

COMMUNICATIONS

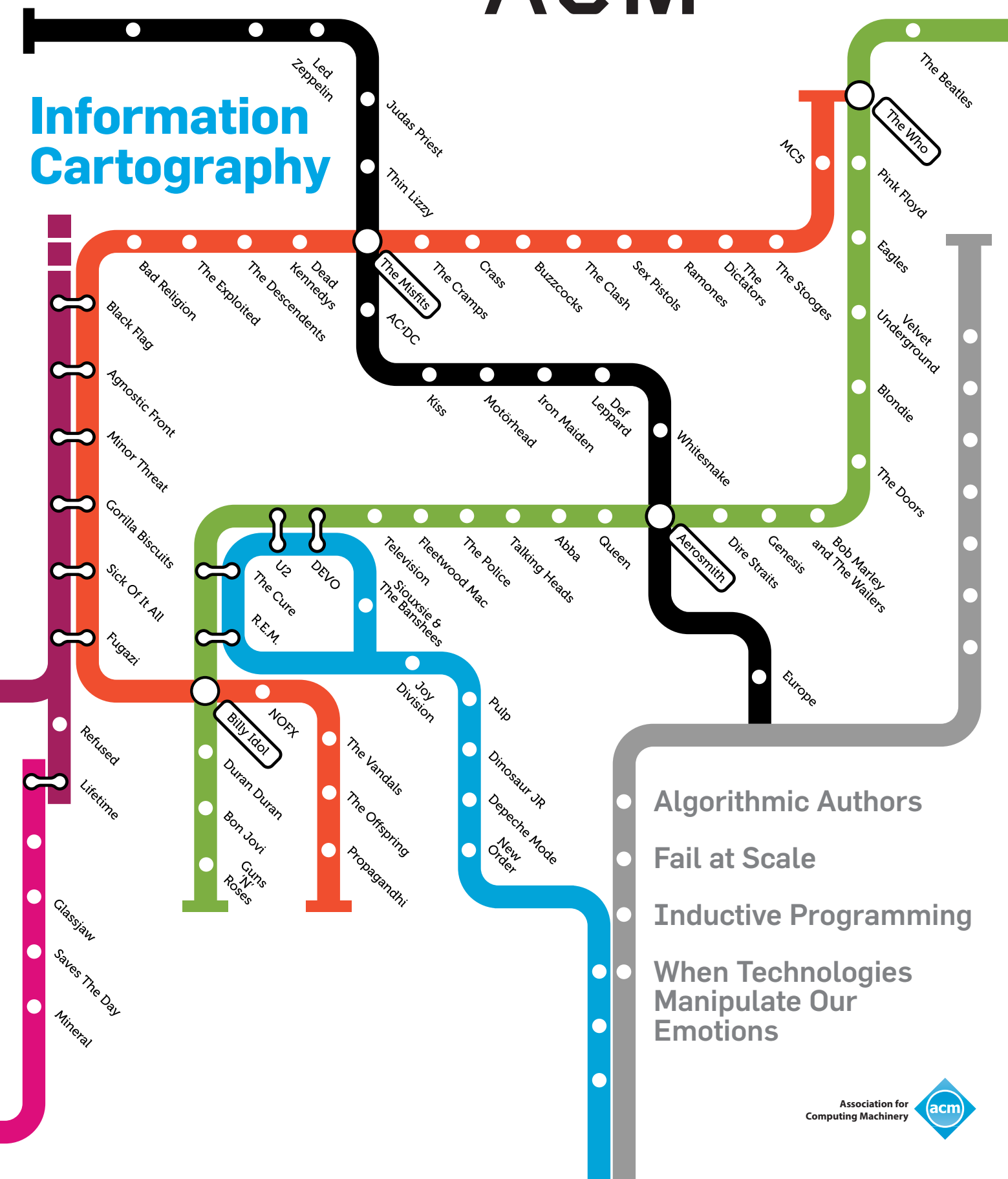
CACM.ACM.ORG

OF THE

ACM

11/2015 VOL.58 NO.11

Information Cartography



Algorithmic Authors

Fail at Scale

Inductive Programming

When Technologies
Manipulate Our
Emotions

30th IEEE INTERNATIONAL Parallel and Distributed Processing SYMPOSIUM

www.ipdps.org



MAY 23-27, 2016
CHICAGO HYATT REGENCY
CHICAGO, ILLINOIS USA

IPDPS 2016 WORKSHOPS: The IPDPS event offers a rich program of workshops which are “bookends” to the three-day technical program of contributed papers, keynote speakers, roundtable workshops, a PhD student forum, and industry participation. Held on the **first and last days of the conference**, workshops provide the IPDPS community an opportunity to explore special topics and to present work that is more preliminary or cutting-edge than the more mature research presented in the main symposium. Each workshop has its own website and submission requirements, and each is linked from the IPDPS Website at www.ipdps.org. The call for papers submission deadline for most workshops is after the main conference author notification date in December.

IPDPS WORKSHOPS- MONDAY 23 MAY 2016 (Check final schedule)

HCW	Heterogeneity in Computing Workshop
RAW	Reconfigurable Architectures Workshop
HIPS	High-Level Parallel Programming Models & Supportive Environments
HiCOMB	High Performance Computational Biology
APDCM	Advances in Parallel and Distributed Computing Models
ASHES (+ PLC)	Accelerators and Hybrid Exascale Systems
NIDISC	Nature Inspired Distributed Computing
PCO	Parallel Computing and Optimization
GABB	Graph Algorithms Building Blocks
EduPar	NSF / TCPP Parallel and Distributed Computing Education
HTBDC	High Throughput Big Data Computing
HPDAV	High Performance Data Analysis and Visualization
VarSys	Variability in Parallel and Distributed Systems

IPDPS WORKSHOPS- FRIDAY 27 MAY 2016 (Check final schedule)

HPPAC	High-Performance, Power-Aware Computing
PDSEC	Parallel and Distributed Scientific and Engineering Computing
DPDNS	Dependable Parallel, Distributed and Network-Centric Systems
LSP	Large-Scale Parallel Processing
ParLearning	Parallel and Distributed Computing for Large Scale Machine Learning and Big Data Analytics
JSSPP	Job Scheduling Strategies for Parallel Processing
iWAPT	International Workshop on Automatic Performance Tuning
CHIOW	Chapel Implementers and Users Workshop
HPBDC	High-Performance Big Data Computing
HPCMASPA	Monitoring and Analysis for High Performance Computing Systems Plus Applications
IPDRM	Emerging Parallel and Distributed Runtime Systems and Middleware
ParSocial	Parallel and Distributed Processing for Computational Social Systems

The 30th IEEE IPDPS will be held in Chicago, the third largest metropolis in the USA, with a dual hub airport system and 46 non-stop global flights daily. The Hyatt Regency, located in downtown Chicago along the shoreline of Lake Michigan, is close to all the attractions the city offers, including parks, beaches, shopping, museums, live entertainment, and the singular international cuisine of Chicago’s diverse ethnic neighborhood eateries.

GENERAL CHAIR

Xian-He Sun (Illinois Institute of Technology, USA)

PROGRAM CHAIR

Jeffrey K. Hollingsworth (University of Maryland, USA)

WORKSHOPS CHAIR

Bora Uçar (CNRS and ENS Lyon, France)

PHD FORUM CO-CHAIRS

Luc Bougé (ENS Cachan, France)
Trilce Estrada (University of New Mexico, USA)

INDUSTRY CO-CHAIRS

Kalyana Chandalavada (Intel, USA)
Narayan Desai (Ericsson, USA)

ROUNDTABLE WORKSHOPS: New in 2016, these condensed workshops to be held on Tuesday and Thursday will be organized and animated by a few people and will focus on emerging areas of interest to IPDPS attendees, especially topics that complement and “round out” the areas covered by the regular workshops.

PHD FORUM & STUDENT MENTORING: This program will include the traditional poster presentations by PhD students and will provide coaching in scientific writing and presentation skills and give all student participants an opportunity to hear from and interact with senior researchers attending the conference.

INDUSTRY PARTICIPATION: Chicago area companies are invited to share in the benefits of associating with an international community of top researchers and practitioners in fields related to parallel processing and distributed computing.

SPONSORED BY:

IEEE computer society
Technical Committee on Parallel Processing

IN COOPERATION WITH:



ACM SIGARCH, IEEE Computer Society Technical Committee on Computer Architecture,
IEEE Computer Society Technical Committee on Distributed Processing

Inviting Young Scientists



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 Heidelberg Laureate Forum is **September 18–23, 2016** 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?

Application deadline—**February 3, 2016**.

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

Successful applicants will be notified by **end of March/early April 2016**.

More information available on Heidelberg social media



Departments

- 5 **CEO's Letter**
A Tale of ACM Visions
By Bobby Schnabel
-
- 7 **Cerf's Up**
Memory with a Twist
By Vinton G. Cerf
-
- 8 **Letters to the Editor**
Who Will Read PACM?
-
- 10 **BLOG@CACM**
How Men Can Help Women in CS; Winning 'Computing's Nobel Prize'
Valerie Barr suggests how men can better promote women in computing, while Michael Stonebraker shares how it feels to receive the top honor in the computer science field.
-
- 31 **Calendar**
-
- 123 **Careers**

Last Byte

- 136 **Upstart Puzzles**
Auction Triplets
By Dennis Shasha

News



- 12 **Algorithmic Authors**
Natural Language Generation software begins delivering on the promise of automated prose.
By Alex Wright
-
- 15 **Companies Proactively Seek Out Internal Threats**
Organizations must balance their concerns with the protection of employee privacy.
By Esther Shein
-
- 18 **Brain Science Helps Computers Separate Speakers in a Crowded Room**
People can listen to a single voice amid the hubbub of a cocktail party; algorithms can help computers do it, too.
By Chris Edwards
-
- 21 **John H. Holland 1929–2015**
-
- 24 **Privacy and Security**
Security Assurance
How can customers tell they are getting it?
By Steven B. Lipner
-
- 27 **Legally Speaking**
Software Patents Are Falling Down
Implications of the U.S. Supreme Court's new test for patentable subject matter.
By Pamela Samuelson
-
- 30 **Economic and Business Dimensions**
Electronic Health Records and Patient Safety
Examining the effects of electronic health records on the safety of patients in medical facilities.
By Muhammad Zia Hydari, Rahul Telang, and William M. Marella
-
- 33 **Education**
Programming in K–12 Science Classrooms
Introducing students to visual programming as a pathway to text-based programming.
By Pratim Sengupta, Amanda Dickes, Amy Voss Farris, Ashlyn Karan, David Martin, and Mason Wright
-
- 36 **Interview**
An Interview with Fred Brooks
ACM Fellow and A.M. Turing Award recipient Fred Brooks reflects on his career.
By Len Shustek
-
- 41 **Viewpoint**
When Technologies Manipulate Our Emotions
Implications of the Facebook emotions study.
By Rafael A. Calvo, Dorian Peters, and Sidney D'Mello



Association for Computing Machinery
Advancing Computing as a Science & Profession

Practice



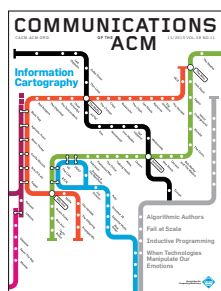
44

- 44 **Fail at Scale**
Reliability in the face of rapid change.
By Ben Maurer

- 50 **It Probably Works**
Probabilistic algorithms are all around us. Not only are they acceptable, some programmers actually seek out chances to use them.
By Tyler McMullen

- 55 **Componentizing the Web**
We may be on the cusp of a new revolution in Web development.
By Taylor Savage

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



About the Cover:
This month's cover story navigates the methodology of creating structured summaries of mountains of data. Metro maps untangle the data and tell compelling stories. Our cover, for example, is gleaned from an iconic metro map of history's most influential rock bands. For more information about the artist, Alberto Antoniazzi, and the project, see p. 62 and <http://www.rocknrollmetromap.com/>

Contributed Articles



74

- 62 **Information Cartography**
A metro map can tell a story, as well as provide good directions.
By Dafna Shahaf, Carlos Guestrin, Eric Horvitz, and Jure Leskovec



Watch the authors discuss their work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/information-cartography>

