPH* (1999), ACM Press/Addison-Wesley Publishing Co., 187–194.
2. Bouaziz, S., Wang, Y., Pauly, M. Online modeling for realtime facial animation. *ACM TOG* 32, 4 (2013), 40.
3. Bregler, C., Covell, M., Slaney, M. Video rewrite: Driving visual speech with audio. *Proc. SIGGRAPH* (1997), ACM Press/Addison-Wesley

Publishing Co., 353–360.

4. Cao, C., Bradley, D., Zhou, K., Beeler, T. Real-time high-fidelity facial performance capture. *ACM TOG 34*, 4 (2015), 46: 1–46:9.
5. Cao, C., Hou, Q., Zhou, K. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM TOG 33*, 4 (2014), 43.
6. Chen, Y.-L., Wu, H.-T., Shi, F., Tong, X., Chai, J. Accurate and robust 3d facial capture using a single rgbd camera. *Proc. ICCV* (2013), 3615–3622.
7. Garrido, P., Valgaerts, L., Rehmsen, O., Thormaehlen, T., Perez, P., Theobalt, C. Automatic face reenactment. *Proc. CVPR* (2014).
8. Garrido, P., Valgaerts, L., Sarmadi, H., Steiner, I., Varanasi, K., Perez, P., Theobalt, C. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. *Computer Graphics Forum*, Wiley-Blackwell, Hoboken, New Jersey, 2015.
9. Hu, L., Saito, S., Wei, L., Nagano, K., Seo, J., Fursund, J., Sadeghi, I., Sun, C., Chen, Y., Li, H. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.* 36, 6 (2017), 195:1–195:14.
10. Kemelmacher-Shlizerman, I., Sankar, A., Shechtman, E., Seitz, S.M. Being john malkovich. In *Computer Vision—ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part I* (2010), 341–353.
11. Li, H., Yu, J., Ye, Y., Bregler, C. Realtime facial animation with on-the-fly correctives. *ACM TOG 32*, 4 (2013), 42.
12. Li, K., Xu, F., Wang, J., Dai, Q., Liu, Y. A data-driven approach for facial expression synthesis in video. *Proc. CVPR* (2012), 57–64.
13. Ramamoorthi, R., Hanrahan, P. A signal-processing framework for inverse rendering. *Proc. SIGGRAPH* (ACM, 2001), 117–128.
14. Saragih, J.M., Lucey, S., Cohn, J.F. Deformable model fitting by regularized landmark mean-shift. *IJCV 91*, 2 (2011), 200–215.
15. Saragih, J.M., Lucey, S., Cohn, J.F. Real-time avatar animation from a single image. *Automatic Face and Gesture Recognition Workshops* (2011), 213–220.
16. Shi, F., Wu, H.-T., Tong, X., Chai, J. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM TOG 33*, 6 (2014), 222.
17. Siegl, C., Lange, V., Stamminger, M., Bauer, F., Thies, J. Faceforge: Markerless non-rigid face multi-projection mapping. *IEEE Transactions on Visualization and Computer Graphics*, 2017.
18. Sumner, R.W., Popović, J. Deformation transfer for triangle meshes. *ACM TOG 23*, 3 (2004), 399–405.
19. Thies, J., Zollhöfer, M., Nießner, M., Valgaerts, L., Stamminger, M., Theobalt, C. Real-time expression transfer for facial reenactment. *ACM Trans. Graph. (TOG)* 34, 6 (2015).
20. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M. Demo of face2face: Real-time face capture and reenactment of RGB videos. *ACM SIGGRAPH 2016 Emerging Technologies, SIGGRAPH '16* (ACM, 2016), New York, NY, USA, 5:1–5:2.
21. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M. Face2Face: Real-time face capture and reenactment of RGB videos. *Proc. Comp. Vision and Pattern Recog. (CVPR), IEEE* (2016).
22. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M. FaceVR: Real-time facial reenactment and eye gaze control in virtual reality. *ArXiv, Non-Peer-Reviewed Prepublication by the Authors*, abs/1610.03151 (2016).
23. Vlastic, D., Brand, M., Pfister, H., Popović, J. Face transfer with multilinear models. *ACM TOG 24*, 3 (2005), 426–433.
24. Weise, T., Bouaziz, S., Li, H., Pauly, M. Realtime Performance-Based Facial Animation 3D, 4 (2011), 77.
25. Weise, T., Li, H., Gool, L.V., Pauly, M. Face/off: Live facial puppetry. *Proc. 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)*, ETH Zurich, August 2009. Eurographics Association.

Justus Thies and Matthias Nießner ([justus.thies, niessner]@tum.de), Technical University Munich, Garching, Germany.

Michael Zollhöfer (zollhoefer@cs.stanford.edu), Stanford University, Stanford, CA, USA.

Marc Stamminger (marc.stamminger@fau.de), University of Erlangen-Nuremberg, Erlangen, Germany.

Christian Theobalt (theobalt@mpi-inf.mpg.de), Max-Planck-Institute for Informatics, Saarbrücken, Germany.

Copyright held by authors/owners. Publication rights licensed to ACM. \$15.00

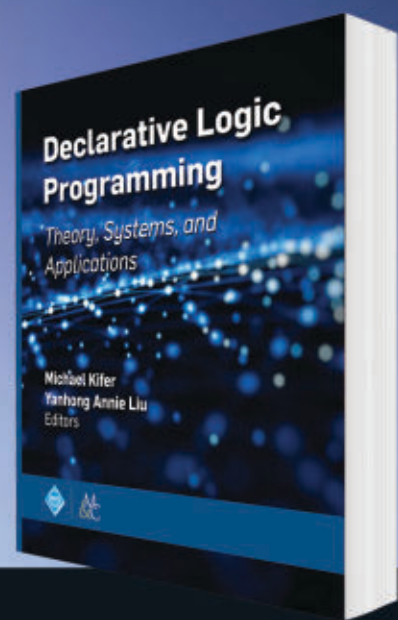


Watch the authors discuss this work in the exclusive *Communications* video. <https://cacm.acm.org/videos/face2face>

Theory, Systems, and Applications

Declarative Logic Programming

Edited by **Michael Kifer & Yanhong Annie Liu**
 ISBN: 978-1-970001-969 | DOI: 10.1145/3191315
<http://books.acm.org>
<http://www.morganclaypoolpublishers.com/acm>



ACM BOOKS

Technical Perspective

Attacking Cryptographic Key Exchange with Precomputation

By Dan Boneh

THE DIFFIE-HELLMAN KEY exchange protocol is at the heart of many cryptographic protocols widely used on the Internet. It is used for session setup in HTTPS (TLS), in SSH, in IPsec, and others. The original protocol, as described by Diffie and Hellman, operates by choosing a large prime p and computing certain exponentiations modulo this prime. For the protocol to be secure one needs, at the very least, that the discrete-log problem modulo the prime p be difficult to solve. This problem is quite easy to state: fix a large prime p , and an integer $0 < g < p$ (a generator). Next, choose an integer $0 < x < p$ and compute $h = g^x$ modulo p . The discrete-log problem is to compute x given only p , g and h . If this problem could be solved efficiently, for most h , then the Diffie-Hellman protocol for the chosen (p, g) would be insecure.

The authors of the following paper show that, in practice, implementations that use Diffie-Hellman tend to choose a universally fixed prime p (and fixed g). For example, many SSH servers and IPsec VPNs use a fixed universal 1,024-bit prime p . The same is true for HTTPS Web servers, although to a lesser extent.

Is it safe to use the same 1,024-bit prime p everywhere? The authors show that the answer is no. The reason is a beautiful precomputation attack on the discrete-log problem modulo a prime. A precomputation attack proceeds in two steps: First, in a one-time offline phase, before trying to attack any particular victim, the attacker works hard to compute a certain table based on the fixed p and g . Then, when attacking a victim session, the attacker uses the precomputed table to quickly compute discrete-log and break the session. The same precomputed table can be used to quickly break many sessions.

Precomputation attacks affect many cryptographic schemes. For example, they are often used to break weak password systems—one first precomputes a rainbow table, and then uses the table

to quickly break many hashed passwords. The beautiful insight of this paper is that precomputation can be devastating for systems that use Diffie-Hellman modulo a prime. Precomputation attacks are a real threat and must be taken into account when choosing parameters for real-world cryptography.


The authors speculate that a precomputation attack on discrete-log modulo a fixed 1,024-bit prime is within reach for a nation state. Because a small number of fixed primes is employed by a large number of websites, a precomputation attack on a few primes can be used to compromise encrypted Internet traffic at many sites.

To make matters worse, the authors show there is no need to break 1,024-bit primes to attack TLS. The reason is a weak TLS cryptography suite called TLS Export. This suite was included in TLS due to export control regulations that were in effect at the time that TLS was designed. TLS Export includes support for 512-bit primes, where discrete-log is woefully insecure. Sadly, TLS Export is still supported by many websites, and many (82%) use a fixed 512-bit prime shipped with the Apache Web server. The precomputation attack is extremely effective against this 512-bit prime. The authors carry out the offline precomputation phase in a few days, and the result-

ing table enables an online attack on a victim session in just under a minute.

To make matters even worse, the authors describe a new clever attack on TLS 1.2, called *logjam*, which lets an attacker downgrade a victim connection to TLS Export. The resulting session is then vulnerable to a precomputation attack. Logjam exposes a significant flaw in the design of TLS 1.2.

So, what should we do? The short answer is that websites must migrate to TLS 1.3. TLS 1.3 is a recent significant upgrade to the TLS protocol. Compliant implementations must support Diffie-Hellman using an elliptic curve group called NIST P-256. It is likely that many websites will use Diffie-Hellman in this group. Using a universally fixed group seems as bad as using a universal prime p , however, currently there is no known practical precomputation attack on elliptic curve Diffie-Hellman, so that the precomputation attacks discussed earlier do not apply, as far as we know. One point of concern is NSA's August 2015 announcement recommending that companies stop their transition to elliptic curve cryptography or, if they already have transitioned, use larger elliptic curve parameters. The official reason in the notice is the concern over a quantum computer that can break elliptic curve Diffie-Hellman. One may wonder, however, if there are other reasons behind this announcement. Is there a yet-to-be discovered practical preprocessing attack on P-256? Currently, there is no indication that such an attack exists.

In summary, preprocessing attacks are a real concern in cryptography. It is critically important to take them into account when choosing cryptographic parameters. The following paper is a wonderful illustration of this. 

Dan Boneh is a professor of computer science and electrical engineering at Stanford University, and co-director of the Stanford Computer Security Lab, Stanford, CA, USA.

Copyright held by author/owner.

The authors of the following paper show that, in practice, implementations that use Diffie-Hellman tend to choose a universally fixed prime p (and fixed g).

Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice

By David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann

Abstract

We investigate the security of Diffie-Hellman key exchange as used in popular Internet protocols and find it to be less secure than widely believed. First, we present Logjam, a novel flaw in TLS that lets a man-in-the-middle downgrade connections to “export-grade” Diffie-Hellman. To carry out this attack, we implement the number field sieve discrete logarithm algorithm. After a week-long precomputation for a specified 512-bit group, we can compute arbitrary discrete logarithms in that group in about a minute. We find that 82% of vulnerable servers use a single 512-bit group, and that 8.4% of Alexa Top Million HTTPS sites are vulnerable to the attack.^a In response, major browsers have changed to reject short groups.

We go on to consider Diffie-Hellman with 768- and 1024-bit groups. We estimate that even in the 1024-bit case, the computations are plausible given nation-state resources. A small number of fixed or standardized groups are used by millions of servers; performing precomputation for a single 1024-bit group would allow passive eavesdropping on 18% of popular HTTPS sites, and a second group would allow decryption of traffic to 66% of IPsec VPNs and 26% of SSH servers. A close reading of published NSA leaks shows that the agency’s attacks on VPNs are consistent with having achieved such a break. We conclude that moving to stronger key exchange methods should be a priority for the Internet community.

1. INTRODUCTION

Diffie-Hellman (DH) key exchange is a popular cryptographic algorithm that allows Internet protocols to agree on a shared key and negotiate a secure connection. It is fundamental to protocols such as Hypertext Transport Protocol Secure (HTTPS), Secure Shell (SSH), Internet Protocol Security (IPsec), Simple Mail Transfer Protocol Secure (SMTPS), and other protocols that rely on Transport Layer Security (TLS). Many protocols use Diffie-Hellman to achieve *perfect forward secrecy*, the property that a compromise of the long-term keys used for authentication does not compromise session keys for past connections. We examine how Diffie-Hellman is commonly implemented and deployed with common protocols and find that, in practice, it frequently offers less security than widely believed.

There are two reasons for this. First, a surprising number of servers use weak Diffie-Hellman parameters or maintain

^a Except where otherwise noted, the experimental data and network measurements for this article were obtained in early 2015.

support for obsolete 1990s-era “export-grade” cryptography. More critically, the common practice of using standardized, hard-coded, or widely shared Diffie-Hellman parameters has the effect of dramatically reducing the cost of large-scale attacks, bringing some within range of feasibility.

The current best technique for attacking Diffie-Hellman relies on compromising one of the private exponents (a , b) by computing the discrete logarithm of the corresponding public value ($g^a \bmod p$, $g^b \bmod p$). With state-of-the-art number field sieve algorithms, computing a single discrete logarithm is more difficult than factoring a Rivest–Shamir–Adleman (RSA) modulus of the same size. However, an adversary who performs a large precomputation for a prime p can then quickly calculate arbitrary discrete logarithms in that group, amortizing the cost over all targets that share this parameter. Although this fact is well known among mathematical cryptographers, it seems to have been lost among practitioners deploying cryptosystems. We exploit it to obtain the following results.

Active attacks on export ciphers in TLS

We introduce Logjam, a new attack on TLS by which a man-in-the-middle attacker can downgrade a connection to export-grade cryptography. This attack is reminiscent of the FREAK attack¹ but applies to the ephemeral Diffie-Hellman ciphersuites and is a TLS protocol flaw rather than an implementation vulnerability. We present measurements that show that this attack applies to 8.4% of Alexa Top Million HTTPS sites and 3.4% of all HTTPS servers that have browser-trusted certificates.

To exploit this attack, we implemented the number field sieve discrete logarithm algorithm and carried out precomputation for two 512-bit Diffie-Hellman groups used by more than 92% of the vulnerable servers. This allows us to compute individual discrete logarithms in about a minute. Using our discrete logarithm oracle, we can compromise connections to over 7% of Alexa Top Million HTTPS sites. Discrete logarithms over larger groups have been computed before,² but, as far as we are aware, this is the first time they have been exploited to expose concrete vulnerabilities in real-world systems.

Risks from common 1024-bit groups

We explore the implications of precomputation attacks for 768- and 1024-bit groups, which are widely used in practice

The full version of this paper was published in *Proceedings of the 22nd Conference on Computer and Communications Security (CCS)*, October 2015, ACM. The full paper and additional materials are available at <https://weakdh.org/>.

and still considered secure. We estimate the computational resources necessary to compute discrete logarithms in groups of these sizes, concluding that 768-bit groups are within range of academic teams, and 1024-bit groups may plausibly be within range of nation-state adversaries. In both cases, individual logarithms can be quickly computed after the initial precomputation.

We then examine evidence from published Snowden documents that suggests that the National Security Agency (NSA) may already be exploiting 1024-bit Diffie-Hellman to decrypt Virtual Private Network (VPN) traffic. We perform measurements to understand the implications of such an attack for popular protocols, finding that an attacker who could perform precomputations for ten 1024-bit groups could passively decrypt traffic to about 66% of Internet Key Exchange (IKE) VPNs, 26% of SSH servers, and 24% of popular HTTPS sites.

Mitigations and lessons

In response to the Logjam attack, mainstream browsers have implemented a more restrictive policy on the size of Diffie-Hellman groups they accept, and Google Chrome has discontinued support for finite field key exchanges. We further recommend that TLS servers disable export-grade cryptography and carefully vet the Diffie-Hellman groups they use. In the longer term, we advocate that protocols migrate to elliptic curve Diffie-Hellman.

2. DIFFIE-HELLMAN CRYPTANALYSIS

Diffie-Hellman key exchange was the first published public-key algorithm.⁵ In the simple case of prime groups, Alice and Bob agree on a prime p and a generator g of a multiplicative subgroup modulo p . Then each generates a random private exponent, a and b . Alice sends $g^a \bmod p$, Bob sends $g^b \bmod p$, and each computes a shared secret $g^{ab} \bmod p$. While there is also a Diffie-Hellman exchange over elliptic curve groups, we address only the “mod p ” case.

The security of Diffie-Hellman is not known to be equivalent to the discrete logarithm problem, but computing discrete logarithms remains the best known cryptanalytic attack. An attacker who can find the discrete logarithm x from $y = g^x \bmod p$ can easily find the shared secret.

Textbook descriptions of discrete logarithm algorithms can be misleading about the computational tradeoffs, for example by optimizing for computing a *single* discrete

logarithm. In fact, as illustrated in Figure 1, a single large precomputation on p can be used to efficiently break *all* Diffie-Hellman exchanges made with that prime.

Diffie-Hellman is typically implemented with prime fields and large group orders. In this case, the most efficient known algorithm for computing discrete logarithms is the Number Field Sieve (NFS).^{9, 11, 18} The algorithm has four stages with different computational properties. The first three steps are only dependent on the prime p and comprise most of the computation.

First is *polynomial selection*, in which one finds a polynomial $f(z)$ defining a number field $\mathbb{Q}[z]/f(z)$ for the computation. This parallelizes well and is only a small portion of the runtime.

In the second stage, *sieving*, one factors ranges of integers and number field elements in batches to find many relations of elements, all of whose prime factors are less than some bound B (called B -smooth). Sieving parallelizes well, but is computationally expensive, because we must search through and attempt to factor many elements.

In the third stage, *linear algebra*, we construct a large, sparse matrix consisting of the coefficient vectors of prime factorizations we have found. This stage can be parallelized in a limited fashion, and produces a database of logarithms which are used as input to the final stage.

The final stage, *descent*, actually deduces the discrete logarithm of the target y . We re-sieve until we find a set of relations that allow us to write the logarithm of y in terms of the logarithms in the precomputed database. Crucially, descent is the only NFS stage that involves y (or g), so polynomial selection, sieving, and linear algebra can be done once for a prime p and reused to compute the discrete logarithms of many targets.

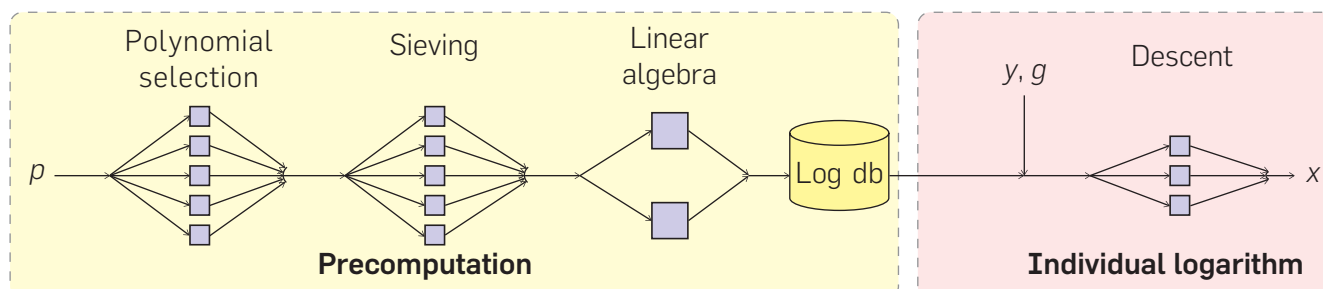
The numerous parameters of the algorithm allow some flexibility to reduce time on some computational steps at the expense of others. For example, sieving more will result in a smaller matrix, making linear algebra cheaper, and doing more work in the precomputation makes the final descent step easier.

Standard primes

Generating safe primes^b can be computationally burdensome, so many implementations use standardized

^b An odd prime p is safe when $(p - 1)/2$ is prime.

Figure 1. Number field sieve for discrete logarithms. This algorithm consists of a precomputation stage that depends only on the prime p and a descent stage that computes individual logarithms. With sufficient precomputation, an attacker can quickly break any Diffie-Hellman instances that use a particular p .



Diffie-Hellman parameters. A prominent example is the Oakley groups,¹⁷ which give “safe” primes of length 768 (Oakley Group 1), 1024 (Oakley Group 2), and 1536 (Oakley Group 5). These groups were published in 1998 and have been used for many applications since, including IKE, SSH, Tor, and Off-the-Record Messaging (OTR).

When primes are of sufficient strength, there seems to be no disadvantage to reusing them. However, widespread reuse of Diffie-Hellman groups can convert attacks that are at the limits of an adversary’s capabilities into devastating breaks, since it allows the attacker to amortize the cost of discrete logarithm precomputation among vast numbers of potential targets.

3. ATTACKING TLS

TLS supports Diffie-Hellman as one of several possible key exchange methods, and prior to public disclosure of our attack, about two-thirds of popular HTTPS sites supported it, most commonly using 1024-bit primes. However, a smaller number of servers also support legacy “export-grade” Diffie-Hellman using 512-bit primes that are well within reach of NFS-based cryptanalysis. Furthermore, for both normal and export-grade Diffie-Hellman, the vast majority of servers use a handful of common groups.

In this section, we exploit these facts to construct a novel attack against TLS, which we call the Logjam attack. First, we perform NFS precomputations for the two most popular 512-bit primes on the web, so that we can quickly compute the discrete logarithm for any key exchange message that uses one of them. Next, we show how a man-in-the-middle, so armed, can attack connections between popular browsers and any server that allows export-grade Diffie-Hellman, by using a TLS protocol flaw to downgrade the connection to export-strength and then recovering the session key. We find that this attack with our precomputations can compromise connections to about 8% of HTTPS servers among Alexa Top Million domains.

3.1. TLS and Diffie-Hellman

The TLS handshake begins with a negotiation to determine the cryptographic algorithms used for the session. The client sends a list of supported ciphersuites (and a random nonce cr) within the ClientHello message, where each cipher-suite specifies a key exchange algorithm and other primitives. The server selects a cipher-suite from the client’s list and signals its selection in a ServerHello message (containing a random nonce sr).

TLS specifies ciphersuites supporting multiple varieties of Diffie-Hellman. Textbook Diffie-Hellman with unrestricted strength is called “ephemeral” Diffie-Hellman, or DHE, and is identified by ciphersuites that begin with `TLS_DHE_*`.^c In DHE, the server is responsible for selecting the Diffie-Hellman parameters. It chooses a group (p, g) , computes g^b , and sends a ServerKeyExchange message containing a signature over the tuple (cr, sr, p, g, g^b) using the long-term signing key from its certificate. The client verifies the signature and responds with a ClientKeyExchange message containing g^a .