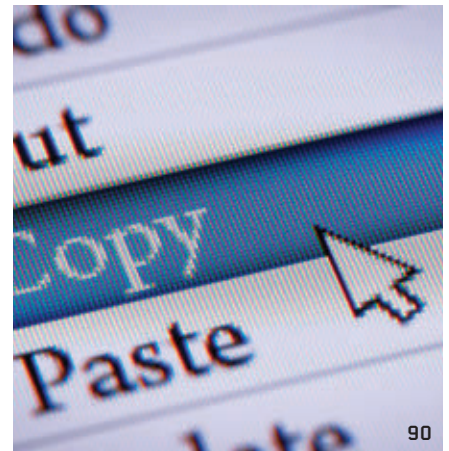
- 74 **Why People Post Benevolent and Malicious Comments Online**
Explaining motivations for online comments, this study looks to help establish a positive, nonthreatening online comment culture.
By So-Hyun Lee and Hee-Woong Kim



Watch the authors discuss their work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/why-people-post-benevolent-and-malicious-comments-online>

- 80 **Hidden In-Game Intelligence in NBA Players' Tweets**
Sentiment analysis helps detect NBA players' pre-game moods from their tweets and predict their on-court performance.
By Chenyan Xu, Yang Yu, and Chun-Keung Hoi

Review Articles



90

- 90 **Inductive Programming Meets the Real World**
Inductive programming can liberate users from performing tedious and repetitive tasks.
By Sumit Gulwani, José Hernández-Orallo, Emanuel Kitzelmann, Stephen H. Muggleton, Ute Schmid, and Benjamin Zorn

Research Highlights

- 102 **Technical Perspective**
The Path to Light Transport
By Frédo Durand
- 103 **Geometric Tools for Exploring Manifolds of Light Transport Paths**
By Wenzel Jakob and Steve Marschner
- 112 **Technical Perspective**
Treating Networks Like Programs
By George Varghese
- 113 **Software Dataplane Verification**
By Mihai Dobrescu and Katerina Argyraki



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
Bobby Schnabel
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
Bernard Rous
Director, Office of Group Publishing
Scott E. Delman

ACM COUNCIL

President
Alexander L. Wolf
Vice-President
Vicki L. Hanson
Secretary/Treasurer
Erik Altman
Past President
Vinton G. Cerf
Chair, SGB Board
Patrick Madden
Co-Chairs, Publications Board
Jack Davidson and Joseph Konstan
Members-at-Large
Erik Allman; Ricardo Baeza-Yates;
Cherri Pancake; Radia Perlman;
Mary Lou Soffa; Eugene Spafford;
Per Stenström
SGB Council Representatives
Paul Beame; Barbara Boucher Owens

BOARD CHAIRS

Education Board
Mehran Sahami and Jane Chu Prey
Practitioners Board
George Neville-Neil

REGIONAL COUNCIL CHAIRS

ACM Europe Council
Fabrizio Gagliardi
ACM India Council
Srinivas Padmanabhuni
ACM China Council
Jiaguang Sun

PUBLICATIONS BOARD

Co-Chairs
Jack Davidson; Joseph Konstan
Board Members
Ronald F. Boisvert; Anne Condon; Nikil Dutt; Roch Guerrin; Carol Hutchins;
Yannis Ioannidis; Catherine McGeoch;
M. Tamer Ozsu; Mary Lou Soffa; Alex Wade

ACM U.S. Public Policy Office

Renee Dopplick, Director
1828 L Street, N.W., Suite 800
Washington, DC 20036 USA
T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association
Lissa Clayborn, Acting 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 GROUP PUBLISHING
Scott E. Delman
cacm-publisher@cacm.acm.org

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Andrew Rosenbloom
Senior Editor/News
Larry Fisher
Web Editor
David Roman
Rights and Permissions
Deborah Cotton

Art Director
Andrij Borys
Associate Art Director
Margaret Gray
Assistant Art Director
Mia Angelica Balaquiot
Designer
Iwona Usakiewicz
Production Manager
Lynn D'Addesio
Director of Media Sales
Jennifer Ruzicka
Publications Assistant
Juliet Chance

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

CONTACT POINTS

Copyright permission
permissions@cacm.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmh@cacm.acm.org
Letters to the Editor
letters@cacm.acm.org

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

AUTHOR GUIDELINES
<http://cacm.acm.org/>

ACM ADVERTISING DEPARTMENT

2 Penn Plaza, Suite 701, New York, NY
10121-0701
T (212) 626-0686
F (212) 869-0481

Director of Media Sales
Jennifer Ruzicka
jen.ruzicka@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
Moshe Y. Vardi
eic@cacm.acm.org

NEWS

Co-Chairs
William Pulleyblank and Marc Snir
Board Members
Mei Kobayashi; Kurt Mehlhorn;
Michael Mitzenmacher; Rajeev Rastogi

VIEWPOINTS

Co-Chairs
Tim Finin; Susanne E. Hambrusch;
John Leslie King
Board Members
William Aspray; Stefan Bechtold;
Michael L. Best; Judith Bishop;
Stuart I. Feldman; Peter Freeman;
Mark Guzdial; Rachelle Hollander;
Richard Ladner; Carl Landwehr;
Carlos Jose Pereira de Lucena;
Beng Chin Ooi; Loren Terveen;
Marshall Van Alstyne; Jeannette Wing

PRACTICE

Co-Chairs
Stephen Bourne
Board Members
Eric Allman; Terry Coatta; Stuart Feldman;
Benjamin Fried; Pat Hanrahan;
Tom Limoncelli; Kate Matsudaira;
Marshall Kirk McKusick; George Neville-Neil;
Theo Schlossnagle; Jim Waldo

The Practice section of the CACM Editorial Board also serves as the Editorial Board of *ACM Queue*.

CONTRIBUTED ARTICLES

Co-Chairs
Andrew Chien and James Larus
Board Members
William Aiello; Robert Austin; Elisa Bertino;
Gilles Brassard; Kim Bruce; Alan Bundy;
Peter Buneman; Peter Druschel;
Carlo Ghezzi; Carl Gutwin; Gal A. Kaminka;
James Larus; Igor Markov; Gail C. Murphy;
Bernhard Nebel; Lionel M. Ni; Kenton O'Hara;
Sriram Rajamani; Marie-Christine Rousset;
Avi Rubin; Krishan Sabnani;
Ron Shamir; Yoav Shoham; Larry Snyder;
Michael Vitale; Wolfgang Wahlster;
Hannes Werthner; Reinhard Wilhelm

RESEARCH HIGHLIGHTS

Co-Chairs
Azer Bestavros and Gregory Morrisett
Board Members
Martin Abadi; Amr El Abbadi; Sanjeev Arora;
Nina Balcan; Dan Boneh; Andrei Broder;
Doug Burger; Stuart K. Card; Jeff Chase;
Jon Crowcroft; Sandhya Dwaekadas;
Matt Dwyer; Alon Halevy; Norm Jouppi;
Andrew B. Kahng; Sven Koenig; Xavier Leroy;
Steve Marschner; Kobbi Nissim;
Steve Seitz; Guy Steele, Jr.; David Wagner;
Margaret H. Wright

WEB

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

ACM Copyright Notice

Copyright © 2015 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@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 \$100.

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 acmh@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 U.S.A.



Association for
Computing Machinery





Bobby Schnabel

DOI:10.1145/2833337

A Tale of ACM Visions

ACM MEMBERS WHO grew up in English-speaking countries are likely familiar with the first words of Charles Dickens' *A Tale of Two Cities*: "It was the best times, it was the worst of times ..."

A somewhat lame paraphrase of Dickens' epic words applies to the vision for ACM in the coming years: *It is a time of great opportunities, and a time of potential major challenges.* (Which immediately shows why I am a technologist, not a novelist.)

It is a great privilege to become ACM's CEO this month, and to be able to work with all of ACM's members on the future of the world's leading professional computing society. This column lays out some brief thoughts on that future, in the hopes of stimulating your reactions and suggestions.

First, on behalf of all ACM members, my deep thanks to John White who served ACM magnificently for many decades, including as president and for the last 17 years as CEO. John's vision and efforts played a major role in positioning ACM for the opportunities that come next.

Great Opportunities: Who Participates in ACM, and What ACM Does

A fully international society. In recent years, ACM has become a broadly international society, with the formation of ACM China and ACM India and the continued development of ACM Europe. What would a fully international society be? It would represent all areas of the world where computing is practiced—the entire world! And its programs, leadership, recognition, and reputation would fully reflect this international diversity and breadth. This is the ACM we are headed toward.

Greater inclusion of practitioners and entrepreneurs. ACM has over 110,000 members, about 60% of whom are computing practitioners. But there are estimated to be 15–20 million computing practitioners in the world. How broadly does ACM wish to reach these people, and what offerings are most effective in doing so? And how can the lifeblood of the computing industry—entrepreneurship and investment—be fully merged with the state-of-the-art technical expertise of ACM conferences and activities? There is a lot we can do.

Keeping up with the evolution and influence of computing. Modern computing consists of verticals and horizontals—the technical areas of computing expertise (for example, graphics), and the applications of computing (for example, autonomous vehicles or healthcare). ACM was configured years ago primarily for the technical areas, and computing now dominates many of the world's most significant applications. And both keep evolving. How we evolve with it and effectively address both dimensions is a great problem to have.

Computing education worldwide. Thanks in good part to ACM leadership and partnerships, we are seeing a surge in many parts of the world in attention to computing education at all levels, particularly pre-university. ACM and its affiliate CSTA will continue to have great opportunities to lead in developing the content, training the teachers, and emphasizing the importance of computing education.

Diversity in computing: The diversity of participants in computing education and jobs continues to be a challenge—women in most parts of the world, underrepresented minority populations in some. The world's leading computing society can and

should be a world leader in ensuring all groups are meaningfully represented in the great field of computing.

Potential Challenges: Membership, Publications

ACM is in wonderful financial shape but shifting sands impact two of its major revenue components:

Evolving membership models. The assumption that "I am a professional in X, therefore I join the X professional society" is undergoing a generational change. This reinforces the need to define and emphasize the benefits that come especially from ACM membership, such as special opportunities to network and connect. This is certain to be an area of attention and innovation.

Publication in the era of open access. The ACM Digital Library is a cornerstone of the society and a tremendous service to computing. In the area of open access publications, it also raises the question: Does the library provide more advantages than the collection of individual publications, which increasingly can be accessed from their authors? The answer can be and needs to be a resounding "yes"—whether through abilities to search and connect, access related materials, and more. Our entire community will help define what is helpful and how we provide it.

My colleagues tell me I am very responsive to email. Granted, 110,000+ members certainly can test and break that reputation, but I would be delighted to hear your thoughts on these issues, and promise to read them and hopefully to respond! 

Bobby Schnabel (schnabel@acm.org) assumed the position of ACM's Executive Director and Chief Executive Officer on November 1, 2015.

Copyright held by author.

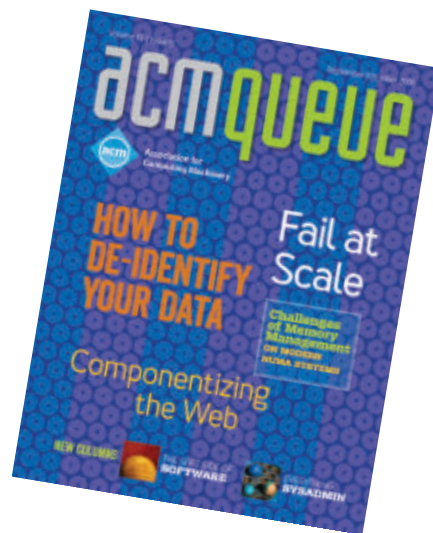
acmqueue

Check out the new acmqueue app

FREE TO ACM MEMBERS

acmqueue is ACM's magazine by and for practitioners, bridging the gap between academics and practitioners of computer science. After more than a decade of providing unique perspectives on how current and emerging technologies are being applied in the field, the new acmqueue has evolved into an interactive, socially networked, electronic magazine.

Broaden your knowledge with technical articles focusing on today's problems affecting CS in practice, video interviews, roundtables, case studies, and lively columns.



Keep up with this fast-paced world on the go. Download the mobile app.



Association for
Computing Machinery

Desktop digital edition also available at queue.acm.org.
Bimonthly issues free to ACM Professional Members.
Annual subscription \$19.99 for nonmembers.