^c New ciphersuites that use elliptic curve Diffie-Hellman (ECDHE) are gaining in popularity, but we focus exclusively on the traditional prime field variety.

To ensure agreement on the negotiation messages, and to prevent downgrade attacks, each party computes the TLS master secret from g^{ab} and calculates a Message Authentication Code (MAC) of its view of the handshake transcript. These MACs are exchanged in a pair of Finished messages and verified by the recipients.

To comply with 1990s-era U.S. export restrictions on cryptography, SSL 3.0 and TLS 1.0 supported reduced-strength DHE_EXPORT ciphersuites that were restricted to primes no longer than 512 bits. In all other respects, DHE_EXPORT protocol messages are identical to DHE. The relevant export restrictions are no longer in effect, but many servers maintain support for backward compatibility.

To understand how HTTPS servers in the wild use Diffie-Hellman, we modified the ZMap⁶ toolchain to offer DHE and DHE_EXPORT ciphersuites and scanned TCP/443 on both the full public IPv4 address space and the Alexa Top Million domains. The scans took place in March 2015. Of 539,000 HTTPS sites among Top Million domains, we found that 68.3% supported DHE and 8.4% supported DHE_EXPORT. Of 14.3mn IPv4 HTTPS servers with browser-trusted certificates, 23.9% supported DHE and 4.9% DHE_EXPORT.

While the TLS protocol allows servers to generate their own Diffie-Hellman parameters, just two 512-bit primes account for 92.3% of Alexa Top Million domains that support DHE_EXPORT (Table 1), and 92.5% of all servers with browser-trusted certificates that support DHE_EXPORT. The most popular 512-bit prime was hard-coded into many versions of Apache; the second most popular is the `mod_ssl` default for DHE_EXPORT.

3.2. Active downgrade to export-grade DHE

Given the widespread use of these primes, an attacker with the ability to compute discrete logarithms in 512-bit groups could efficiently break DHE_EXPORT handshakes for about 8% of Alexa Top Million HTTPS sites, but modern browsers never negotiate export-grade ciphersuites. To circumvent this, we show how an attacker can downgrade a regular DHE connection to use a DHE_EXPORT group, and thereby break both the confidentiality and integrity of application data.

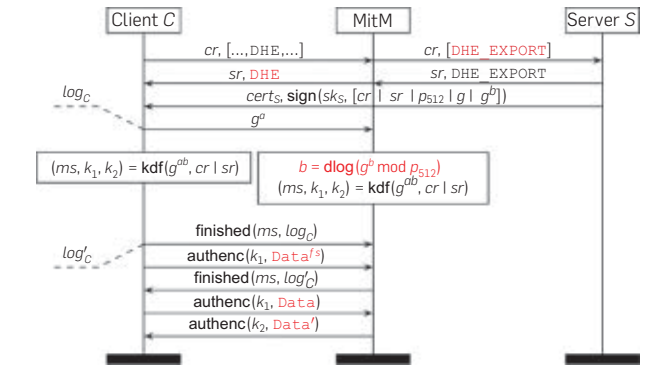
The attack, which we call Logjam, is depicted in Figure 2 and relies on a flaw in the way TLS composes DHE and

Table 1. Top 512-bit Diffie-Hellman primes for TLS^d.

Source	Popularity	Prime
Apache	82%	9fd8b8b8a004544f0045f1737d0ba2e0b274cdf1a9f588218fb435316a16e374171fd19d8d8f37c39bf863fd60e3e300680a3030c6e4c3757d08f70e6aa871033
mod_ssl	10%	d4bcd52406f69b35994b88de5db89682c8157f62d8f33633ee5772f11f05ab22d6b5145b9f241e5acc31ff090a4bc71148976f76795094e71e7903529f5a824b
(others)	8%	(463 distinct primes)

^d 8.4% of Alexa Top Million HTTPS domains allow DHE_EXPORT, of which 92.3% use one of the two most popular primes, shown here.

Figure 2. The Logjam attack. A man-in-the-middle can force TLS clients to use export-strength Diffie-Hellman with any server that allows DHE_EXPORT. Then, by finding the 512-bit discrete log, the attacker can learn the session key and arbitrarily read or modify the contents. $Data^s$ refers to False Start application data that some TLS clients send before receiving the server's Finished message.



DHE_EXPORT. When a server selects DHE_EXPORT for a handshake, it proceeds by issuing a signed ServerKeyExchange message containing a 512-bit p_{512} , but the structure of this message is identical to the message sent during standard DHE ciphersuites. Critically, the signed portion of the server's message fails to include any indication of the specific ciphersuite that the server has chosen. Provided that a client offers DHE, an active attacker can rewrite the client's ClientHello to offer a corresponding DHE_EXPORT ciphersuite accepted by the server and remove other ciphersuites that could be chosen instead. The attacker rewrites the ServerHello response to replace the chosen DHE_EXPORT ciphersuite with a matching non-export ciphersuite and forwards the ServerKeyExchange message to the client as is. The client will interpret the export-grade tuple (p_{512}, g, g^b) as valid DHE parameters chosen by the server and proceed with the handshake. The client and server have different handshake transcripts at this stage, but an attacker who can compute b in close to real time can then derive the master secret and connection keys to complete the handshake with the client.

There are two remaining challenges in implementing this active downgrade attack. The first is to compute individual discrete logarithms in close to real time, and the second is to delay handshake completion until the discrete logarithm computation has had time to finish.

3.3. 512-bit discrete logarithm computations

We modified CADO-NFS¹⁹ to implement the number field sieve discrete logarithm algorithm and applied it to the top two DHE_EXPORT primes shown in Table 1. Precomputation took seven days for each prime, after which computing individual logarithms requires a median of 70 seconds.

Precomputation. As illustrated in Figure 1, the precomputation phase includes the polynomial selection, sieving, and linear algebra steps. For this precomputation, we deliberately sieved more than strictly necessary. This enabled two optimizations: first, with more relations obtained from sieving, we eventually obtain a larger database of known logarithms, which makes the descent faster. Second, more

sieving relations also yield a smaller linear algebra step, which is desirable because sieving is much easier to parallelize than linear algebra.

For the polynomial selection and sieving steps, we used idle time on 2000–3000 microprocessor cores in parallel. Polynomial selection ran for about 3hrs (7,600 core-hours). Sieving ran for 15hrs (21,400 core-hours). This sufficed to collect 40mn relations of which 28mn were unique, involving 15mn primes of at most 27 bits.

From this data set, we obtained a square matrix with 2.2mn rows and columns, with 113 nonzero coefficients per row on average. We solved the corresponding linear system on a 36-node cluster using the block Wiedemann algorithm.^{4,20} Using unoptimized code, the computation finished in 120hrs (60,000 core-hours).

The experiment above was done with CADO-NFS in early 2015. As of 2017, release 2.3 of CADO-NFS¹⁹ performs 20% faster for sieving, and drastically faster for linear algebra, since 9,000 core-hours suffice to solve the same linear system on the same hardware. In total, the wall-clock time for each precomputation was slightly over one week in 2015, and is reduced to about two days with current hardware and more recent software.

Descent. Once this precomputation was finished, we were able to run the final descent step to compute individual discrete logarithms in about a minute. We implemented the descent calculation in a mix of Python and C. On average, computing individual logarithms took about 70sec, but the time varied from 34sec to 206sec on a server with two 18-core Intel Xeon E5-2699 CPUs. For purposes of comparison, a single 512-bit RSA factorization using the CADO-NFS implementation takes about four days of wall-clock time on the computer used for the descent.¹⁹

3.4. Active attack implementation

The main challenge in performing this attack is to compute the shared secret g^{ab} before the handshake completes in order to forge a Finished message from the server. With our descent implementation, the computation takes an average of 70sec, but there are several ways an attacker can work around this delay:

Non-browser clients. Different TLS clients impose different time limits, after which they kill the connection. Command-line clients such as curl and git have long or no timeouts, and we can hijack their connections without difficulty.

TLS warning alerts. Web browsers tend to have shorter timeouts, but we can keep their connections alive by sending TLS warning alerts, which are ignored by the browser but reset the handshake timer. For example, this allows us to keep Firefox TLS connections alive indefinitely.

Ephemeral key caching. Many TLS servers do not use a fresh value b for each connection, but instead compute g^b once and reuse it for multiple negotiations. For example, F5 BIG-IP load balancers will reuse g^b by default. Microsoft Schannel caches g^b for two hours — this setting is hard-coded. For these servers, an attacker can compute the discrete logarithm of g^b from one connection and use it to attack later handshakes.

TLS False Start. Even when clients enforce shorter timeouts and servers do not reuse values for b , the attacker can still break the confidentiality of user requests that use TLS False Start. Recent versions of Chrome, Internet Explorer, and Firefox implement False Start, but their policies on when to enable it vary. Firefox 35, Chrome 41, and Internet Explorer (Windows 10) send False Start data with DHE.

In these cases, a man-in-the-middle can record the handshake and decrypt the False Start payload at leisure.

4. NATION-STATE THREATS TO DIFFIE-HELLMAN

The previous sections demonstrate the existence of practical attacks against Diffie-Hellman key exchange as currently used by TLS. However, these attacks rely on the ability to downgrade connections to export-grade cryptography. In this section we address the following question: how secure is Diffie-Hellman in broader practice, as used in other protocols that do not suffer from downgrade, and when applied with stronger groups?

To answer this question we must first examine how the number field sieve for discrete logarithms scales to 768- and 1024-bit groups. As we argue below, 768-bit groups in relatively widespread use are now within reach for academic computational resources. Additionally, performing precomputations for a small number of 1024-bit groups is plausibly within the resources of nation-state adversaries. The precomputation would likely require special-purpose hardware, but would not require any major algorithmic improvements. In light of these results, we examine several standard Internet security protocols — IKE, SSH, and TLS — to determine their vulnerability. Although the cost of the precomputation for a 1024-bit group is several times higher than for an RSA key of equal size, a one-time investment could be used to attack millions of hosts, due to widespread reuse of the most common Diffie-Hellman parameters. Finally, we apply this new understanding to a set of recently published documents to evaluate the hypothesis that the National Security Agency has *already* implemented such a capability.

4.1. Scaling NFS to 768- and 1024-bit Diffie-Hellman

Estimating the cost for discrete logarithm cryptanalysis at larger key sizes is far from straightforward due to the complexity of parameter tuning. We attempt estimates up to 1024-bit discrete logarithm based on the existing literature

and our own experiments but further work is needed for greater confidence. We summarize all the costs, measured or estimated in Table 2.

DH-768: done in 2016. When the ACM CCS version of this article was prepared, the latest discrete logarithm record was a 596-bit computation. Based on that work, and on prior experience with the 768-bit factorization record in 2009,¹² we made the conservative prediction that it was possible, as explained in Section 2, to put more computational effort into sieving for the discrete logarithm case than for factoring, so that the linear algebra step would run on a slightly smaller matrix. This led to a runtime estimate of around 37,000 core-years, most of which was spent on linear algebra.

This estimate turned out to be overly conservative, for several reasons. First, there have been significant improvements in our software implementation (Section 3.3). In addition, our estimate did not use the Joux-Lercier alternative polynomial selection method,¹¹ which is specific to discrete logarithms. For 768-bit discrete logarithms, this polynomial selection method leads to a significantly smaller computational cost.

In 2016, Kleinjung et al. completed a 768-bit discrete logarithm computation.¹³ While this is a massive computation on the academic scale, a computation of this size has likely been within reach of nation-states for more than a decade. This data is mentioned in Table 2.

DH-1024: Plausible with nation-state resources. Experimentally extrapolating sieving parameters to the 1024-bit case is difficult due to the trade-offs between the steps of the algorithm and their relative parallelism. The prior work proposing parameters for factoring a 1024-bit RSA key is thin, and we resort to extrapolating from asymptotic complexity. For the number field sieve, the complexity is $\exp((k + o(1))(\log N)^{1/3}(\log \log N)^{2/3})$, where N is the integer to factor or the prime modulus for discrete logarithm and k is an algorithm-specific constant. This formula is inherently imprecise, since the $o(1)$ in the exponent can hide polynomial factors. This complexity formula, with $k = 1.923$, describes the overall time for both discrete logarithm and factorization, which are both dominated by sieving and linear algebra in the precomputation. Evaluating the formula for 768- and 1024-bit N gives us estimated multiplicative factors by which time and space will increase from the 768- to the 1024-bit case.

Table 2. Estimating costs for factoring and discrete log*.

	Sieving		Linear Algebra		Descent	
	Log ₂ B	Core-years	Rows	Core-years	Core-time	
RSA-512	29	0.3	4.2mn	0.03		Timings with default CADO-NFS parameters.
DH-512	27	2.5	2.2mn	1.1	10min	For the computations in this paper; may be suboptimal.
RSA-768	37	800	250mn	100		Est. based on Kleinjung and Aoki et al. ¹² with less sieving.
DH-768	36	4,000	24mn	920	43hrs	Data from, Kleinjung and Diem et al. ¹³ , Table 1.
RSA-1024	42	≈1,000,000	≈8.7bn	≈120,000		Crude estimate based on complexity formula.
DH-1024	40	≈5,000,000	≈0.8bn	≈1,100,000	30 days	Crude estimate based on formula and our experiments.

* For sieving, we give one important parameter, which is the number of bits of the smoothness bound B . For linear algebra, all costs for DH are for safe primes; for Digital Signature Algorithm (DSA) primes with group order of 160 bits, this should be divided by 6.4 for 1024 bits, 4.8 for 768 bits, and 3.2 for 512 bits.

For 1024-bit precomputation, the total time complexity can be expected to increase by a factor of 1220 using the complexity formula, while space complexity increases by its square root, approximately 35. These ratios are relevant for both factorization and discrete logarithm since they have the same asymptotic behavior. For DH-1024, we get a total cost estimate for the precomputation of about 6mn core-years. In practice, it is not uncommon for estimates based merely on the complexity formula to be off by a factor of 10. Estimates of Table 2 must therefore be considered with due caution.

For 1024-bit descent, we experimented with our early-abort implementation to inform our estimates for descent initialization, which should dominate the individual discrete logarithm computation. For a random target in Oakley Group 2, initialization took 22 core-days, and yielded a few primes of at most 130 bits to be descended further. In twice this time, we reached primes of about 110 bits. At this point, we were certain to have bootstrapped the descent and could continue down to the smoothness bound in a few more core-days if proper sieving software were available. Thus we estimate that a 1024-bit descent would take about 30 core-days, once again easily parallelizable.

Costs in hardware. Although several million core-years is a massive computational effort, it is not necessarily out of reach for a nation-state. At this scale, significant cost savings could be realized by developing application-specific hardware given that sieving is a natural target for hardware implementation. To our knowledge, the best prior description of an Application-Specific Integrated Circuit (ASIC) implementation of 1024-bit sieving is the 2007 work of Geiselmann and Steinwandt.⁸ Updating their estimates for modern techniques and adjusting parameters for discrete logarithm allows us to extrapolate the financial and time costs.

We increase their chip count by a factor of ten to sieve more and save on linear algebra as above, giving an estimate of 3mn chips to complete sieving in one year. Shrinking the dies from the 130 nanometer technology node used in the paper to a more modern size reduces costs as transistors are cheaper at newer technologies. With standard transistor costs and utilization, it would cost about \$2 per chip to manufacture after fixed design and tape-out costs of roughly \$2mn.¹⁴ This suggests that an \$8mn investment would buy enough ASICs to complete the DH-1024 sieving precomputation in one year. Since a step of descent uses sieving, the same hardware could likely be reused to speed calculations of individual logarithms.

Estimating the financial cost for the linear algebra is more difficult since there has been little work on designing chips that are suitable for the larger fields involved in discrete logarithm. To derive a rough estimate, we can begin with general purpose hardware and the core-year estimate from Table 2. Using the 300,000 CPU core Titan supercomputer it would take four years to complete the 1024-bit linear algebra stage (notwithstanding the fact that estimates from Table 2 are known to be extremely coarse, and could be optimistic by a factor of maybe 10). Titan was constructed in 2012 for \$94mn, suggesting a cost of under \$400mn in supercomputers to finish this step in a year. In the context of factorization, moving linear algebra from general purpose CPUs to ASICs has been estimated to reduce costs by a factor

of 80.⁷ If we optimistically assume that a similar reduction can be achieved for discrete logarithm, the hardware cost to perform the linear algebra for DH-1024 in one year is plausibly on the order of \$5mn.

Combining these estimates, special-purpose hardware that can perform the precomputation for one 1024-bit group per year would cost roughly \$13mn. This is much less than the “hundreds of millions of dollars” that we conservatively estimated in 2015, making it even more likely that nation-state adversaries have implemented the attack.

To put this dollar figure in context, the FY 2012 budget for the U.S. Consolidated Cryptologic Program (which includes NSA) was \$10.5bn.²² The 2013 budget request, which prioritized investment in “groundbreaking cryptanalytic capabilities to defeat adversarial cryptography and exploit internet traffic” included notable \$100mn+ increases in two programs under Cryptanalysis & Exploitation Services: “Cryptanalytic IT Systems” (to \$247mn), and the cryptically named “PEO Program C” (to \$360mn).²²

4.2. Is NSA breaking 1024-bit Diffie-Hellman?

Our calculations suggest that it is plausibly within NSA’s resources to have performed number field sieve precomputations for a small number of 1024-bit Diffie-Hellman groups. This would allow them to break any key exchanges made with those groups in close to real time. If true, this would answer one of the major cryptographic questions raised by the Edward Snowden leaks: How is NSA defeating the encryption for widely used VPN protocols?

Virtual private networks are widely used for tunneling business or personal traffic across potentially hostile networks. We focus on the IPsec VPN protocol using the IKE protocol for key establishment and parameter negotiation and the Encapsulating Security Payload (ESP) protocol for protecting packet contents.

IKE. There are two versions, IKEv1 and IKEv2, which differ in message structure but are conceptually similar. For the sake of brevity, we will use IKEv1 terminology.¹⁰

Each IKE session begins with a Phase 1 handshake in which the client and server select a Diffie-Hellman group from a small set of standardized parameters and perform a key exchange to establish a shared secret. The shared secret is combined with other cleartext values transmitted by each side, such as nonces and cookies, to derive a value called SKEYID. In IKEv1, SKEYID also incorporates a Pre-Shared Key (PSK) used for authentication.

The resulting SKEYID is used to encrypt and authenticate a Phase 2 handshake. Phase 2 establishes the parameters and key material, KEYMAT, for protecting the subsequently tunneled traffic. Ultimately, KEYMAT is derived from SKEYID, additional nonces, and the result of an optional Phase 2 Diffie-Hellman exchange.

NSA’s VPN exploitation process. Documents published by Der Spiegel describe NSA’s ability to decrypt VPN traffic using passive eavesdropping and without message injection or man-in-the-middle attacks on IPsec or IKE. Figure 3 illustrates the flow of information required to decrypt the tunneled traffic.

When the IKE/ESP messages of a VPN of interest are collected, the IKE messages and a small amount of ESP

traffic are sent to the Cryptanalysis and Exploitation Services (CES).^{21, 23, 25} Within the CES enclave, a specialized “attack orchestrator” attempts to recover the ESP decryption key with assistance from high-performance computing resources as well as a database of known PSKs (“CORALREEF”).^{21, 23, 25} If the recovery was successful, the decryption key is returned from CES and used to decrypt the buffered ESP traffic such that the encapsulated content can be processed.^{21, 24}

Evidence for a discrete logarithm attack. The ability to decrypt VPN traffic does not necessarily indicate a defeat of Diffie-Hellman. There are, however, several features of the described exploitation process that support this hypothesis.

The IKE protocol has been extensively analyzed^{3,15} and is not believed to be exploitable in standard configurations under passive eavesdropping attacks. Absent a vulnerability in the key derivation function or transport encryption, the attacker must recover the decryption keys. This requires the attacker to calculate SKEYID generated from the Phase 1 Diffie-Hellman shared secret after passively observing an IKE handshake.

While IKE is designed to support a range of Diffie-Hellman groups, our Internet-wide scans (Section 4.3) show that the vast majority of IKE endpoints select one particular 1024-bit Diffie-Hellman group even when offered stronger groups. Conducting an expensive, but feasible, precomputation for this single 1024-bit group (Oakley Group 2) would allow the

efficient recovery of a large number of Diffie-Hellman shared secrets used to derive SKEYID and the subsequent KEYMAT.

Given an efficient oracle for solving the discrete logarithm problem, attacks on IKE are possible provided that the attacker can obtain the following: (1) a complete two-sided IKE transcript, and (2) any PSK used for deriving SKEYID in IKEv1. The available documents describe both of these as explicit prerequisites for the VPN exploitation process outlined above and provide the reader with internal resources available to meet these prerequisites.²³

Of course, this explanation is not dispositive and the possibility remains that NSA could defeat VPN encryption using alternative means. A published NSA document refers to the use of a router “implant” to allow decryption of IPsec traffic, indicating the use of targeted malware is possible. However, this implant “allows passive exploitation with just ESP”²³ without the prerequisite of collecting the IKE handshake messages. This indicates it is an alternative mechanism to the attack described above.

The most compelling argument for a pure cryptographic attack is the generality of NSA’s VPN exploitation process. This process appears to be applicable across a broad swath of VPNs without regard to endpoint’s identity or the ability to compromise individual endpoints.

4.3. Effects of a 1024-bit break

In this section, we use Internet-wide scanning to assess the impact of a hypothetical DH-1024 break on IKE, SSH, and HTTPS. Our measurements, performed in early 2015, indicate that these protocols would be subject to widespread compromise by a nation-state attacker who had the resources to invest in precomputation for a small number of 1024-bit groups.

IKE. We measured how IPsec VPNs use Diffie-Hellman in practice by scanning a 1% random sample of the public IPv4 address space for IKEv1 and IKEv2 (the protocols used to initiate an IPsec VPN connection) in May 2015. We used the ZMap UDP probe module to measure support for Oakley Groups 1 and 2 (two popular 768- and 1024-bit, built-in groups) and which group servers prefer. Of the 80K hosts that responded with a valid IKE packet, 44.2% were willing to negotiate a connection using one of the two groups. We found that 31.8% of IKEv1 and 19.7% of IKEv2 servers supported Oakley Group 1 (768-bit) while 86.1% and 91.0% respectively supported Oakley Group 2 (1024-bit). In our sample of IKEv1 servers, 2.6% of profiled servers preferred

Figure 3. NSA’s VPN decryption infrastructure. This classified illustration published by Der Spiegel²⁵ shows captured IKE handshake messages being passed to a high-performance computing system, which returns the symmetric keys for ESP session traffic. The details of this attack are consistent with an efficient break for 1024-bit Diffie-Hellman.