Vinton G. Cerf

DOI:10.1145/2831359

Memory with a Twist

I continue to look for ways to preserve digital information over long periods of time. Often, it is not the medium that fails but, rather, the necessary reading gear. For something

considered extremely important, it is not unusual to find a way to read only media. Think about laser scanning of long-playing records, for example. But for most users, such extreme measures may be infeasible for reasons of cost. Of course, even if the binary information can be recovered, there is still the problem of correctly decoding/interpreting the information. This is typically the job of the software that created it in the first place. I have written before about the OLIVE project at CMU that supports the execution of old operating systems and applications on computer hardware that is emulated through virtual machines.^a

In this column, however, I want to explore an idea I recently learned about in a 2013 article in *Nature* magazine.^b I spoke with two of the authors recently. The basic idea is to encode binary data in base 3, Huffman code the result, and then synthesize DNA using the standard base-pair triplets in a ternary encoding scheme. Care is taken not to use the same base pair in sequence to avoid homopolymers that might confuse the decoding stage. Additional “bits” can be used for indexing and error correction. Moreover, each fragment of about 100 bases is made to overlap with previously encoded fragments by about

75 base pairs, creating a fourfold redundancy in the encoded strings. On top of this, one can use additional base pairs, not to encode bits but to allow the sequencing equipment to select subsets of the encoded strings to avoid having to decode everything to get at only the content desired. One might think of that as a sort of biochemical file folder mechanism.

In today's world, the cost of synthesizing DNA is non-trivial but as we have seen with computing and other high technologies, costs come down and what was once laboratory-only equipment has become, literally, a household item. Setting this important cost question aside for a moment, I learned that DNA, once dehydrated, is remarkably stable. It could survive in dry condition for tens of thousands or even hundreds of thousands of years. Successful resequencing of prehistoric DNA illustrates this feature. One could imagine storing petabytes of information in vials containing dried DNA and using robots like those in today's tape libraries to retrieve an appropriate vial whose contents are to be rehydrated and sequenced. The authors report they achieved a storage density of 2.2PB/gram.

I was reminded of the IBM 1360 Photo-Digital Storage System^c that used capsules of stiff microfilm to store information. I actually had a chance to watch one of the machines that had been installed at Lawrence Livermore National Laboratory. The

electromechanical design of the system was astonishing and I found myself mesmerized as capsules of film moved from their storage locations to readers that mechanically popped open the capsules, extracted the film, read it, replaced it into the capsule, and returned the capsule to a storage bin. There could be up to 13 capsules “in flight” at one time in the system using a pneumatic tube system not unlike the way banking and payment systems once worked in department stores. A programmable controller kept track of the capsules.

While on the subject of the stability of DNA, it has occurred to me (and, of course, many others) that the theory behind *panspermia*^d is made somewhat more credible by the observation that dehydrated DNA is so stable that passage through the vacuum of space would likely not harm it. Over long periods, though, one might find the DNA is altered at the molecular level by radiation. Since travel over interstellar distances could take hundreds of thousands to millions of years, there appears to be reason to be somewhat skeptical about successful propagation of life in this fashion. For the same reason, slow interstellar propagation of information might be similarly improbable. □

d <http://www.panspermia-theory.com/>

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

Copyright held by author.

a <https://olivearchive.org/>

b N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E.M. LeProust, B. Sipos and E. Birney. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. Letter, *Nature* 494, (Feb. 7, 2013), 77 ff; doi: 10.1038/nature11875.

c https://en.wikipedia.org/wiki/IBM_1360

Who Will Read PACM?

NONE OF THE three articles on the proposed *Proceedings of the ACM*—proposal, pro, and con—as coordinated by Joseph A. Konstan and Jack W. Davidson and introduced through their column “Should Conferences Meet Journals and Where? A Proposal for ‘PACM’” (Sept. 2015) addressed what should be the fundamental concern for any publication: Who will actually read (and subscribe to) it? Would PACM, independent of the merits of the proposal for authors, have any value for ACM if it has no readers? I have some sympathy for academic researchers who produce high-quality conference papers yet do not get full academic credit for their work. But that is a problem with “academic scoring,” and producing a publication whose sole value proposition is fixing that problem receives no sympathy from me.

The challenge for ACM should be how to deliver quality research to the more general ACM membership, not just the research community but practitioners who can actually apply the results of the research. Not every paper needs to have an immediate practical application, but I have read many conference papers that deserved exposure to a wider audience, particularly for papers I read 25 years ago that addressed problems only now becoming widely understood within industry. *Communications* has made substantial progress in this direction over the past few years, and *ACM Queue* should be considered a success. ACM must understand the era of making money with paper is over, and adding a publication whose only subscribers are potential authors and libraries with the money and space for “archival publications” represents a step in the wrong direction.

David Emery, Reston, VA

Authors’ Response:

Emery raises the question of how ACM can publish more content for a broad

audience. We reject the trade-off between research and practice content as a false one. The PACM proposal does not create new content; it is an effort to recognize and systematically improve the quality of ACM’s best conference papers. These papers are already well read and well cited. With PACM, they will continue to be easily accessible in the ACM Digital Library.

ACM recognizes the importance of reaching practicing professionals. We continue to invest in Queue, are working to enhance practitioner content in Communications, and welcome suggestions for new publications for practicing professionals.

Joseph A. Konstan
and Jack W. Davidson,
ACM Publications Board Chairs

Who Pays When Autonomous Fails?

Gregory Mone’s news story “The New Smart Cities” (July 2015) was both alarming and enlightening, on smart things being for dumb people, while Keith Kirkpatrick’s news story “The Moral Challenges of Driverless Cars” (Aug. 2015) pondered what it means for paid scientists to dance before the golden idol of autonomous vehicles.

Consider, as quoted by Kirkpatrick, Patrick Lin et al. carefully sidestepped the core question of liability. Who would be held responsible if using an autonomous vehicle would result in damage or even death? The user, the vehicle manufacturer, or the involved software companies? It was convenient for Lin et al. to cite “legal and ethical challenges” instead of calling a stick a stick.

But wait, Jonathan Handel, the computer scientist turned lawyer mentioned by Kirkpatrick came to the rescue by proposing “ethics review boards,” so we may yet have as a “procedural solution” certification by a board. Software companies would thus be off the hook by certifying their own products, and vehicle manufacturers would be off the hook by using certified software. Who would be left to actually take responsibility?

This might sound like legalizing manslaughter by different means. But review boards would be a tremendous help for companies like Google, Volkswagen, and Mercedes “pressing ahead for a fully autonomous vehicle,” per Kirkpatrick, because there is money to be made in autonomous vehicles. Concerning cars, what avid driver would be interested in an autonomous vehicle that hints of mass transportation? And what else are autonomous vehicles but an effort to combine the efficiency of mass transportation with the luxury of personal transport? As for trucks, autonomous trucks are a missing link in a fully integrated logistics chain, and research in and infrastructure for autonomous vehicles would benefit them enormously.

Ethical shmetical ... Given the economic implications (and scientists blind eye) this line of research probably is, I am sorry to say, “without alternatives.”

Hans Grünberger, Wiesbaden, Germany

Give Me Code Access

Dorothy E. Denning’s Viewpoint “Toward More Secure Software” (Apr. 2015) mentioned “... very few users are in a position to inspect source code,” making me wonder if she based this view on a “user survey” in an open source community like Linux groups or intended to communicate something else. I wonder because, as Microsoft critics often say, the company’s product-access policies lead to users being unable to look into their software’s (hidden) behaviors. On the other side, Linux proponents and other open source advocates usually claim users and developers alike can search for bugs and (unwanted) program routines in all kinds of Linux software. As a devoted Linux practitioner, I can say I have looked into the sources just a few times, usually during the compiling procedures of program executables; I am simply too preoccupied with other tasks to do that kind of inspecting more often.

It looks to me the question of access is really about whether the end-user and developer communities are able to inspect the code, in light of their willingness, free time, and/or professional qualifications.

Miroslav Škorić, Novi Sad, Serbia

Author's Response:

I was referring to the general user population, not communities of software developers. Most people have never written a program. But I cannot even imagine myself inspecting the tens of millions of lines of source code that generated the OS and applications running on my computer, despite my background in computer science and information security.

Dorothy E. Denning, Monterey, CA

Finite State Machines for UI Design Up Front

Arie van Deursen's article "Testing Web Applications with State Objects" (Aug. 2105) elaborated on the utility of representing user interface (UI) elements by state machines or state objects for creating test scenarios. The finite state machine (FSM) is an excellent model for earlier development steps like requirement, design, and implementation. Behavior and process decisions, as described by van Deursen, should be taken much earlier in specifying the developed element. Developers and test specialists alike should then use the same model for implementing the application and its test scenarios.

The advantages of using finite state machines for development should be appreciated. The same FSM transitions set (state-transitions table or diagram) tends to be used throughout the process, from requirements to test, streamlining development. Content of distinct states creates a "layered FSM,"^{1,2} with requirements, design, implementation, and test layers for each state. Another advantage is the extensive set of structure and semantic rules for evaluating FSMs for correctness, in particular the event-state analysis introduced in Avnur² that can also be used to construct an FSM with fewer defects in the first place. Using them, developers can bring FSM-based soft-

ware to a very high level of quality even before the test phase. Additionally, FSM implementation can be automated through a code generator (such as the State Machine Compiler, <http://smc.sourceforge.net/>) or through a generic FSM class, as in Avnur,² implementation. Reducing the scope of manual coding reduces defect rates even further.

UI design using state machines is not common, but for no good reason; all the general benefits described here apply to UI software as well; tools like state objects and articles like van Deursen's will hopefully encourage it.

A development process that uses tools like FSM models to facilitate high-quality forward development even before testing is more correct by design and thus more effective than relying on test-fix cycles.

References

1. Avnur, A. A finite state machine model for requirements engineering. *IREB Requirements Engineering Magazine* (2015-03); <http://re-magazine.ireb.org/issues/2015-3-thinking-without-limits/a-finite-state-machine-model/>
2. Avnur, A. *The Key FSM*; <http://www.aa-sw-dev.com/KeyFSM/KeyFSM.pdf>

Arie Avnur, Chestnut Hill, MA

Ada, Not ADA

Thomas Haigh's and Mark Priestley's Viewpoint "Innovators Assemble: Ada Lovelace, Walter Isaacson, and the Superheroines of Computing" (Sept. 2015) mentioned "... when the language ADA was named in her honor ..." The title of the ANSI/MIL-STD-1815A is "Ada Programming Language." It, and everything since on the subject, uses the capitalization of a person's name, not an acronym.

Tom Moran, Saratoga, CA

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

© 2015 ACM 0001-0782/15/11 \$15.00

Coming Next Month in COMMUNICATIONS

Personalized Maps

**Smart Data Pricing:
Using Economics
to Manage Network
Congestion**

Propositions as Types

**Internet Use
and Psychological
Well-being**

NoDB

**What Makes Paris
Look Like Paris?**

**How to De-Identify
Your Data**

**Challenges of Memory
Management on
Modern NUMA
Systems**

**Lean Software
Development:
Building and Shipping
Two Versions**

**Creating a New
Generation of
Computational
Thinkers**

Plus the latest news about the reproducibility of code, smart technologies for treating injuries, and the evolution of online education.

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



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

DOI:10.1145/2820419

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

How Men Can Help Women in CS; Winning 'Computing's Nobel Prize'

Valerie Barr suggests how men can better promote women in computing, while Michael Stonebraker shares how it feels to receive the top honor in the computer science field.



Valerie Barr
"Things That Men Can Do To Be Real Allies For Women In Computing"

<http://bit.ly/1lffBfE>

August 3, 2015

With thanks to the authors who contributed to "11 things white people can do to be real anti-racist allies" (<http://bit.ly/1IKk2Bf>).

1. Talk to other men in computing in order to create more allies. Talk to other men who are not in computing. Talk to women who are not in computing. Women in computing need lots of allies.

2. As supportive as you are, it is not enough to sit back and watch from the sidelines. Help build a movement to make computing a fair and equitable place for everyone.

3. Because you are a man, you get a seat at lots of tables. Do not be embar-

rassed about that, do not try to pretend it is not the case. Use that to help change who else gets a seat at the table with you.

4. Think about how you use language, both in speech and all forms of writing (both formal and informal writing). Are you using terminology and phrasing that reinforces stereotypes about who belongs in computing?

5. Read what women in computing are writing about their experiences. Listen to their TED talks and conference presentations. Just as importantly, read about the technical work being done by women in computing and refer to it in your own work.

6. Every time a man you know in tech announces he is about to become a father, ask him when he is taking paternity leave, and for how long, and whether he will be the primary parent at home during that time, and whether he will shut down his work email while he is on leave.

7. Be prepared with a response when you hear someone say the company

should not hire women because they will leave to have babies (yes, it may be the 21st century, but people still think and say that). There are lots of ways to counter that. Be creative. And see #6, it might give you inspiration.

8. Speaking of kids—visit toy stores. Study how toys are packaged and marketed. Begin to understand the gendering of the toy aisle. Buy cool puzzles and building games for all the children you know, ones that are designed to encourage kids to use their imagination. Plan in advance and shop online in the "education" store of companies instead of buying the pre-packaged imagination-dulling products that the same companies sell in stores.