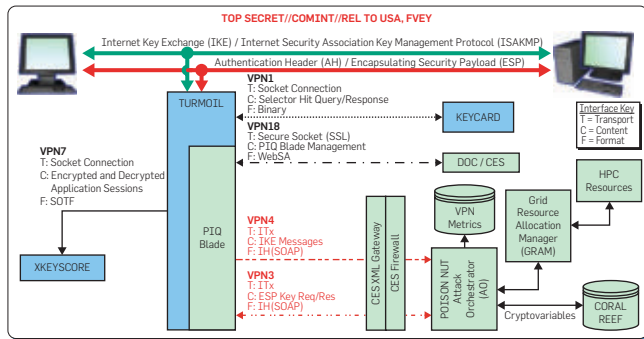


Table 3. Estimated impact of Diffie-Hellman attacks in early 2015^a.

	<i>Vulnerable servers, if the attacker can precompute for...</i>			
	All 512-bit groups	All 768-bit groups	One 1024-bit group	Ten 1024-bit groups
HTTPS Top Million w/ active downgrade	45,100 (8.4%)	45,100 (8.4%)	205,000 (37.1%)	309,000 (56.1%)
HTTPS Top Million	118 (0.0%)	407 (0.1%)	98,500 (17.9%)	132,000 (24.0%)
HTTPS Trusted w/ active downgrade	489,000 (3.4%)	556,000 (3.9%)	1,840,000 (12.8%)	3,410,000 (23.8%)
HTTPS Trusted	1,000 (0.0%)	46,700 (0.3%)	939,000 (6.56%)	1,430,000 (10.0%)
IKEv1 IPv4	-	64,700 (2.6%)	1,690,000 (66.1%)	1,690,000 (66.1%)
IKEv2 IPv4	-	66,000 (5.8%)	726,000 (63.9%)	726,000 (63.9%)
SSH IPv4	-	-	3,600,000 (25.7%)	3,600,000 (25.7%)

^a We used Internet-wide scanning to estimate the number of real-world servers for which typical connections could be compromised by attackers with various levels of computational resources. For HTTPS, we provide figures with and without downgrade attacks on the chosen ciphersuite. All others are passive attacks.

the 768-bit Oakley Group 1 and 66.1% preferred the 1024-bit Oakley Group 2. For IKEv2, 5.8% of profiled servers chose Oakley Group 1, and 63.9% chose Oakley Group 2.

SSH. All SSH handshakes complete either a finite field or elliptic curve Diffie-Hellman exchange. The protocol explicitly defines support for Oakley Group 2 (1024-bit) and Oakley Group 14 (2048-bit) but also allows a server-defined group to be negotiated. We scanned 1% random samples of the public IPv4 address space in April 2015. We found that 98.9% of SSH servers supported the 1024-bit Oakley Group 2, 77.6% supported the 2048-bit Oakley Group 14, and 68.7% supported a server-defined group.

During the SSH handshake, the server selects the client's highest priority mutually supported key exchange algorithm. To estimate what servers will prefer in practice, we performed a scan in which we mimicked the algorithms offered by OpenSSH 6.6.1p1, the latest version of OpenSSH. In this scan, 21.8% of servers preferred the 1024-bit Oakley Group 2, and 37.4% preferred a server-defined group. 10% of the server-defined groups were 1024-bit, but, of those, nearly all provided Oakley Group 2 rather than a custom group.

Combining these equivalent choices, we find that a nation-state adversary who performed NFS precomputations for the 1024-bit Oakley Group 2 could passively eavesdrop on connections to 3.6mn (25.7%) publicly accessible SSH servers.

HTTPS. Our 2015 scans found that DHE was commonly deployed on web servers. 68.3% of Alexa Top Million sites supported DHE, as did 23.9% of sites with browser-trusted certificates. Of the Top Million sites that supported DHE, 84% used a 1024-bit or smaller group, with 94% of these using one of five groups.

Despite widespread support for DHE, a passive eavesdropper can only decrypt connections that organically agree to use Diffie-Hellman. We estimated the number of sites for which this would occur by offering the same sets of ciphersuites as Chrome, Firefox, and Safari. We found that browser connections to approximately 24% of browser connections with HTTPS-enabled Top Million sites (and 10% of all sites with browser-trusted sites certificates) would negotiate DHE using one of the ten most popular 1024-bit primes. After completing the NFS precomputation for only the most popular 1024-bit prime, an adversary could passive eavesdrop on browser connections to 17.9% of Top Million sites.

5. RECOMMENDATIONS

In this section, we present concrete recommendations to recover the expected security of Diffie-Hellman.

Transition to elliptic curves

Transitioning to Elliptic Curve Diffie-Hellman (ECDH) key exchange avoids all known feasible cryptanalytic attacks. Current elliptic curve discrete logarithm algorithms do not gain as much of an advantage from precomputation. In addition, ECDH keys are shorter and computations are faster. We recommend transitioning to elliptic curves; this is the most effective solution to the vulnerabilities in this paper. We note that in August 2015, NSA announced that it was planning to transition away from elliptic curve cryptography

for its Suite B cryptographic algorithms and would replace them with algorithms resistant to quantum computers.¹⁶ However, since no fully vetted and standardized quantum-resistant algorithms exist currently, elliptic curves remain the most secure choice for public key operations.

Increase minimum key strengths

To protect against the Logjam attack, server operators should disable `DHE_EXPORT` and configure DHE ciphersuites to use primes of 2048 bits or larger. Browsers and clients should raise the minimum accepted size for Diffie-Hellman groups to at least 1024 bits in order to avoid downgrade attacks.

Don't deliberately weaken cryptography


The Logjam attack illustrates the fragility of cryptographic "front doors." Although the key sizes originally used in `DHE_EXPORT` were intended to be tractable only to NSA, two decades of algorithmic and computational improvements have significantly lowered the bar to attacks on such key sizes. Despite the eventual relaxation of cryptography export restrictions and subsequent attempts to remove support for `DHE_EXPORT`, the technical debt induced by the additional complexity has left implementations vulnerable for decades. Like FREAK,¹ our results warn of the long-term debilitating effects of deliberately weakening cryptography.

6. CONCLUSION

We find that Diffie-Hellman key exchange, as used in practice, is often less secure than widely believed. The problems stem from the fact that the number field sieve for discrete logarithms allows an attacker to perform a single precomputation that depends only on the group, after which computing individual logarithms in that group has a far lower cost. Although this is well known to cryptographers, it apparently has not been widely understood by system builders. Likewise, many cryptographers did not appreciate that a large fraction of Internet communication depends on a few small, widely shared groups.

A key lesson is that cryptographers and creators of practical systems need to work together more effectively. System builders should take responsibility for being aware of applicable cryptanalytic attacks. Cryptographers should involve themselves in how cryptography is actually being applied, such as through engagement with standards efforts and software review. Bridging the perilous gap that separates these communities will be essential for keeping future systems secure.

Acknowledgments

The authors thank Michael Bailey, Daniel Bernstein, Ron Dreslinski, Tanja Lange, Adam Langley, Kenny Paterson, Andrei Popov, Ivan Ristic, Edward Snowden, Brian Smith, Martin Thomson, and Eric Rescorla. This work was supported by the U.S. National Science Foundation, the Office of Naval Research, the European Research Council, and the French National Research Agency, with additional support from the Mozilla Foundation, Supermicro, Google, Cisco, the Morris Wellman Professorship, and the Alfred P. Sloan Foundation. Some experiments used the Grid'5000 testbed, supported by INRIA, CNRS, RENATER, and others. 

References

1. Beurdouche, B., Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Kohlweiss, M., Pironti, A., Strub, P.-Y., Zinzindohoue, J.K. A messy state of the union: Taming the composite state machines of TLS. In *IEEE Symposium on Security and Privacy* (2015).
2. Bouvier, C., Gaudry, P., Imbert, L., Jeljeli, H., Thomé, E. New record for discrete logarithm in a prime finite field of 180 decimal digits, 2014. <http://caramel.loria.fr/p180.txt>.
3. Canetti, R., Krawczyk, H. Security analysis of IKE's signature-based key-exchange protocol. In *Crypto* (2002).
4. Coppersmith, D. Solving linear equations over GF(2) via block Wiedemann algorithm. *Math. Comp.* 62, 205 (1994).
5. Diffie, W., Hellman, M.E. New directions in cryptography. *IEEE Trans. Inform. Theory* 22, 6 (1976), 644–654.
6. Durumeric, Z., Wustrow, E., Halderman, J.A. ZMap: Fast Internet-wide scanning and its security applications. In *Usenix Security* (2013).
7. Geiselmann, W., Kopfer, H., Steinwandt, R., Tromer, E. Improved routing-based linear algebra for the number field sieve. In *Information Technology: Coding and Computing* (2005).
8. Geiselmann, W., Steinwandt, R. Non-wafer-scale sieving hardware for the NFS: Another attempt to cope with 1024-bit. In *Eurocrypt* (2007).
9. Gordon, D.M. Discrete logarithms in GF(p) using the number field sieve. *SIAM J. Discrete Math.* 6, 1 (1993).
10. Harkins, D., Carrel, D. The Internet key exchange (IKE). RFC 2409 (Nov. 1998).
11. Joux, A., Lercier, R. Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the Gaussian integer method. *Math. Comp.* 72, 242 (2003), 953–967.
12. Kleinjung, T., Aoki, K., Franke, J., Lenstra, A.K., Thomé, E., Bos, J.W., Gaudry, P., Kruppa, A., Montgomery, P.L., Osvik, D.A., te Riele, H., Timofeev, A., Zimmermann, P. Factorization of a 768-bit RSA modulus. In *Crypto* (2010).
13. Kleinjung, T., Diem, C., Lenstra, A.K., Priplata, C., Stahlke, C. Computation of a 768-bit prime field discrete logarithm. In *EUROCRYPT* (2017).
14. Lipacis, M. Semiconductors: Moore stress = structural industry shift. Technical report, Jefferies, 2012.
15. Meadows, C. Analysis of the Internet key exchange protocol using the NRL protocol analyzer. In *IEEE Symposium on Security and Privacy* (1999).
16. National Security Agency. Cryptography today, August 2015. https://web.archive.org/web/20150905185709/https://www.nsa.gov/ia/programs/suiteb_cryptography/.
17. Orman, H. The Oakley key determination protocol. RFC 2412 (Nov. 1998).
18. Schirokauer, O. Virtual logarithms. *J. Algorithms* 57, 2 (2005), 140–147.
19. The CAD0-NFS Development Team. CAD0-NFS, an implementation of the number field sieve algorithm. <http://cado-nfs.gforge.inria.fr/>, 2017. Release 2.3.0.
20. Thomé, E. Subquadratic computation of vector generating polynomials and improvement of the block Wiedemann algorithm. *J. Symbolic Comput.* 33, 5 (2002), 757–775.
21. Fielded capability: End-to-end VPN SPIN 9 design review. Media leak. <http://www.spiegel.de/media/media-35529.pdf>.
22. FY 2013 congressional budget justification. Media leak. <https://cryptome.org/2013/08/spy-budget-fy13.pdf>.
23. Intro to the VPN exploitation process. Media leak, Sept. 2010. <http://www.spiegel.de/media/media-35515.pdf>.
24. SPIN 15 VPN story. Media leak. <http://www.spiegel.de/media/media-35522.pdf>.
25. TURMOIL VPN processing. Media leak, Oct. 2009. <http://www.spiegel.de/media/media-35526.pdf>.

David Adrian, Zakir Durumeric, J. Alex Halderman, Drew Springall, Benjamin VanderSloot, and Eric Wustrow, University of Michigan, Ann Arbor, MI, USA.