9. Talk to girls about math and science and computing. Ask them what is going on in school. Do not sit by idly when you hear the teacher is not calling on the girls or is letting the boys dominate the techie activities.

10. Find out what is going on in your area, and then get your company to support those activities financially *and* by letting staff take time to contribute to the activities. Is there a Girls Who Code or a Black Girls Code group? Is there an ACM-W Celebration of Women in Computing being planned? Is there an ACM-W student chapter at a nearby school? Is there a Girls Inc. or Girl Scouts tech program? Is there an all-girls FIRST Robotics team? An investment of your time will help the girls in the program (short-term impact) and help change the culture in your workplace (longer-term impact).

11. Finally, do not expect a lot of pats on the back and thank you calls

and messages. There will not be an overwhelming wave of acclaim because you finally decided to act on the belief that women belong in tech. We will be happy to have larger numbers of people making the case, but we expect everyone to wage this fight. You will be welcome, but you will not be exceptional. Thanks in advance.



Michael Stonebraker
**"What It's Like to Win
 the Turing Award"**

<http://bit.ly/1MrKPcK>
 July 24, 2015

For about the last 15 years, I have had a fantasy concerning the paper I would write if I actually won the Turing Award. After all, it is a once-in-a-lifetime opportunity to write a paper that is not subject to the whims of external reviewers. My fantasy was to write something personal and "outside the box," and it had several components. The first was to try to explain why system software is so hard to build, and why good teams screw it up on a regular basis. Second, it takes real perseverance to "stick it out" and make something actually work. The third was to talk about the start-up experience, and why venture capitalists usually deserve their reputation as "land sharks." Lastly, it is clear that luck plays a significant role in successful startups, and I wanted to explain that. The overarching theme was to use a significant physical challenge as a metaphor for system software development. Over the years, the physical challenge has varied between our cross-country bike ride in 1988, and my climbing all 48 4,000-foot mountains in New Hampshire.

So when Barbara Liskov asked to see me on March 6th, I was immediately apprehensive. It was unlikely to be a technical meeting, since I was not collaborating with her on any project at the moment. I also knew she was the head of the Turing committee, so I was expecting either good news (I won) or bad news (somebody else in system software had won and I would not realistically be a candidate for a few years). When she said it was the former, I teared up. The recognition and validation for my lifetime work was incredibly gratifying. On a

"I am in a bit of a quandary about 'what next'. After all, there is no possible recognition to strive for ... It is like being at the top of Mt. Everest; there is no 'up' in sight."

more basic level, I actually got to realize my fantasy!

She said I could share the news with my wife, but nobody else until the press release came out some two weeks hence. Also, she mentioned I would be asked to sign a letter committing to write a Turing paper in a timely fashion, give the Turing Lecture at FCRC on June 14th, attend the awards ceremony on June 20th, and make speaking appearances internationally. She also said there would be a ton of other speaking invitations. It was very sweet that she gave me the news in person, rather than via a phone call out of the blue. My wife and I had a private celebration that night.

Over the next two weeks, the hardest part was remembering who knew and who did not, and acting accordingly. Obviously Barbara knew, as well as the faculty who nominated me; however, I had a couple of conversations of the form, "By the way, I know about it, and there is something we want to plan to do ..."

On the morning of the announcement, I got about 400 email messages, mostly from system software researchers and former students saying how delighted they were. It was a very touching show of support. It was especially touching to talk to my friend and colleague Dave Dewitt. He said it was truly ironic that the two events he had been eagerly awaiting (his daughter Lizzy getting married and me winning the Turing Award) were both going to happen on the same day 2,000 miles apart, and he could only attend one. I am sure you can guess which event won.

Now, I had to actually write the fantasy paper I mentioned. It turned out

to be very uncomfortable. I am accustomed to writing and presenting dry technical material; however, my fantasy was a real emotional stretch for me. Presenting it was even harder. I tried it out on my wife, and then on Dave Dewitt and a few other senior DBMS researchers. I then presented it to the MIT DBMS community, and to audiences at two of the companies I started. Trying out the talk on these friendly audiences gave me the confidence to actually go through with the plan.

It was exceedingly draining to give the talk at FCRC, since there were no DBMS people there to offer moral support. Also, the room was completely dark, so I could not judge how it was being received. I still do not have a good idea whether the audience liked it or not.

The next week was the actual awards ceremony in San Francisco. About 30 friends and colleagues showed up, and it was a very touching event. Again, I felt like it validated the relevance of database management to ACM. I hauled my tuxedo out of mothballs for the event (I hope it can go back into mothballs for the rest of my life).

I am in a bit of a quandary about "what next." After all, there is no possible recognition to strive for off into the future. It is like being at the top of Mt. Everest; there is no "up" in sight. Some thoughts that occur to me: try to help/mentor assistant professors. After all, my experience (many moons ago) as an assistant professor was miserable, and stressful beyond belief. Helping out budding entrepreneurs is also on my radar. I am also concerned that U.S. enterprises are not very good at information technology. It is widely reported that two-thirds of all IT projects fail. Ultimately, successful companies will have to get good at IT, since that is obviously going to be a major differentiator off into the future. But how to make a difference?

I have a year ahead of me with quite a bit of travel, where I will get to talk to lots of people to try to figure this stuff out. □

Valerie Barr is a professor at Union College. **Michael Stonebraker** is an adjunct professor in the Massachusetts Institute of Technology's Computer Science and Artificial Intelligence Laboratory, and recipient of the 2014 ACM A.M. Turing Award.

© 2015 ACM 0001-0782/15/11 \$15.00

Algorithmic Authors

Natural Language Generation software begins delivering on the promise of automated prose.

SOON AFTER THE Sonoma County Little League's Falcons faced off against their rivals the Mustangs on April 28, 2015, the league's website ran this game recap:

Anthony T got it done on the bump on the way to a win. He allowed two runs over 2-1/3 innings. He struck out four, walked two, and surrendered no hits.

Anders Mathison ended up on the wrong side of the pitching decision, charged with the loss. He lasted just two innings, walked two, struck out one, and allowed four runs.

That story—and millions more like it—was written not by a cub reporter, but by Quill, a software program developed by Chicago-based Narrative Science. By analyzing game statistics entered using a mobile app and uploaded via the Internet, Quill identifies patterns in the data to produce original stories intended to mimic the voice of a flesh-and-blood scribe.

Along with companies like ARRIA, Yseop, and Automated Insights, Narrative Science is helping to launch a new market for editorial content created using the principles of Natural Language Generation (NLG).

In contrast to the better-known field of Natural Language Processing (NLP)—in which software transforms raw text into structured datasets—NLG

addresses the opposite problem: turning structured data into prose.

There is nothing terribly new about NLG per se; the foundational research goes back decades, and it has long been a staple of Artificial Intelligence

programs. The field has enjoyed a resurgence in recent years, thanks largely to the rise of big data.

“With all the investment in big data, the world has a tremendous amount to tell us,” says Kris Hammond, professor



of computer science at Northwestern University and chief scientist for Narrative Science. “The need for this kind of technology has exploded.”

Hammond began exploring the possibilities of computer-generated journalism along with his research partner Larry Birnbaum at Northwestern University, where he supervised a student research project called StatsMonkey, an experiment in auto-generating college sports stories in collaboration with the school’s journalism department.

Since co-founding Narrative Science with former Doubleclick executive Stuart Frankel in 2010, Hammond has seen the company grow beyond its roots in the sports world to serve not just beleaguered Little League coaches, but professional purveyors of prose like Forbes.com, and several Fortune 500 firms who use Quill to produce management reports from internal business data.

Automated Insights has produced more than a billion stories to date ranging from quarterly earnings reports for the Associated Press to car descriptions for Edmunds.com, digital marketing material for Comcast, and game recaps for Yahoo! Fantasy Football.

With its reliance on structured data like scores, player stats, and win/loss records, sports journalism is a natural fit for NLG applications. Similarly, financial and other business data lends itself well to automated report writing. The potential applications extend into any number of domains where human beings are trying to make sense of data.

Both Quill and Automated Insights’ Wordsmith software perform similar functions: gathering data from third-party sources, performing data analysis to glean a set of insights, and then applying NLG techniques to translate those insights into readable stories.

Wordsmith can ingest data in a wide range of formats (like XML, CSV, spreadsheets, and various APIs), then analyze the material to identify trends, changes, and other pertinent patterns. The software can then produce stories in any number of narrative formats, from traditional inverted-pyramid-style news stories to bullet-point summaries, tweets, and headlines. The software exports the resulting content in any number of formats, including

For Hammond, the deeper challenge lies not in generating copy, but in finding the most pertinent meaning in a given dataset. “It’s not just about converting numbers to language.”

XML, JSON, HTML, Twitter, or email.

While recapping baseball games or churning out company earnings reports might seem like reasonably straightforward operations, the vagaries of human language and cognition can pose thorny computational challenges in creating credibly human-sounding stories.

At minimum, machine-generated content must meet basic standards of journalistic accuracy. “NLG should always strive to be well written, but the text must always support the data,” says Automated Insights’ James Kotecki. “In some cases, changing one word, adding an adjective, or even switching the placement of words can create unintended factual errors.”

As any professional editor will tell you, a good story has to do more than just get the facts straight; it also has to engage the reader’s attention. This is where the real AI challenges come into play.

“Truly human-sounding content must adhere to the contextual and tonal expectations of its intended audience,” says Kotecki. To bring a story to life, the software must imbue the data with one of the essential virtues of human language: variation. This means not just mixing up synonyms, but determining the appropriate tone and tenor of the story, given the patterns in the data.

“If I’m telling a story about a game that was a blow-out, where one team took the lead and held on and pounded the other team into the ground, that’s

Milestones

Computer Science Awards

CRA HONORS JAHANIAN, GATES

The Computing Research Association (CRA) has awarded Farnam Jahanian, vice president for research at Carnegie Mellon University, its 2015 CRA Distinguished Service Award.

Farnam was U.S. National Science Foundation assistant director for Computer & Information Science & Engineering from 2011 to 2014. He also was co-chair of the Networking and Information Technology Research and Development subcommittee of the National Science and Technology Council Committee on Technology.

CRA also presented Ann Quiroz Gates, chair of the department of computer science at the University of Texas at El Paso, with the 2015 A. Nico Habermann Award, for “outstanding contributions aimed at increasing the numbers and/or successes of underrepresented groups in the computing research community.” Long a leader in initiatives supporting Hispanics and other underrepresented groups in the computing field, Gates led the Computing Alliance of Hispanic-Serving Institutions, and helped establish the Affinity Research Group model for research mentoring and peer support.

PAPADIMITRIOU RECEIVES 2015 EATCS AWARD

The European Association for Theoretical Computer Science recently awarded Christos Papadimitriou, C. Lester Hogan Professor of Electrical Engineering and Computer Science in the Computer Science division of the University of California at Berkeley, the EATCS Award 2015.

The organization said Papadimitriou’s body of work has had a profound influence on many areas of computer science, contributing “truly seminal work” to fields including algorithmics, complexity theory, computational game theory, database theory, Internet and sensor nets, optimization, and robotics.

a different story than if it was a squeaker,” says Hammond.

For Hammond, the deeper challenge lies not in generating copy, but in finding the most pertinent meaning in a given dataset. “It’s not just about converting numbers to language,” he says. “Those numbers need context.”

That context comes not from data analysis alone, but from understanding the larger domain and, in some cases, historical patterns. The software applies core AI principles like inference and reasoning to analyze the data in context, and to deduce the most relevant and interesting set of potential meanings. “It goes beyond simple NLG,” says Hammond. “How do you introduce ideas, expand on them, and explain them?”

Quill uses its own proprietary case-frame language, which Hammond characterizes as an “old-school AI” strategy of capturing the entities and languages within a given domain and encoding them in an ontology.

For example, the entities in a baseball game might include pitchers, hitters, balls, strikes, and runs; the system understands the semantic relationships between those entities, and can produce a set of variable terms to describe all the possible interactions between them. “For every relationship/object/predicate, you need variability built into the system,” says Hammond.

Can this programmatic combination of data analysis and AI techniques really produce credible journalism? Will automated stories ever pass the journalistic equivalent of a Turing test?

In 2014, Swedish researcher Christer Clerwall published a study entitled “Enter the Robot Journalist” that gauged reader responses to a news article produced by Automated Insights’s Wordsmith program with a story on the same topic penned by a flesh-and-blood scribe from the *Los Angeles Times*.

Participants in the study found the software-generated story to be more informative and trustworthy than the one written by the journalist, but they also characterized it as “descriptive, boring, and objective.”

In some cases—like quarterly earnings reports—boringness might ac-

tually seem like a virtue, but in other domains, this lack of narrative sparkle may preclude machine-generated stories from making it to the front page anytime soon.

For now, these companies are focused primarily on producing the kinds of content that are unlikely to displace any workaday writers. Hammond likes to talk about producing stories that serve “an audience of one.” There are not too many full-time reporters out there working the Little League beat, after all.

In a similar vein, Automated Insights is developing plans to make its software more accessible to the general public, so individual consumers can upload their own data and generate automated stories on topics of their choosing.

Future applications for NLG may extend well beyond the realm of the traditional long-form narrative. At Aalto University in Finland, Ph.D. student Eric Malmi has created a software program called DopeLearning that is capable of generating its own rap lyrics. Other promising NLG projects include companies like x.ai, which offers an artificial intelligence-driven calendaring service that provides users with a virtual assistant named “Amy Ingram,” who can arrange meetings for a user by sending email in a computer-generated writing style that is difficult to distinguish from that of a human assistant.

As NLG applications continue to evolve, developers may set their sights on more complex topics than sports and corporate earnings reports. Hammond and others have high hopes for the potential of NLG techniques to turn out high-quality, original journalism.

Moving beyond these kinds of just-the-facts news stories will require another conceptual leap in NLG application development, taking computer-generated writing beyond the realm of interpreting structured data sources and into the far messier world of interpreting free-form written material, much of it currently generated by human journalists. To bridge that gap, NLG software may need to start incorporating some of the text-parsing methods of its kindred discipline Natural Language Processing (NLP).

Already, Narrative Science is start-

ing to incorporate data drawn from other news stories’ headlines. For example, the software can analyze another news outlet’s headline about an executive hiring or firing, then relate that data to subsequent movements in the stock price. However, scanning for keywords in a headline is a far cry from understanding the complexity and nuance of more interpretive journalism.

Generating information from unstructured text sources also poses much thornier problems of fact-checking, since it is one thing to double-check a stock price, but quite another to assess the reliability and motivations of a human subject. Seasoned editors and reporters rely on their years of journalistic experience to help vet their stories and screen out the implausible, which may prove difficult to replicate algorithmically.

While there is no shortage of computational challenges on the horizon, the sheer demand for written material across the Internet seems to point toward continued innovation in this nascent field. Could there be a robot Hemingway on the horizon? Unlikely as that may seem, Hammond has already gone on record as predicting a computer program will one day win the Pulitzer Prize.

Whether or not that ever happens, Hammond is convinced NLG applications can serve a basic human need: to help us make sense of the world around us not as a set of data, but as a web of deeply interconnected narratives. “We’re all wired for storytelling.” ■

Further Reading

Christer Clerwall.

Enter the Robot Journalist. *Journalism Practice*. 8, 5 (2014), 519-531. <http://bit.ly/1HDkz8C>

Reiter, E., and Dale, R., 1997.

Building applied natural language generation systems. *Nat. Lang. Eng.* 3, 1 (March 1997), 57-87. <http://bit.ly/1CGSsJ8>

Malmi, E., Takala, P., Toivonen, H.,

Raiko, T., and Gionis, A.,

DopeLearning: A Computational Approach to Rap Lyrics Generation (May 18, 2015). <http://bit.ly/1CGSvVn>

Alex Wright is a writer and researcher based in Brooklyn, NY.

© 2015 ACM 0001-0782/15/11 \$15.00

Companies Proactively Seek Out Internal Threats

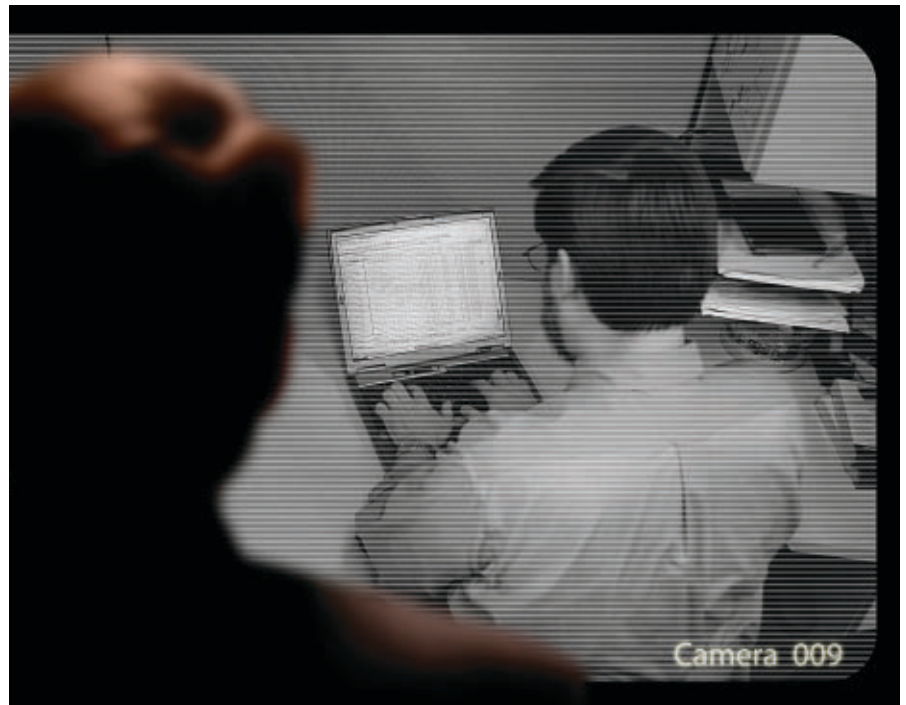
Organizations must balance their concerns with the protection of employee privacy.

ALTHOUGH HE IS not certain, Kevin Stecko, founder of 80sTees.com, believes a former employee hacked into his e-commerce system in 2013. With the benefit of hindsight, Stecko's takeaway is, "understand that no matter how small you think you are, there's a good chance you are being targeted by hackers on a regular basis. They are probing for any weakness in your systems, practices, or people."

While organizations increasingly are the victims of external hackers, many are finally realizing an uncomfortable truth: insider threats are and have long been a very real problem. Consequently, even as they are taking measures to protect their networks from external threats, there is a growing trend for companies to scrutinize their own employees more closely and take proactive measures to detect internal breaches.

A whopping 93% of respondent U.S. organizations believe they are vulnerable to insider threats, and plan to increase or maintain what they spend on information technology (IT) security and data protection, according to Vormetric's 2015 Insider Threat Report. Based on interviews by Harris Poll of "more than 800 senior business managers and IT professionals in major global markets, roughly half from the U.S. and the rest from the U.K., Germany, Japan, and ASEAN countries," 55% of respondents said their "privileged users" pose the biggest internal threat to their corporate data, followed by contractors and service providers (46%).

"Now you're seeing corporate America, as well as government, coming to grips with the need to be more compre-



hensive and proactive with approaching inside threats" by identifying individuals who present a "threat posture" and could potentially violate an organization's policies, says Scott Weber, managing director of Stroz Friedberg, a digital forensics consulting firm.

Insiders are much better positioned to gain access to corporate systems, Weber explains. "There's a trust factor; they know where the interesting information is, the crown jewels." At the same time, Weber adds, companies recognize the need to balance their concerns about internal threats with the protection of employee privacy. "The desire to obtain balance is part of every conversation we have," he says. "Employers want to trust employees and employees want to trust employers," and no one wants their

ability to do their job and support themselves to be negatively impacted by someone inside the organization who has access to data.

With greater attention being focused on insider threats thanks to leaks of classified documents by Edward Snowden and Bradley (Chelsea) Manning, companies are seeking out technologies that use algorithms to develop employee profiles of what is 'normal' behavior versus actions that should send up a red flag. This puts them on the offense, rather than in a "mitigation posture," Weber says.

One of those red flags: an employee who normally accesses and transfers 100 files a week, who suddenly one day is transferring 1,000 files, says Mohan Koo, CEO of security company Dtex Systems. "If they actually rename those

files before sending them out...or if they delete files after the transfer, that would trigger a higher level of warning," he explains. "It indicates they're trying to cover their tracks."

Other indicators of suspicious behavior may be the time of day (or night) an employee is accessing files, how often, and how long they look at them, Koo says. "All of sudden we might see a person using [files] after hours and 10 times during the night."

Dtex's technology is used by financial services firms to predict when a person is getting ready to leave an organization and steal intellectual property, he says. The software is deployed to all computers and devices an employee uses to access the corporate network and operates in the background, Koo says. "It's logging metadata so it's non-intrusive." The software does not scan files for content or look at what people are saying on social media or email, but rather scans apps they run, files they access, and what they do with them, as well as tracking and tracing the movement of files throughout the organization. Employees are aware the software is being used, he says.

One of the companies Dtex works with is a large financial exchange that deals with very sensitive market data. "We've been able to identify people getting ready to leave the organization [who were] searching for files they've never searched for before," says Koo. "We're looking for behaviors that are different from normal behaviors. Erratic behavior generally indicates a risk for the business."

Normal or baseline behavior is based on 90 days' worth of anonymous data that is studied to see which files and apps an employee typically uses. Dtex will then study the behavior of people who have resigned from an organization to see how their behavior changed in the weeks leading up to their leaving by applying behavioral analytics, Koo says.

"They will log on later and later in the morning and earlier in the evening, and the amount of work they do on their laptop [decreases] and becomes more personal," he notes. "We've successfully caught people trying to walk out with data before they leave."

Stroz Friedberg uses proprietary technology called SCOUT, which uti-

"We're looking for behaviors that are different from normal behaviors. Erratic behavior generally indicates a risk for the business."

lizes psycholinguistic algorithms to analyze electronic communications to assess things like word choice and frequency of use of particular words and phrases, in order to identify and measure risk indicators, Weber says. "Psycholinguistics is the science that involves the psychology of language," he explains. The algorithms are used to analyze language using different methods, and then that language is scored statistically in an automated way.

For one client, Stroz Friedberg analyzed over 28 million messages from about 57,000 senders, Weber says. Once filters were applied, the messages were whittled down to about 1,700 email messages deemed "statically significant," but even then, analysts will not read every one, only those identified based on "scientific discretion. That is frankly more respectful of privacy," he says. "You would never use one source to make a decision on how to proceed with an employee; you're always going to use a layered approach" such as psycholinguistic language and technical indicators such as suspicious behaviors. "The key is to figure out what data is important and what data provides actionable intelligence and correlate that with as many indicators as possible before taking action."

Insider threats are not necessarily increasing but are more frequently publicized, perhaps due to growing awareness of cyberattacks in general, observes Tony Kolish, senior vice president of customer services at cybersecurity firm FireEye.

Koo agrees. He has seen outside organizations influencing people inside competitors to come work for them

and bring proprietary data with them in the telecom, finance, pharmaceutical, and automotive industries, "so it really cuts across sectors."

IT employees are a particular target "because they have privileges around or know how to navigate systems, and they know what's being tracked and what isn't and how to get it out without the organization knowing about it," Koo says. "That's where visibility and knowing what people are doing becomes very, very critical."

Typically, FireEye is contacted after an attack has already occurred "and it's news to them, almost always," says Kolish. "When we go in and do a compromise assessment, [a hacker has] been in there on average longer than 200 days." While that is an alarmingly long period of time, Kolish says two years ago, the average was 267 days.

Like Dtex, FireEye's technology focuses on detecting behavioral anomalies, like employees accessing data they normally would not access, and doing so at odd times of day. Kolish sees data being stolen in the biotech sector "where there's been a fair amount of money to be made by stealing intellectual property related to drug trials."

What Companies Can/Should Do

FireEye emphasizes to customers that they cannot prevent all breaches. "There is no wall you're going to put up or depth of defense architecture ... that will prevent all breaches," says Kolish. What companies should instead be focusing on is detecting and containing security breaches as quickly as possible. "We're in defensive mode here, all of us, and attackers are always going to find something new. Take reasonable steps to prevent them."

He advises clients to make certain they are encrypting data and using two-factor authentication. Additionally, he says, companies should "identify the data you care about the most and make sure your eyes and ears are tuned to access of that data."

Lockheed Martin has gone a step further; last year, the aerospace manufacturer and military contractor developed an insider threat detection program to proactively detect and mitigate theft or misuse of intellectual property and trade secrets. The impetus was the growing trend of data theft,

as well as economic and industrial espionage affecting corporate America, says Douglas Thomas, director of Counterintelligence Operations and Corporate Investigations.