Karthikeyan Bhargavan, INRIA Paris-Rocquencourt, Paris, France.

Pierrick Gaudry, Emmanuel Thomé, and Paul Zimmermann, INRIA Nancy-Grand Est, CNRS, and Université de Lorraine, France.

Matthew Green, Johns Hopkins University, Baltimore, MD, USA.

Nadia Heninger and Luke Valenta, University of Pennsylvania, Philadelphia, PA, USA.

Santiago Zanella-Béguelin, Microsoft Research, Cambridge, England, UK.

Copyright held by authors/owners.



There is no silver bullet...
there IS information you
can USE.

EDITED BY
Per Larsen, *Immunant, Inc.*
Ahmad-Reza Sadeghi, *Technische Universität Darmstadt*

ISBN: 978-1-970001-80-8 DOI: 10.1145/3129743
<http://books.acm.org>
<http://www.morganclaypoolpublishers.com/acm>

Auburn University
Department of Computer Science and Software Engineering (CSSE)
Multiple Faculty Positions in Data Science & Engineering

Auburn CSSE invites applications from candidates specializing in all areas related to data: analytics, engineering, mining, science and techniques for massive data storage, querying and analysis to solve real-world problems. We seek candidates at the Assistant Professor level, however outstanding candidates at a senior level will also be considered. A Ph.D. degree in computer science, software engineering or a closely related field must be completed by the start of appointment. Excellent communication skills are required.

The department will offer a new joint (with the Department of Mathematics and Statistics) M.S. degree in Data Science & Engineering in fall 2019. Successful candidates will play an active role in this program as well as develop a nationally recognized and extramurally funded research program in Data Science & Engineering.

CSSE is home to the Auburn Cyber Research Center (<http://cyber.auburn.edu>), and is affiliated with the McCrary Institute for Critical Infrastructure Protection and Cyber Systems (<http://mccrary.auburn.edu>). The department currently has 21 full-time tenure-track and six teaching-track faculty members, who support strong undergraduate and graduate programs (M.S. in CSSE, M.S. in Cybersecurity Engineering and Ph.D. in CSSE). Faculty research areas include artificial intelligence, architecture, computational biology, computer science education, cybersecurity, data science, energy-efficient systems, human-computer interaction, Internet of Things, learning science, machine learning, modeling and simulation, multi-agent systems, networks, software engineering and wireless engineering. Further information may be found at the department's home page <http://www.eng.auburn.edu/csse>.

Auburn University is one of the nation's premier public land-grant institutions. It is ranked 52nd among public universities by U.S. News and World Report. The university is nationally recognized for its commitment to academic excellence, its positive work environment, its student engagement, and its beautiful campus. Auburn residents enjoy a thriving community, recognized as one of the "best small towns in America," with moderate climate and easy access to major cities or to beach and mountain recreational facilities. Situated along the rapidly developing I-85 corridor between Atlanta, Georgia, and Montgomery, Alabama, Auburn residents have access to excellent public school systems and regional medical centers.

Applicants should submit a cover letter, curriculum vita, research vision, teaching

philosophy, and names of three to five references at <http://aufacultypositions.peopleadmin.com/postings/3222>. There is no application deadline. The application review process will continue until successful candidates are identified. Selected candidates must be able to meet eligibility requirements to work legally in the United States at the time of appointment for the proposed term of employment. Auburn University is an Affirmative Action/Equal Opportunity Employer. It is our policy to provide equal employment opportunities for all individuals without regard to race, sex, religion, color, national origin, age, disability, protected veteran status, genetic information, sexual orientation, gender identity, or any other classification protected by applicable law.

Boston College
Assistant Professor of the Practice or Lecturer in Computer Science

The Computer Science Department of Boston College seeks to fill one or more non-tenure-track teaching positions, as well as shorter-term visiting teaching positions. All applicants should be committed to excellence in undergraduate education, and be able to teach a broad variety of undergraduate computer science courses. Faculty in longer-term positions will participate in the development of new courses that reflect the evolving landscape of the discipline.

Minimum requirements for the title of Assistant Professor of the Practice, and for the title of Visiting Assistant Professor, include a Ph.D. in Computer Science or closely related discipline. Candidates who have only attained a Master's degree would be eligible for the title of Lecturer, or Visiting Lecturer. See <https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html> for more information.

To apply go to
<http://apply.interfolio.com/54268>.
Application process begins October 1, 2018.

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at <http://www.bc.edu/offices/diversity>.

Boston College
Associate or Full Professor of Computer Science

Description:

The Computer Science Department of Boston College is poised for significant growth over the next several years and seeks to fill faculty positions at all levels beginning in the 2019-2020 academic year. Outstanding candidates in all areas will be considered, with a preference for those who demonstrate a potential to contribute to cross-disciplinary teaching and research in conjunction with the planned Schiller Institute for Integrated Science and Society at Boston College. See <https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html> and <https://www.bc.edu/bc-web/schools/mcas/sites/schiller-institute.html> for more information.

Qualifications:

A Ph.D. in Computer Science or a closely related discipline is required, together with a distinguished track record of research and external funding, and evidence of the potential to play a leading role in the future direction of the department, both in the recruitment of faculty and the development of new academic programs.

To apply go to <http://apply.interfolio.com/54226>.

Application process begins October 1, 2018.

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at <http://www.bc.edu/offices/diversity>.

Boston College
Tenure Track, Assistant Professor of Computer Science

The Computer Science Department of Boston College is poised for significant growth over the next several years and seeks to fill faculty positions at all levels beginning in the 2019-2020 academic year. Outstanding candidates in all areas will be considered, with a preference for those who demonstrate a potential to contribute to cross-disciplinary teaching and research in con-

junction with the planned Schiller Institute for Integrated Science and Society at Boston College. A Ph.D. in Computer Science or a closely related discipline is required for all positions. See <https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html> and <https://www.bc.edu/bc-web/schools/mcas/sites/schiller-institute.html> for more information.

Successful candidates for the position of Assistant Professor will be expected to develop strong research programs that can attract external research funding in an environment that also values high-quality undergraduate teaching.

Minimum requirements for all positions include a Ph.D. in Computer Science or closely related discipline, an energetic research program that promises to attract external funding, and a commitment to quality in undergraduate and graduate education.

To apply go to <https://apply.interfolio.com/54208>.

Application review begins October 1, 2018.

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at <http://www.bc.edu/offices/diversity>.

Case Western Reserve University Faculty Positions

The Department of Electrical Engineering and Computer Science at Case Western Reserve University invites applications for three faculty positions:

Tenure-Track Faculty Position in Data Science: While exceptional candidates in all areas of Computer and Data Sciences will be considered for this position, our priority areas include Big Data Management and Systems, Databases, Data Mining, and Machine Learning. While all ranks will be considered, preference will be given to candidates at the Assistant Professor level.

Tenure-Track Faculty Position in Cyber-Security: In conjunction with the Institute for Smart, Secure, and Connected Systems (ISSACS), we are seeking candidates with research interests including but not limited to: theory and algorithms (e.g., cryptography, secure computing, secure data analysis, data privacy), systems (e.g., secure networks, distributed systems, cloud and virtualized environments, mobile devices), and applications (e.g., security in Internet-of-Things, cyber-physical systems, health, computer forensics). While all ranks will be considered, preference will be given to candidates at the Associate or Full Professor level.

For the tenure-track positions, candidates for the junior positions should have potential for excellence in innovative research. Candidates for the senior positions should have an established

record of research excellence. All successful candidates are expected to develop a vibrant, high-quality externally sponsored research program, supervise graduate students, and interact and collaborate with faculty across the department and campus. Applicants should have a strong commitment to high quality teaching at the undergraduate and graduate levels. Candidates must have a Ph.D. in Computer Science or a closely related field. Current departmental strengths include Artificial Intelligence, Bioinformatics, Internet of Things, Machine Learning, Networks and Distributed Systems, Cyber-Security and Privacy, and Software Engineering, and successful candidates will be expected to be synergistic with these strengths.

Non-Tenure-Track Faculty Position in Computer Science: We are seeking applicants dedicated to curriculum development and teaching in foundational areas of Computer and Data Sciences, including introductory programming, discrete mathematics, data structures, data science, and computer systems. The rank of the candidate will be commensurate with experience. In addition to teaching, successful candidates are also expected to be involved in departmental service.

Applicants must submit (i) a cover letter, (ii) current curriculum vita, (iii) statement of research interests, (iv) statement of teaching interests, and (v) contact information for at least three references for a junior position and six references for a senior position. Applications will be reviewed starting immediately and will continue until the positions are filled.

Application materials may be sent by email to:

Faculty Search Committee
Dept. of Electrical Engineering and
Computer Science
Case Western Reserve University
c/o YoLonda Stiggers (yxs307@case.edu)
10900 Euclid Avenue, Glennan 321
Cleveland, OH 44106-7071

Founded in 1826, Case Western Reserve University is a highly ranked private research university located in Cleveland, Ohio. As a vibrant and up-and-coming city, Cleveland was named one of the top 15 best places to live in the US by timeout.com in 2016. The campus is in the heart of University Circle, a world-renowned area for its cultural vibrancy, hosting the Cleveland Museum of Art (the second highest ranked art museum in the country), Cleveland Orchestra, the Museum of Natural History, Cleveland Institute of Music, and the Cleveland Botanical Garden, as well as two world-class health institutions, The Cleveland Clinic and University Hospitals of Cleveland. With generous support from the Cleveland Foundation, Case Western Reserve University recently launched the Institute for Smart, Secure and Connected Systems and is an anchor partner in the IOT Collaborative.

In employment, as in education, Case Western Reserve University is committed to Equal Opportunity and Diversity. Women, veterans, members of underrepresented minority groups, and individuals with disabilities are encouraged to apply.

Case Western Reserve University provides reasonable accommodations to applicants with disabilities. Applicants requiring a reasonable accommodation for any part of the application

and hiring process should contact the Office of Inclusion, Diversity and Equal Opportunity at 216-368-8877 to request a reasonable accommodation. Determinations as to granting reasonable accommodations for any applicant will be made on a case-by-case basis.

Columbia Quantum Initiative at Columbia University Open Rank Faculty Positions in the School of Engineering and Applied Science

Columbia Engineering is pleased to invite applications for faculty positions in Quantum Science and Technology as part of the Quantum Initiative at Columbia University in the City of New York. Applications at all ranks will be considered. Areas of interest in computing, communication, and theoretical research include novel computation and communication approaches, programming paradigms, algorithms, and protocols for quantum information applications. Areas of interest in experimental research include novel physical phenomena, electronic/optical materials, devices, circuits and integrated systems for quantum communication, computing, sensing, and metrology. We are seeking researchers who can benefit from the highly multidisciplinary environment and the state-of-the-art shared facilities/infrastructure within Columbia University such as the Columbia Nano Initiative and the Data Science Institute. The candidate is expected to hold a full or joint appointment in the Departments of Computer Science, Electrical Engineering, Applied Physics and Applied Mathematics, Industrial Engineering and Operations Research, or Mechanical Engineering and is expected to contribute to the advancement of their field, the department(s) and the School by developing an original and leading externally funded research program, establishing strong collaborations in research and education with related disciplines such as Physics and Chemistry, and contributing to the undergraduate and graduate educational mission of the Department(s) and the School. Columbia fosters multidisciplinary research and encourages collaborations with academic departments and units across Columbia University.

Candidates must have a Ph.D. or its professional equivalent by the starting date of the appointment. Applicants for this position must demonstrate the potential to do pioneering research and to teach effectively. The school is especially interested in qualified candidates who can contribute, through their research, teaching, and/or service, to the diversity and excellence of the academic community.

For additional information and to apply, please see: <http://engineering.columbia.edu/faculty-job-opportunities>. Applications should be submitted electronically and include the following: curriculum-vitae including a publication list, a description of research accomplishments, a statement of research and teaching interests and plans, contact information for three experts who can provide letters of recommendation, and up to three pre/reprints of scholarly work. All applications received by February 1, 2019 will receive full consideration.

Applicants can consult <http://www.engineering.columbia.edu> for more information about the school.

If you would like to apply, please visit <http://pa334.peopleadmin.com/postings/1894>

Columbia University is an Equal Opportunity Employer / Disability / Veteran

The Harvard John A. Paulson School of Engineering and Applied Sciences (SEAS)

Tenured Professor in Computer Science

The Harvard John A. Paulson School of Engineering and Applied Sciences (SEAS) seeks applicants for a position at the tenured level in the area of Machine Learning, with an expected start date of July 1, 2019.

We seek a computer scientist whose research accomplishments include fundamental advances in machine learning.

Computer Science at Harvard is enjoying a period of substantial growth in numbers of students and faculty hiring, and in expanded facilities. We benefit from outstanding undergraduate and graduate students, world-leading faculty, an excellent location, significant industrial collaboration, and substantial support from the Harvard Paulson School. For more information, see <http://www.seas.harvard.edu/computer-science>.

The associated Center for Research on Computation and Society (<http://crs.seas.harvard.edu/>), Berkman Klein Center for Internet & Society (<http://cyber.harvard.edu/>), Data Science Initiative (<https://datascience.harvard.edu/>), and Institute for Applied Computational Science (<http://iacs.seas.harvard.edu>) foster connections among computer science and other disciplines throughout the university.

Candidates are required to have a doctoral degree in computer science or a related area.

Required application documents include a cover letter, CV, a statement of research interests, a teaching statement, and up to three representative papers. Candidates are also required to submit the names and contact information for at least three references. Applicants can apply online at <http://academicpositions.harvard.edu/postings/8609>.

University of Illinois at Chicago

Lecturer - Non-Tenure Track - Computer Science

The Computer Science Department at the University of Illinois at Chicago is seeking multiple full-time teaching faculty members to start Fall 2019. The lecturer teaching track is a long-term career track that starts with the Lecturer position, and offers opportunities for advancement to Senior Lecturer. Candidates would be working alongside 13 full-time teaching faculty with over 150 years of combined teaching experience and 12 awards for excellence. The department seeks candidates dedicated to teaching; candidates must have evidence of effective teaching, or present a convincing case of future dedication and success in the art of teaching. Content areas of interest include introductory programming, data structures, computer organization/systems, web development, data science, software engineering, and machine learning. The standard teaching load is 2-3 undergraduate courses per semester (depending on course enrollment).

The University of Illinois at Chicago (UIC) is one of the top-10 most diverse universities in the US (US News and World Report), a top-10 best value (Wall Street Journal and Times Higher Education) and a hispanic serving institution. UIC's hometown of Chicago epitomizes the modern, livable, vibrant city. Located on the shore of Lake Michigan, Chicago offers an outstanding array of cultural and culinary experiences. As the birthplace of the modern skyscraper, Chicago boasts one of the world's tallest and densest skylines, combined with an 8100-acre park system and extensive public transit and biking networks.

Minimum qualifications include an MS in Computer Science or a closely related field or appropriate graduate degrees for specific course material (e.g., computer ethics), and either (a) demonstrated evidence of effective teaching, or (b) convincing argument of future dedication and success in the art of teaching. Applications are submitted online at <https://jobs.uic.edu/>. In the online application, include a curriculum vitae, names and addresses of at least three references, a statement providing evidence of effective teaching, and a statement describing your past experience in activities that promote diversity and inclusion (or plans to make future contributions), and recent teaching evaluations. For additional information contact Professor Mitch Theys, Committee Chair, mtheys@uic.edu.

For fullest consideration, please apply by December 14, 2018. We will continue to accept and review applications until the positions are filled. The University of Illinois is an Equal Opportunity, Affirmative Action employer. Minorities, women, veterans and individuals with disabilities are encouraged to apply. The University of Illinois conducts background checks on all job candidates upon acceptance of contingent offer of employment. Background checks will be performed in compliance with the Fair Credit Reporting Act.

Requirements

Minimum qualifications include an MS in Computer Science or a closely related field or appropriate graduate degrees for specific course material (e.g., computer ethics), and either (a) demonstrated evidence of effective teaching, or (b) convincing argument of future dedication and success in the art of teaching.

University of Maryland, Baltimore County

Computer Science and Electrical Engineering Multiple Tenured/Tenure-Track Positions in Computer Science and Computer Engineering

UMBC's Department of Computer Science and Electrical Engineering invites applications for multiple, open rank, tenured/tenure-track positions in Computer Science (CS) and Computer Engineering (CE) to begin in the Fall of 2019. Applicants should have or be completing a Ph.D. in a relevant discipline, have demonstrated the ability to pursue a research program, and have a strong commitment to undergraduate and graduate teaching.

We welcome candidates in all areas of specialization. Some areas of interest for CS applicants include but are not limited to: information assurance and cybersecurity; mobile, wearable,

and IoT systems; big data with an emphasis on machine learning, data science, brain-inspired methods, and high-performance computing; knowledge and database systems; visualization.

Some areas of interest for CE applicants include but are not limited to: hardware focused applicants in Digital, Analog, Mixed-mode VLSI design and test, integrated sensors and processing, SoC, new and emerging design technologies, hardware implementations for neuroscience and health-related wearables, cyber physical systems, hardware security and assurance.

The CSEE department is research-oriented and multi-disciplinary, with programs in Computer Science, Computer Engineering, Electrical Engineering, Data Science, and Cybersecurity. Our faculty (33 tenure-track, 10 teaching and 18 research) enjoy collaboration, working across our specializations as well as with colleagues from other STEM, humanities and the arts departments and external partners. We have more than 2000 undergraduate and 560 M.S. and Ph.D. students in our programs.

UMBC is a dynamic public research university integrating teaching, research and service. The 2018 US News and World Report Best Colleges report placed UMBC 7th in the *Most Innovative National Universities* category and 13th in *Best Undergraduate Teaching, National Universities*. Our strategic location in the Baltimore-Washington corridor is close to many federal laboratories and agencies and high-tech companies, facilitating interactions, collaboration, and opportunities for sabbaticals and visiting appointments.

Applicants should submit a cover letter, statement of teaching and research experience and interests, CV, and three letters of recommendation at <http://apply.interfolio.com/57564>. Candidates who are under consideration for an on-campus interview will be required to submit a commitment to inclusive excellence statement, which can be submitted as part of the initial application. For full consideration submit application materials by December 15, 2018. Applications will be accepted until the position is filled. Send questions to jobsTT@csee.umbc.edu and see <http://csee.umbc.edu/jobs> for more information. UMBC is an affirmative action/equal opportunity employer.

University of Memphis

Assistant Professor

The Department of Computer Science at the University of Memphis is seeking candidates for an Assistant Professor position beginning Fall 2019. The candidate's research will be jointly supported by the Department of Computer Science and the Institute of Intelligent Systems (IIS). Focus area for this position include Machine Learning, Data Mining, and Big Data. Candidates whose research areas complement the language & discourse or learning focus area of the IIS are particularly encouraged to apply. Candidates from minority and underrepresented groups are highly encouraged to apply. Successful candidates are expected to develop externally sponsored interdisciplinary research programs, teach both undergraduate and graduate courses and provide academic advising to students at all levels.

Applicants should hold a PhD in Computer Science, or related discipline, and be commit-

ACM Transactions on Social Computing



ACM TSC seeks to publish work that covers the full spectrum of social computing including theoretical, empirical, systems, and design research contributions. TSC welcomes research employing a wide range of methods to advance the tools, techniques, understanding, and practice of social computing, particularly research that designs, implements or studies systems that mediate social interactions among users, or that develops theory or techniques for application in those systems.



For further information
or to submit your
manuscript,
visit tsc.acm.org

ted to excellence in both research and teaching. Salary is highly competitive and dependent upon qualifications.

The Department of Computer Science (www.cs.memphis.edu) offers B.S., M.S., and Ph.D. programs as well as graduate certificates in Data Science and Information Assurance, and participates in an M.S. program in Bioinformatics (through the College of Arts and Sciences). The Department has been ranked 55th among CS departments with federally funded research. The Department regularly engages in large-scale multi-university collaborations across the nation. For example, CS faculty led the NIH-funded Big Data “Center of Excellence for Mobile Sensor Data-to-Knowledge (MD2K)” and the “Center for Information Assurance (CfIA)”.

The Institute for Intelligent Systems consists of 54 faculty members across 14 departments including Communication Sciences and Disorders, Computer Science, Engineering, Education, Linguistics, Philosophy and Psychology. The IIS offers a graduate certificate in Cognitive Science, a minor in Cognitive Science, and is affiliated with BA and MS programs in other departments. The IIS receives \$4-5 million in external awards per year from federal agencies such as NSF, IES, DoD, and NIH. Further information about the Institute for Intelligent Systems can be found at <http://iis.memphis.edu>.

Known as America’s distribution hub, Memphis ranked as America’s 6th best city for jobs by Glassdoor in 2017. Memphis metropolitan area has a population of 1.3 million. It boasts a vibrant culture and has a pleasant climate with an average temperature of 63 degrees.

Screening of applications begins immediately. For full consideration, application materials should be received by January 7, 2019. However, applications will be accepted until the search is completed.

To apply, please visit <https://workforum.memphis.edu/postings/20504>. Include a cover letter (please include a reference to this position as “CS-IIS”), curriculum vitae, statement of teaching philosophy, research statement, and three letters of recommendation. Direct all inquiries to Corinne O’Connor (cconnor2@memphis.edu).

A background check will be required for employment. The University of Memphis is an Equal Opportunity/Equal Access/Affirmative Action employer committed to achieving a diverse workforce.

University of South Carolina College of Engineering and Computing Multiple Open-Rank, Tenured or Tenure-Track Faculty Positions

The Department of Computer Science and Engineering (<http://cse.sc.edu>) seeks multiple tenured and tenure-track faculty members at all ranks and in all areas for Fall 2019. Applicants are expected to:

- ▶ Possess a Ph.D. degree in computer science, computer engineering, or a closely-related field by the beginning date of employment, and a demonstrated record of research accomplishments.
- ▶ Demonstrate evidence of commitment to diversity, equity, and inclusion through research, teaching, and/or service efforts.
- ▶ Develop internationally-recognized, externally-funded research programs that: (1) complement

existing departmental strengths, (2) leverage exceptional interdisciplinary collaboration opportunities, and (3) align with vital college-level, cross-cutting research themes including smart & connected communities, transformative computing, healthcare transformations, and agile manufacturing (for details on these initiatives, please visit: <http://cec.sc.edu/employment>).