The key to developing a successful insider threat detection program, say Thomas and Weber, is including team members from a cross-section of business units. Lockheed Martin's team includes employees from the company's human resources, information security, communications, legal, ethics, and privacy divisions, says Thomas, who previously served as principal deputy director of Counterintelligence for the U.S. government.

Team members analyze data "for indications of anomalous behavior and to determine where there might be a need for further investigation." They also use data to "proactively identify employees who might be at a higher risk for approach by foreign intelligence officers. This gives us the opportunity to get in front of those employees to provide defensive counterintelligence awareness training."

Thomas says the team also devotes significant time to employee training and awareness. They utilize technology developed at Lockheed Martin called LM WISDOM ITI, which he says analyzes the spectrum of employee attributes, behaviors, and actions that may be indicators of potential insider threat activity. It was built from best practices and feedback from seasoned counterintelligence professionals, as well data sources and modeling behavioral indicators.

"It's important to remember that there is no magic indicator of insider threat behavior," Thomas says. "Look-

ing at previous cases in both government and industry, we can see there are some common pre-event behaviors of concern, including: attempts to expand access to information without clearance or need to know, sudden and unexplained reversal of [a] financial situation, repeated security violations or a blatant disregard for security policies, or a sudden or unexplained withdrawal from co-workers or decline in work performance."

The challenge for IT and security personnel is when upper management does not place a high priority on the importance of proactively detecting insider threats, says Kolish. Network security personnel often feel if software they recommend does not work, "it makes me look stupid," he notes, "so you're less likely to see organizational change. There's less resistance to organizational change when the C-suite takes the perspective of [recognizing] there is risk to the company."

At the same time, companies can and should strike a balance between having a culture of security and respecting employee privacy, says Weber. "There's no reason for a person to be reading email until a red flag goes up," he says. There are people who may be concerned about the 'big brother' perception, but when used correctly, the tools are taking more of a scientific approach to studying individuals, he emphasizes.

"Everyone needs to step back and say when an individual causes material harm to an organization, it affects every single employee," says Weber. "There's an obligation to ensure the workplace and the assets of an organization are

safe so people can continue to be safe and go to work." **C**

Further Reading

The Insider Threat: Understanding the Cyber Security Threats from Personnel, *Federal Technology Insider* <http://bit.ly/1Ka0w7V>

Ramin, M., and Levy, Y. **Securing E-Learning Systems: A Case of Insider Cyber Attacks and Novice IT Management in a Small University. *Journal of Cases on Information Technology*, 2006. Volume 8, Issue 4.** <http://bit.ly/1Spja10>

Hong, J., Kim, J., and Cho, J. **The Trend of the Security Research for the Insider Cyber Threat. *Security Technology, Communications in Computer and Information Science*, Vol. 58, 2009, pp 100-107** <http://bit.ly/1CFL0MS>

Editors: Probst, C.W., Hunker, J., Bishop, M., and Gollman, D. **Insider Threats in Cyber Security. 2010, Springer Science+Business Media, LLC. ISBN 978-1-4419-7133-3** <http://bit.ly/1gDvoB0>

Anderson, D.F., Weaver, E.A., Zagonel, A., Cappelli, D., Gonzalez, J., Mojtahedzadeh, M., Moore, A.P., Rich, E., Sarriegui, J.M., Shimeall, T.J., and Stanton, J.M. **Preliminary System Dynamics Maps of the Insider Cyber-Threat Problem. 2005, Software Engineering Institute, Carnegie Mellon University** <http://bit.ly/108uHKG>

Cyber Council: Insider Threat Task Force A Preliminary Examination of Insider Threat Programs in the U.S. Private Sector. 2013. Intelligence and National Security Alliance. <http://1.usa.gov/1CFLC0r>

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

© 2015 ACM 0001-0782/15/11 \$15.00

Celebrations

ACM Observes Lovelace, Boole Bicentennials

ACM has joined the University of Oxford and the Computer History Museum to celebrate the 200th birthdays of computer visionary Ada Lovelace (1815–1852) and English logician, mathematician, and philosopher George Boole (1815–1864).

Please visit <http://bit.ly/1K8kQki> before Nov. 23 to register for the "Ada Lovelace 200 Symposium: Celebrating the Life and Legacy

of Ada Lovelace," which will take place Dec. 9–10 at the University of Oxford, with ACM president Alexander Wolf, ACM vice president Vicki Hanson, *Communications* editor-in-chief Moshe Vardi, and ACM-W chair Valerie Barr participating, among others.

A related event, the exhibit "Ada Lovelace 1815–1852: Computer Visionary," will be on

view at Oxford's Bodleian Library through Dec. 18. For more information, please visit <http://bit.ly/1YIvTw>.

The Computer History Museum (www.computerhistory.org) in Mountain View, CA, will screen the biographical film "George Boole: Father of Digital Logic" on Nov. 16, followed by a panel discussion, and a lecture on Boole by John Naughton, a Senior Research

Fellow at the Centre for Research in the Arts, Social Sciences, and Humanities at Cambridge University, UK. These programs are free to the public.

The Museum's Ada Lovelace Exhibit will open to the public on Dec. 10. On Dec. 13, the Museum will host "Adapalooza," a community day focused on Ada Lovelace that is open to the public and emphasizes family activities.

Brain Science Helps Computers Separate Speakers in a Crowded Room

People can listen to a single voice amid the hubbub of a cocktail party; algorithms can help computers do it, too.

THE MECHANICS of the human brain and hearing system are providing inspiration for the development of algorithms that could lead to better hearing aids, with researchers combining neural networks and techniques that mirror biological behavior. Yet researchers dealing with the computer science are cautious about pushing the emulation of biology too far.

For decades, researchers in both neuroscience and artificial intelligence have been fascinated by the so-called ‘cocktail party problem’. Cognitive scientist Colin Cherry coined the term in 1953 during a project for the U.S. Office of Naval Research while working at the Massachusetts Institute of Technology (MIT) to find out how it is possible for humans to “recognize what one person is saying when others are speaking at the same time.”

Humans can separate voices that lie in the same frequency range, and may even pull out sentences from background conversation and other noises that are louder. University of Surrey research fellow Andrew Simpson says trying to find out how to do this artificially is enticing: “The cocktail party paradigm is compelling because we are trying to equal the brain on its own terms.”

Such research may lead to better hearing aids that can suppress noise and background chatter while boosting and even cleaning up the speech of the people talking to them. Researchers see other applications in human-machine interaction, particularly for robots working closely with people so they can distinguish background chatter from commands meant for them.



Research from biology is helping to shape the strategy for a number of projects that aim to solve the cocktail party problem. Biologists have found the brain learns to separate out individual voices using a number of cues, including tell-tale harmonic frequencies in an individual voice, as well as the understanding of how sounds change over time because of the shaping of words and consonants.

In 2012, Nai Ding and Jonathan Simon of the University of Maryland published research that claimed ‘phase locking’ is an important method humans use to home in on one speaker; the brain detects the rhythms of that speaker’s speech and filters out others. Repeating sounds also play a role,

according to Josh McDermott of MIT’s Department of Brain and Cognitive Sciences, following research that found the brain stores information on sonic textures, using them to pick out similar sounds from the many others the brain may hear at the same time. Similar techniques can be used by artificial systems, researchers say.

Simpson says, “Mimicking nature gives us a sensible architecture for our neural networks and a cost-effective architecture for our sensors.”

DeLiang Wang of Ohio State University’s Center for Cognitive and Brain Studies adds, “The knowledge of auditory physiology and psychology has helped propel the field, but its role shouldn’t be overstated, and

auditory physiology serves more as inspiration than detailed instruction for algorithm design. Domain knowledge does not have to come from auditory physiology or psychophysics. Much can be gained by analyzing the problem itself, as David Marr advocated in his influential study in computational vision decades ago.”

Wang points to the use of the mel scale of pitches in this kind of research to convert raw audio into a time-frequency representation that can be used as the input to other algorithms. The mel scale mirrors the non-linear way in which the ear interprets pitch. Wang’s team performs a further conversion, called cepstral analysis, which shows harmonic relationships between frequencies more clearly.

“The mel scale is inspired by psychoacoustics, although more from behavioral studies than physiological studies,” says Wang, cautioning, “but literally following auditory neuroscience would not lead one to cepstral analysis or learning-based methods.”

The biggest breakthrough in recent years, Wang notes, is the use of systems that can be trained on large quantities of data. Simpson, Wang, and others have applied deep-learning neural networks (DNNs) to the problem. These systems are themselves inspired by neurons in the brain, although they are extremely simplified compared to their biological equivalents.

Simpson’s work includes an algorithm called Deep Transform that, as well as separating out the audio associated with a single speaker, uses re-synthesis to recreate speech fragments that are heavily obscured by noise in the source audio.

“The use of DNNs has elevated performance to a level that can rival human performance in limited situations,” Wang claims.

The DNNs used in this line of research are based on techniques developed by Geoffrey Hinton and Ruslan Salakhutdinov at the University of Toronto for training large neural networks. The digital neurons process data, such as speech samples or the strength of a frequency at a point in time, by multiplying them with trained weights and feeding them to neurons in subsequent layers that filter the data and recognize structure in the input.

“Mimicking nature gives us a sensible architecture for our neural networks and a cost-effective architecture for our sensors.”

The networks are trained by iteratively adjusting the weights that each neuron applies to its input data to try to minimize the error between the output of the entire network and the desired result. The features that the neural networks learn depend strongly on the way the training data is applied.

Deep learning could be locking into the same low-level speech features as those used by the brain, but without being explicitly programmed to do so. Simpson says of the type of harmonic analysis performed by the brain: “There is no reason to presume that present deep learning approaches do not inherently learn such filters; they probably do. In a recent paper, I demonstrated that DNN learning is mirrored in the crest factor of the filters learned; filters get sharper as the network learns.”

Other features such as tracking speech rhythms may call for bigger, more processing-intensive DNNs. Simpson adds: “If a deep neural network is exposed during training to useful information at this level of abstraction, and features sufficient layers to demodulate the envelopes, then it will learn filters to exploit it. These filters would be broadly equivalent to the ‘modulation filters’ of the auditory system. However, in practice, this means very long time windows.”

Being able to deal with those longer time windows may call for the use of more complex neural-network structures. However, Simpson says of these types of neural networks that their

ACM Member News

STAYING OPEN TO OPPORTUNITIES



Valerie King’s work on randomized algorithms is an apt analogy for her approach to life and career

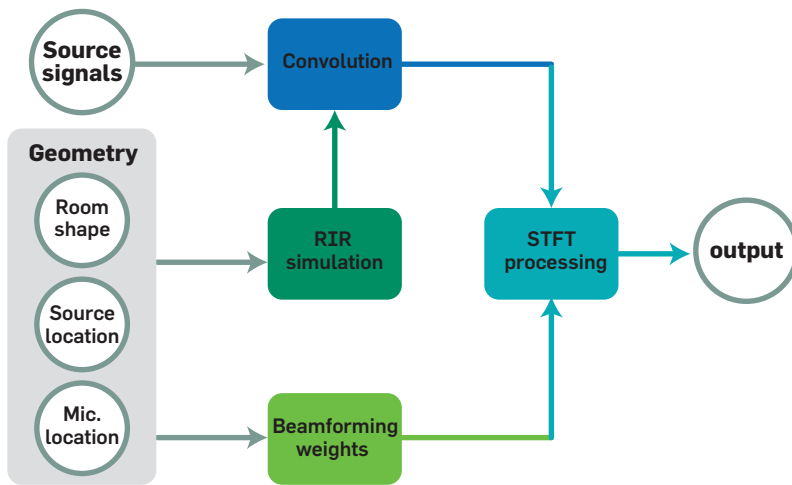
choices. King, a professor of computer science at the University of Victoria, British Columbia, since 1992, never planned on a computer science career. “I follow where things lead me and keep myself open to opportunities,” she says.

King, a native of Long Island, NY, received her undergraduate degree in mathematics from Princeton University in 1977, then went to Berkeley School of Law at the University of California, Berkeley, which she graduated from in 1983. She became a practicing attorney, but a chance airplane encounter resulted in a career switch. King had studied artificial intelligence and legal reasoning based on case-study analysis in graduate school, intending to focus on automating the legal research process. “I was on a plane celebrating taking the bar exam; my seatmate knew someone working in automated legal research. It convinced me to go back to Berkeley and get my doctorate in CS,” King says.

Regarding the paper “Byzantine Agreement in Polynomial Expected Time,” which she co-authored with Jared Saia in 2013, King recalls, “The solution came to me while I was in a workshop looking at the mountains. Big ideas and solutions don’t necessarily take a long time.”

King says she has experienced a rise in her creative energy now that her children are grown, giving her more time to devote to personal and professional projects. She currently is researching a method for finding a broadcast tree in a network, obviating the need to send messages over every link. “There are huge advances in distributed computing and dynamic graph algorithms. I’m excited to be part of it.”

—Laura DiDio



Flowchart of the simulation setup used for numerical experiments on an acoustic rake receiver to validate theoretical results of its efficiency in suppressing an interferer.

“abstract learning seems poorly understood at present.”

DNNs can be trained on audio data, but researchers tend to find better results if the inputs receive some preprocessing. Simpson says, “I’ve implemented versions of Deep Transform both in the time domain and using time-frequency representations. Results are better using the time-frequency representations. The same lesson comes from biology.”

Adds Wang, “A successful learning algorithm needs both good features and a good learning machine. DNN plays the latter role. Preprocessing, or feature extraction, is as important. Typical feature extraction involves time-frequency analysis and subsequent filtering, to extract amplitude-

“A successful learning algorithm needs both good features and a good learning machine. DNN (deep-learning neural network) plays the latter role.”

modulation patterns, for example.”

Although DNNs offer a good technique for handling monaural audio, some researchers, particularly those looking at robotic applications, have focused on the spatial cues made possible by the use of two or more microphones, moving further away from what biological systems do. Wang says although the human brain could use the difference in sound reaching the left and right ears to handle the cocktail party problem, these cues are used mainly just to identify where a speaker is, and they are easily confused by echoes from walls. However, fed with clear binaural data, Simpson’s Deep Transform seemed to train itself primarily on spatialization cues.

“The deep learning approach is very general and will learn whatever abstract feature space it is able to exploit to minimize the cost function during training. The binaural gradient provides a very convenient separation plane and I would interpret the resulting binaural Deep Transform as having learned ‘spatial filters’.”

Using more traditional signal processing approaches rather than deep learning, Tobias May and colleagues from the University of Oldenburg and Philips Research used a combination of voice-frequency detection and binaural localization to filter out individual speakers within a room of many, among other noise sources and echoes that can make localization difficult.

Other approaches have even made

use of confusing echoes. Martin Vetterli and colleagues at the Swiss Federal Institute of Technology in Lausanne (EPFL) borrowed a concept from cellular communications. The 3G cellular protocols introduced the concept of the rake-receiver algorithm, which is highly effective at separating the main signals from echoes caused by radio waves bouncing off the sides of buildings and other surfaces. The ability to detect the echoes turned out to be useful in speech processing; even if a loud interferer stands in the way of the person to which the system is tuned, the algorithm can analyze the echoes to successfully separate and amplify their speech, according to the researchers.

Other teams have used additional microphones to try to provide algorithms with more spatial clues, although Simpson suggests combining the input from cameras with the audio data so robots can link lip movements with sounds might provide greater gains.

Further advances can come from building larger deep-learning systems, Simpson says. “We will see upscaling, from neural networks of a billion parameters, to massive parallel GPU farms running hundreds of thousands of deep nets with cumulative parameters reaching to the range of trillions. Source separation is entering the arms race in a big way.” **C**

Further Reading

Narayanan A., and Wang, D.
Improving Robustness of Deep Neural Network Acoustic Models via Speech Separation and Joint Adaptive Training, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol.23, No.1 (2015)

McDermott, J.
The Cocktail Party Problem, *Current Biology*, Vol.19 No.22 (2010)

Simpson, A.
Deep Transform: Cocktail Party Source Separation via Probabilistic Re-Synthesis, *arXiv preprint arXiv:1502.04617* (2015)

Dokmanić, I., Scheibler, R., and Vetterli, M.
Raking the Cocktail Party, *IEEE Journal on Selected Topics in Signal Processing*, August (2015)

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

© 2015 ACM 0001-0782/15/11 \$15.00

John H. Holland 1929–2015

JOHN HENRY HOLLAND, a pioneer in the study of complex adaptive systems and of what became known as genetic algorithms, died in August at the age of 86.

Holland, a longtime professor of computer science and engineering and professor of psychology at the University of Michigan, Ann Arbor (where he founded and led the Center for the Study of Complex Systems), also was a Santa Fe Institute (SFI) professor and external professor for many years and, at the time of his passing, a member of the Institute's Board of Trustees and Science Board.

He had been interested for six decades in what are now called complex adaptive systems, starting with his early work at IBM in the 1950s on computer simulations of Hebb's theory of cell assemblies. He formulated genetic algorithms, classifier systems, and the Echo models as tools for studying the dynamics of such systems.

Holland was born in Fort Wayne, IN, in 1929. He studied physics at the Massachusetts Institute of Technology and received a B.S. degree in 1950, then studied mathematics at the University of Michigan, receiving an M.A. in 1954. In 1959, he received the first computer science Ph.D. from the University of Michigan.

In 1975, Holland published the book *Adaptation in Natural and Artificial Systems*, intended to be the foundation for a general theory of adaptation, introducing genetic algorithms as a mathematical idealization to develop his theory of schemata in adaptive systems. Genetic algorithms eventually became widely used as an optimization and search method in computer science.

Holland's Learning Classifier System statistical learning system incorporated a reinforcement learning algorithm for non-Markovian environments. His 1989 book *Induction*, co-written with psychologists Keith Holyoak and Richard Nisbett, applied his ideas about classifier systems to induction in cognitive science.



An early champion of interdisciplinary approaches to science and engineering, many of Holland's lesser-known projects anticipated trends and advances in modern computer science. He trained generations of graduate students, many of whom have taken inspiration from his interdisciplinary approach to make seminal contributions of their own.

Holland received the MacArthur Fellowship in 1992, and was a fellow of the World Economic Forum. He received the Louis E. Levy Medal from The Franklin Institute in 1961, and the MacArthur Fellowship in 1992.

Stephanie Forrest, a Distinguished Professor of computer science at the University of New Mexico in Albuquerque, observed while Holland had been her Ph.D. advisor, "he was so much more."

"In 1979, when I walked into the Computer and Communications Department at the University of Michigan looking to take a computer programming course, I was sent down the hall to talk with John, at that time the associate chair of the department. Forty-five dizzying minutes later, I walked out with an application form for the Ph.D. program, and the rest is history. As I recall, our conversation ranged from Godel's Incompleteness and Undecidability results to flying gliders to some weird ideas I couldn't parse about combining genetics and computers. Since that time, John never failed to be interested and encouraging of my ideas, and over the years he offered up great, if nonstandard, career advice. I credit most of my own limited success in the academic world to John's

great example and inspiration (and the many, many letters of recommendation he wrote on my behalf)."

Melanie Mitchell, a professor of computer science at Portland State University, recalled that when she started graduate school, "I didn't know anything about the CS department. It was my great luck that I ended up taking John Holland's class 'Adaptation in Natural and Artificial Systems', [which] completely changed my perspective on what computer science was (and should be) about."

Holland, she said, "became my co-advisor at Michigan, and did so much to support and encourage me in my work. He let me know when I had done enough for my Ph.D. (and told me it was time to get my 'union card', as he called it). He recommended me for the Michigan Society of Fellows, and then invited me to join the famous BACH group during my fellowship. He also invited me to visit SFI, first for a summer, and then asked me to direct SFI's Adaptive Computation program. This led to my faculty appointment at (and hopefully lifelong engagement with) the Institute."

"I was fortunate to become one of John's close-knit group of former Ph.D. students, all of us, including John, meeting every now and then to talk about everyone's research projects and to speculate on big questions. Our last meeting was in fall 2014; John, in spite of his illness, was in great spirits, and regaled us with his new ideas and enthusiasms. In addition to his great intellect, John was perhaps the most enthusiastic, cheerful, and lively person I've ever known. I'll miss him greatly."

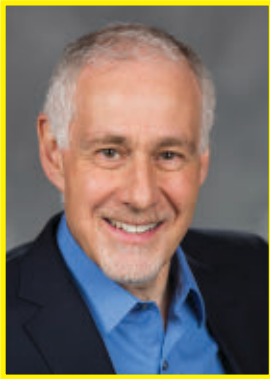
"An Interview with John H. Holland" appeared in *ACM SIGEVOLUTION*, the newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation, in 2008. It may be accessed in the Digital Library at <http://dl.acm.org/citation.cfm?id=1562109>. □

—ACM Staff

© 2015 ACM 0001-0782/15/11 \$15.00

ACM

ON A MISSION TO SOLVE TOMORROW.



Dear Colleague,

Computing professionals like you are driving innovations and transforming technology across continents, changing the way we live and work. We applaud your success.

We believe in constantly redefining what computing can and should do, as online social networks actively reshape relationships among community stakeholders. We keep inventing to push computing technology forward in this rapidly evolving environment.

For over 50 years, ACM has helped computing professionals to be their most creative, connect to peers, and see what's next. We are creating a climate in which fresh ideas are generated and put into play.

Enhance your professional career with these exclusive ACM Member benefits:

- Subscription to ACM's flagship publication ***Communications of the ACM***
- Online books, courses, and webinars through the **ACM Learning Center**
- Local Chapters, Special Interest Groups, and conferences all over the world
- Savings on peer-driven specialty magazines and research journals
- The opportunity to subscribe to the **ACM Digital Library**, the world's largest and most respected computing resource

We're more than computational theorists, database engineers, UX mavens, coders and developers. Be a part of the dynamic changes that are transforming our world. Join ACM and dare to be the best computing professional you can be. Help us shape the future of computing.

Sincerely,

A stylized, handwritten signature in black ink.

Alexander Wolf
President
Association for Computing Machinery



Association for
Computing Machinery

Advancing Computing as a Science & Profession

SHAPE THE FUTURE OF COMPUTING.

JOIN ACM TODAY.

ACM is the world's largest computing society, offering benefits and resources that can advance your career and enrich your knowledge. We dare to be the best we can be, believing what we do is a force for good, and in joining together to shape the future of computing.

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

- ☐ Professional Membership: \$99 USD
- ☐ Professional Membership plus
ACM Digital Library: \$198 USD (\$99 dues + \$99 DL)
- ☐ ACM Digital Library: \$99 USD
(must be an ACM member)

ACM STUDENT MEMBERSHIP:

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

- ☐ **Join ACM-W:** ACM-W supports, celebrates, and advocates internationally for the full engagement of women in all aspects of the computing field. Available at no additional cost.

Priority Code: CAPP

Payment Information

Name _____

ACM Member # _____

Mailing Address _____

City/State/Province _____

ZIP/Postal Code/Country _____

Email _____

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc., in U.S. dollars or equivalent in foreign currency.

☐ AMEX ☐ VISA/MasterCard ☐ Check/money order

Total Amount Due _____

Credit Card # _____

Exp. Date _____

Signature _____

Purposes of ACM

ACM is dedicated to:

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

Return completed application to:
ACM General Post Office
P.O. Box 30777
New York, NY 10087-0777

Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

Satisfaction Guaranteed!

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.



Association for
Computing Machinery

1-800-342-6626 (US & Canada)
1-212-626-0500 (Global)

Hours: 8:30AM - 4:30PM (US EST)
Fax: 212-944-1318

acmhelp@acm.org
acm.org/join/CAPP

Privacy and Security Security Assurance

How can customers tell they are getting it?

SECURITY ASSURANCE HAS been the most challenging topic I have had to contend with in my 45 years working on computer security. While designing and building security features for user identification and authentication or for access control can be challenging, the most difficult task is assurance: making systems that can resist attack. Assurance affects not only security features but also any system component that will respond to untrusted input.

Assurance is achieved by integrating security into the process of designing, building, and testing systems. If security is well integrated into the development process, the resulting software will generally resist attack. In this column, I summarize my experience building a process for security assurance and how and why it works. I also describe providing customers with confidence a system achieves an appropriate degree of assurance.

Creating a Process

While I started working on problems of assurance in the early 1970s, my perspective on assurance today is especially influenced by my 10 years of experience as the director of the team

responsible for Microsoft's Security Development Lifecycle (SDL). The story of Microsoft's work to integrate security assurance into development has been told in the trade press and professional papers.^{2,5,7} The company stepped up its focus on security in late 2001 in response to its concerns about the security of Microsoft products and to customer feedback—much of it heated—about the security vulnerabilities exploited by Code Red and Nimda worms.

It was clear the company's security challenges were not limited to security features but encompassed any component that dealt with potentially untrusted input. For that reason, Microsoft stopped *all* development on major products—Windows Server 2003 was the largest. This was a major commitment—it involved thousands of engineers, delayed the release of products by months, and impacted costs and revenues—but it was clear that an error handling untrusted input could occur anywhere.