Applicants from all traditional as well as non-traditional and interdisciplinary areas of Computer Science and Engineering are urged to apply. Research areas of special interest include:

- ▶ Human in the loop or knowledge-enhanced AI, deep learning, natural language processing, question-answering/conversational AI, brain-inspired computing, semantic/cognitive/perceptual computing;
- ▶ Big data - including social, sensor, biological, and health - and scalable computing/analysis of big data;
- ▶ Computer vision, robotics, and human-computer interaction Including personal digital/assistive technology;
- ▶ Cyber-physical systems and Internet of Things;
- ▶ Software analysis and testing, adaptive and autonomous systems, and search-based software engineering; and
- ▶ Next generation networking, cybersecurity, and privacy

The Department of Computer Science and Engineering offers B.S. degrees in Computer Science, Computer Information Systems, and Computer Engineering; M.S. and Ph.D. degrees in Computer Science and Computer Engineering; M.S. degrees in Software Engineering and Information Security; and a Graduate Certificate in Cyber Security Studies. The Department has 23 full-time faculty members (10 of whom are NSF CAREER Award recipients), an undergraduate enrollment of 935 students, and a graduate enrollment of 161 students.

Review of applications will begin on December 1, 2018 and continue until positions are filled. Expected start date is August 16, 2019. Interested applicants should apply online at <http://uscjobs.sc.edu/postings/43854> with a: (1) letter of intent, (2) curriculum vitae, (3) concise description of research plans, (4) teaching plan, and (5) names & contact information of 3-5 references.

The University of South Carolina does not discriminate in educational or employment opportunities on the basis of race, color, religion, national origin, sex, sexual orientation, gender, age, disability, protected veteran status or genetics.

[CONTINUED FROM P. 120] can then hypothesize that jar J1 corresponds to package 133 and will continue to choose two more pills from J1. The consumer checks these pills against the intended colors from package 133. If both are indeed the intended color, then J1 consists of real pills with probability $\frac{3}{4}$. In this case, the consumer can turn to jar J2, and if the colors of two selected pills correspond to package 152, then, with probability $\frac{3}{4}$, J2 is also good.

If the pills from J1 reveal a mismatch with 133, the consumer would then declare J1 to be fake. Now J2 could be 133 or 152 or could again be fake. From J2, the consumer chooses pill numbers where 133 and 152 differ in their intended color (such as pill 5 and pill 7). If pill 1 from J2 is consistent with package 133, the consumer should then check pill 5 and pill 7 for consistency with 133. Otherwise, if pill 1 from J2 is consistent with the colors from package 152, then the consumer should check pill 5 and pill 7 for consistency with 152 next.

However, consumers can sometimes be even more absent-minded. Suppose, for example, a consumer mixes the pills from package 133 and package 152 together in one jar, as in the figure here.

This time, the consumer has bought the packages in different cities so can legitimately assume that at most one of them is fake. The consumer also assumes that every self-respecting counterfeiter would put harmless and impotent fake pills in a fake pill bottle. So the jar has two identical-looking pills numbered 1, two identical-looking pills numbered 2, . . . , two identical-looking pills numbered 10. With the same requirement that the consumer should never take more than one real pill per day, what strategy of pill-taking can be used so that, after four days, the consumer knows, with a probability $\geq \frac{3}{4}$, that the consumer will be able to take exactly one real pill (and possibly one or more fake ones) over the subsequent eight days?

Solution. The consumer takes one pair of pills with the same number and color as, say, the pills numbered 2. If either pill is blue, then one of the two packages is fake. The consumer does not know which one, but it does

Counterfeiters produce and sell packages full of fakes (usually simple sugar pills) in a \$100 billion-per-year worldwide business.

not matter, and can subsequently take both pills with the same number from then on every day (such as the next day), take the two pills numbered 1, then the two pills numbered 3, and so on. Because the consumer knows at least one package is good, the consumer will be getting exactly one real pill every day, as desired.

On the other hand, if both pills numbered 2 are red after day two, then the consumer takes the numbered 5 pills on days three and four, respectively. If there is one red and one blue pill, the consumer then knows with probability $\frac{3}{4}$ that both packages are good, so can take the rest of the pills, one per day, preferring pairs with the same color, as with, say, pills 3, 4, 6, 8, and 10, unless a mismatch is found in which case the consumer can take the pairs with the same number from then on.

Upstart. More colors help, but more packages hurt. Suppose there are c colors of pills, and k packages have been put in the same jar and at most $f < k$ can be fake. What is a good consumer strategy for taking at most one real pill per day and having a high probability of a real one every day, too?

All are invited to submit their solutions to upstartpuzzles@cacm.acm.org; solutions to upstarts and discussion will be posted at <http://cs.nyu.edu/cs/faculty/shasha/papers/cacmpuzzles.html>

Dennis Shasha (dennisshasha@yahoo.com) is a professor of computer science in the Computer Science Department of the Courant Institute at New York University, New York, USA, as well as the chronicler of his good friend the omniheurist Dr. Ecco.

Copyright held by author.

ACM Transactions on Reconfigurable Technology and Systems



ACM TRETs is a peer-reviewed and archival journal that covers reconfigurable technology, systems, and applications on reconfigurable computers. Topics include all levels of reconfigurable system abstractions and all aspects of reconfigurable technology including platforms, programming environments and application successes.



For further information or to submit your manuscript, visit tret.s.acm.org



DOI:10.1145/3293576

Dennis Shasha

Upstart Puzzles

Randomized Anti-Counterfeiting

CONSUMERS AND PHARMACEUTICAL companies have been known to disagree over drug prices but have at least one interest in common: Neither wants the consumer to consume counterfeit drugs. The drug companies do not want to lose the sales, and consumers may have a critical need for the drug they paid for.

Counterfeiters have other ideas, however, so produce and sell packages full of fakes (usually simple sugar pills) in a \$100 billion-per-year worldwide business. The drug companies have fought such fakery by incorporating special packaging (holograms, unique numbers, sometimes even electronic tags) on the drug containers. With so much money to gain by selling sugar pills for high prices, however, the counterfeiters have managed to copy the packaging very expertly.

A clever but so far fictitious drug company has implemented the following random algorithm-style invention: Give each drug package (or bottle) a unique identifier, number each pill within a package—1, 2, 3, . . . —and insert an innocuous food coloring, from a palette of at least two colors, inside each pill. The food coloring is invisible until the pill is consumed.

Now suppose each package receives a random sequence of red or blue food colors, with equal probability. For concreteness, suppose package 133 has colors in the following numeric sequence

1: red, 2: red, 3: blue, 4: blue, 5: blue, 6: red, 7: blue, 8: red, 9: red, 10: blue

Package 152 has

1: blue, 2: red, 3: blue, 4: blue, 5: red, 6: red, 7: red, 8: red, 9: blue, 10: blue



While the colors are invisible before consumption, the consumer sees the color after taking the pill. A consumer worried about counterfeiting can log into a drug company website and, upon demonstrating some proof of purchase, look up the package number to see the pills associated with each number (such as, 1: red, 2: red, 3: blue, . . . for package 133). Once the consumer starts taking the pills, the consumer can compare the pill's color with the one intended for that pill number in that package. If even a single color does not match, that pill, as well as all the other pills in the package, should be presumed fake. Because the counterfeiter would have to guess the coloring, after only two pills have been consumed, the consumer has probability $1 - ((1/2) \times (1/2))$ or $3/4$ probability of knowing the package is fake.

But mistakes happen. For example, suppose the consumer has opened

packages 133 and 152 and separated the pills into two jars—J1 and J2—but does not remember which jar corresponds to which package. The consumer does not want to take two real pills on any particular day (they can be toxic in high doses) and wants to, of course, avoid taking only fake pills any day.

Problem. How can the consumer still determine whether the pills in each of the two jars are fake with a probability of $3/4$ for each jar after taking at most six pills altogether?

Solution. The consumer picks a number i between 1 and 10, such that the intended color of pill i from package 133 differs from pill i from package 152. In our example, i could be 1 because pill 1 in package 133 should be red, and pill 1 in package 152 should be blue. If the consumer picks pill 1 from jar J1 and it is red, the consumer

[CONTINUED ON P. 119]

Signal Processing, Architectures, and Detection of Emotion and Cognition

The Handbook of Multimodal-Multisensor Interfaces, Volume 2

The first authoritative resource on the dominant paradigm for new computer interfaces: users input involving new media (speech, multi-touch, hand and body gestures, facial expressions, writing) embedded in multimodal-multisensor interfaces that often include biosignals. This second volume begins with multimodal signal processing, architectures, and machine learning. It includes deep learning approaches for processing multisensorial and multimodal user data and interaction, as well as context-sensitivity. It focuses attention on how interfaces are most likely to advance human performance during the next decade.

Edited by Sharon Oviatt et al

ISBN: 978-1-970001-686

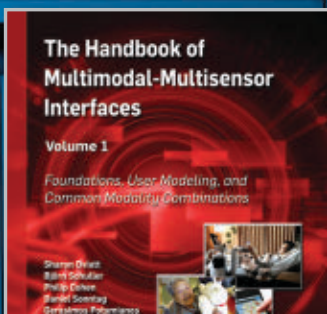
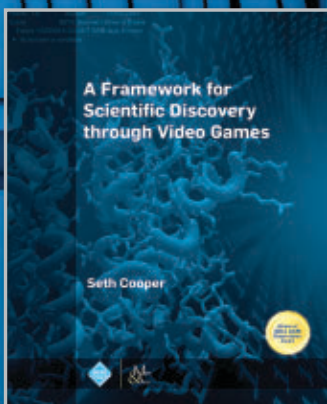
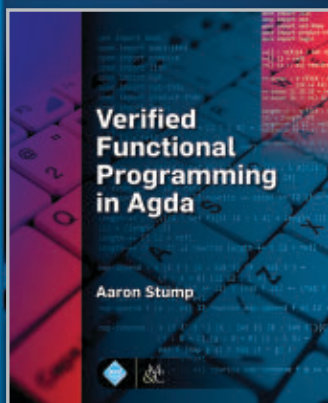
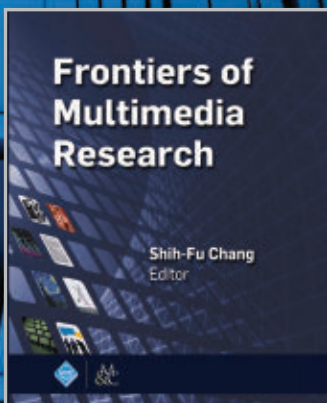
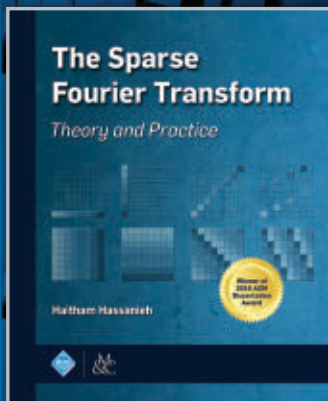
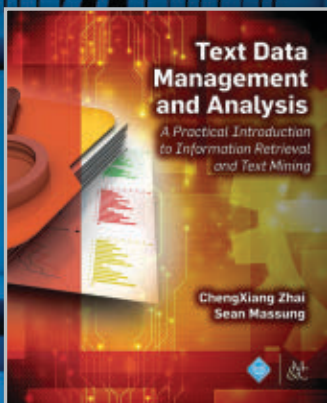
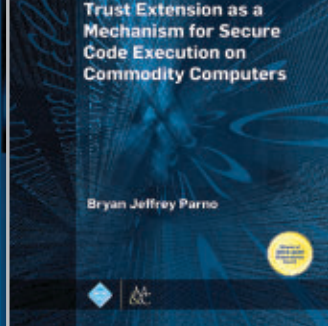
DOI: 10.1145/3107990

<http://books.acm.org>

<http://www.morganclaypoolpublishers.com/acm>



ACM BOOKS



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
2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA
Phone: +1-212-626-0658 Email: acmbooks-info@acm.org