Our team trained the entire development staff on techniques we believed would help improve products' resistance to attack, and required the developers to apply those techniques to the code for which they were responsible.

Developers used tools and code review to search for vulnerabilities, changed defaults to reduce attack surface, and added mitigations that made it more difficult to exploit any vulnerabilities that might remain. This initial approach—the security push—reduced the prevalence and severity of security vulnerabilities in products that had gone through security pushes. It also motivated developers to build secure software.

The Slammer and Blaster worms of 2003 affected product versions that had not yet gone through security pushes and gave Microsoft clear evidence that security needed to be an integral part of its development process and culture. In 2004, Microsoft introduced SDL as a mandatory process for software development. Those of us who worked on the SDL knew security must be built into products and the best approach was to integrate security into the development workflow. In summary, the SDL required that threat models be produced as a product was being designed, that specific coding requirements be met and static analysis and attack surface analysis tools be applied as the code was being written, that mitigations be integrated into software, and that security testing (for



example, fuzz testing) be conducted as part of product qualification.

While the initial Security Development Lifecycle was released with little in the way of supporting tools, Microsoft updated the process to make it more effective and efficient. The SDL team built and mandated tools to support security assurance in design, development, and testing. We treated product vulnerability reports as feedback that told us whether we needed to update the SDL or do a better job of executing it. We introduced a tracking system that automated the task of ensuring product components had met the applicable requirements. The tools and tracking help ensure the SDL is followed, and provide a consistent mechanism for engineers.

By 2005–2006, the customers who had been providing heated feedback realized the company was serious about security and the SDL. Interactions with chief security officers (CSOs) became more cordial: in some cases, they urged other software vendors to adopt processes similar to the SDL. Whether because of that urging or because they had applied similar logic to ours, during the mid-2000s a number of other vendors developed

processes similar to the SDL and adoption of such processes across the industry has continued to grow.^a

Is a Process Enough?

When customers ask about secure development, they do so because they need products that will resist attack. Ideally a metric would enable customers to compare products without having to delve into the way products were built. Unfortunately, the quest for a measure of product security has been futile, and I expect it will continue to be so. We can measure the number and severity of reported product vulnerabilities, but when we do that, we are measuring the talents and interests of vulnerability researchers as well as inherent product security. A product that no one uses may have no reported vulnerabilities, but be extremely vulnerable if attacked.

Customers understand that measuring security is infeasible so they ask about a developer's process. They also understand that vulnerabilities will occur—both because processes are not perfect and because new attacks get

discovered—so they want a process that incorporates continuous improvement.

So why do the SDL and similar processes improve product security? I believe an effective process has to meet three tests:

- *It must be fundamentally sound.* It must incorporate measures that will improve security if applied. Fundamental soundness implies continuous improvement: as new classes of vulnerabilities or techniques for building more secure software are discovered, the process must be updated.

- *It must be adapted to the development organization.* Individual developers must be trained and motivated (security must be part of the culture), elements of the process must be integrated into development, and development tools must reflect the requirements of the process. For example, if the process says, “do static analysis,” the development organization must select a static analysis tool, “tune” it to their code base and identify “must-fix” errors.

- *It must affect the code.* Threat models result in a set of work items that must be addressed so the delivered code responds correctly to threats. Static analysis tools or fuzzers find errors that must be analyzed and fixed in the code. The SDL meets all three tests but differ-

^a For a non-exhaustive list of companies that have adopted SDL-like processes, see <http://www.safecode.org>.

entiates itself by meeting the third. The industry has many processes—CMMi is one example—that stop with documentation. If a process does not actually affect code, it is just paper.

How Can Customers Tell?

If customers accept the proposition that an SDL-like process is effective, they are still left with a question whether their suppliers are actually implementing such a process in a way that affects delivered code. A supplier should be able to describe their secure development process, and in particular how the process affects code. In practical terms, this means describing not just process generalities, but what specific tools are used, what errors are “must fix,” and how the results of security analysis are managed in the supplier’s work item management system.

Some customers have documented their expectations for their suppliers’ secure development processes, and put “teeth” into those expectations by having in-depth interviews with suppliers’ secure development staffs.^b This is a good approach although it has scaling problems: suppliers’ secure development staffs will not scale to participate in interviews with every customer and not every customer has the resources to interview every supplier. The customer’s interviewers must have hard-to-find expertise to evaluate what they are hearing and to consistently assess processes across a range of suppliers. But the experience of such interviews appears beneficial—it informs customers and provides useful feedback to suppliers.

A new ISO standard—ISO 27034-1: Information technology—Security techniques—Application security—recognizes the importance of developer process and of ensuring delivered code reflects the requirements of the process.³ The standard is still evolving, but I believe it will provide a solid basis for consistently assessing suppliers’ secure development processes. The final standard is planned to address all three of the tests listed previously, in-

^b For one example, see Third Party Software Working Group, Appropriate Software Security Control Types for Third Party Service and Product Providers, Financial Services Information Sharing and Analysis Center; <http://bit.ly/1Fw6jhs>.

Perfect assurance would be great, but we do not have the tools or techniques to achieve it, and I do not see any reason we ever will.

cluding the link between process and delivered code. Some customers have expressed an interest in using the standard to drive procurement decisions, supplanting vendor interviews.

Some Things That Don’t Help

When I started working in computer security, I expected we would formally verify that a secure system corresponded to a mathematical model, and that the system’s security was correct down to the source code; this approach failed. Formal verification was not able to cope with systems large enough to be useful and no customers wanted the systems that were simplified to enable that approach to be applied.⁶

Earlier this year, Dorothy Denning wrote a *Communications Privacy and Security* column advocating legal liability for developers who released products vulnerable to attack.¹ Unfortunately, the list of such developers is “all of them.” If a liability regime were put in place, I expect it would result in a lot of lawsuits against software developers and slow software innovation as developers attempted legally rather than technically defensive coding. If the goal is to create software that is more secure and usable by customers, encouraging the use of best practices that address threats is a more rational approach than liability.

Denning’s proposal includes an “out” for developers who release the source code for their products. That “out” might incentivize developers to release their source code and change some business practices, but I do not believe it would improve assurance. Fifteen years ago, some researchers

argued that open source software was inherently more secure than closed, but faced with reality that argument faded away.⁸ The recent spate of vulnerabilities in the open source OpenSSL package is one example: the Linux Foundation’s Core Infrastructure Initiative is bringing developers and users of open source software together with the aim of introducing secure development practices—many similar to the SDL—to widely used open source software.⁴

What Next?

Security assurance is a challenge for developers and a necessity for customers. Perfect assurance would be great, but we do not have the tools or techniques to achieve it, and I do not see any reason we ever will. We can achieve practical assurance if we commit to practical measures and apply them.

► Secure development processes work if they reflect continuous improvement and are followed by the people who design, implement, and test code.

► Assessment of secure development processes is feasible, but it will take a lot of work and it must consider impact on delivered code, not merely process. ISO 27034 represents a way forward here.

► Neither product liability nor open source will serve as a “silver bullet” that substitutes for secure development processes that are rigorously applied and continuously improved. **■**

References

1. Denning, D.E. Toward more secure software. *Commun. ACM* 58, 4 (Apr. 2015), 24–26.
2. Howard, M. and Lipner, S.B. *The Security Development Lifecycle*. Microsoft Press, 2006.
3. ISO/IEC. ISO/IEC 27034-1:2011. Information technology—Security techniques—Application security—Part 1: Overview and concepts; http://www.iso.org/iso/catalogue_detail.htm?csnumber=44378.
4. Linux Foundation. Core Infrastructure Initiative site; <https://www.coreinfrastructure.org/>.
5. Lipner, S.B. The trustworthy computing security development lifecycle. In *Proceedings of the Twentieth Annual Computer Security Applications Conference* (Tucson, AZ, 2004).
6. Lipner, S.B., Jaeger, T., and Zurko, M.E. Lessons from VAX SVS for high assurance VM systems. *IEEE Security and Privacy* (Nov.–Dec. 2012).
7. Microsoft Corporation. Life in the Digital Crosshairs, 2014; <http://bit.ly/1NnOgS4>.
8. Panel: Security and Source Code Access: Issues and Realities. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2000.

Steven B. Lipner (lipner@outlook.com) retired in 2015 as Partner Director of Software Security, Trustworthy Computing, at Microsoft Corporation. He continues to consult with and advise organizations on computer security and assurance.

Copyright held by author.



Legally Speaking Software Patents Are Falling Down

Implications of the U.S. Supreme Court's new test for patentable subject matter.

IN *ALICE V. CLS BANK*, the U.S. Supreme Court ruled that a computer-implemented method and system for facilitating settlements of financial transactions was unpatentable subject matter as an abstract idea. In the year or so since that June 2014 decision, many computer-implemented and other software-related patents have been struck down. Moreover, patent infringement lawsuits dropped by 13% in 2014, which a PricewaterhouseCoopers 2015 Patent Litigation Study reports lowered the value of software patent assertions that may be a result of the *Alice* decision. Litigating software patents is now a much riskier proposition for plaintiffs than in the past.

This column discusses the background to the *Alice* decision and the Court's new test for patentable subject matter. It then gives examples of software-related patents that have been invalidated on subject matter grounds when tested in litigation. Some software patents may survive subject matter challenges, but still be at risk of invalidation as indefinite after the Federal Circuit's ruling in *Williamson v. Citrix*.

Background to *Alice v. CLS Bank*

Novel and nonobvious machines, manufactures, compositions of matter, and processes are eligible for patenting so long as these subject matters are properly claimed and vetted through the patent examination process. Because software is a virtual machine and a technological process, some contend that software in-



novations should always qualify as patentable inventions. However, since the Supreme Court's decision in *Gottschalk v. Benson* in 1972, this contention has been, in a word, contentious.

At issue in *Benson* was whether a method for transforming binary coded decimals to pure binary form was eligible subject matter for patenting. One claim covered carrying out this method with the aid of a general-purpose computer; a second claim was for the method more generally, so that a person who carried it out with the aid of paper

and pencil would infringe it. The U.S. Supreme Court ruled that both claims, despite being set forth as a set of steps in method form, failed to recite patentable subject matter.

The Court's rationale for its *Benson* decision is not a model of clarity. The Court was clearly troubled by the abstractness of *Benson*'s method claims, which could be infringed, assuming a patent issued, in a very wide array of applications. The mathematical character of the *Benson* algorithm raised the specter of a patent impeding use of a

fundamental building block of knowledge. The Court also worried about issuing patents to non-technological innovations that would, in effect, cover mental processes.

The Court revisited the patentability of software-related inventions in 1981 in its *Diamond v. Diehr* decision. *Diehr* upheld the patentability of a rubber-curing process that included a software component. From the mid-1980s and to the late 2000s, the Court of Appeals for the Federal Circuit adopted an expansive interpretation of patentable subject matter. For a time, it appeared everything under the sun made by humans was patentable subject matter as long as it produced a “useful, concrete, and tangible result” (a test that never resulted in the invalidation of any issued patents). Under this standard, hundreds of thousands of software-related and computer-implemented patents were issued in those decades.

Benson Revived in Alice

To make a long story short, the Supreme Court in *Alice v. CLS Bank* de-

cided the *Benson* decision was good law and the Federal Circuit’s expansive interpretation of patentable subject matter was erroneous. (My November 2013 *Communications Legally Speaking* column correctly predicted the Supreme Court would invalidate *Alice*’s patent.)

In *Alice*, the Court decided that abstract methods, such as one facilitating settlements of financial transactions (so that money would only flow when the parties to the transaction had completed their obligations), did not become patentable merely by being carried out through use of a general-purpose computer. Nor did such an abstraction become more patentable if claimed as a machine (or system) to carry out the functions instead of as a process (or method).

To determine if a claimed innovation is patentable subject matter, the Court in *Alice* directed courts to look, first, at the claim language to discern if it recites a law of nature, a natural phenomenon, or an abstract concept, and if it does, then to consider whether the claim as a whole contains additional

elements that “transform the nature of the claim” so that it satisfies patent subject matter requirements.

The Court ruled that *Alice*’s patent recited an abstract concept and the claims as a whole did not satisfy the “something more” second step of the test because they were simply for computer implementations of the abstract concept. However, the Court also implied that software inventions will be patentable subject matter if there is an “inventive concept” in either an improvement in “the functioning of the computer itself” or in “any other technology or technical field.”

Software Patents After Alice

Since *Alice*, the U.S. Patent and Trademark Office (USPTO) has issued revised guidelines on patentable subject matter.^a These guidelines offer numerous examples of software-related inventions, distinguishing among claims that are subject matter eligible under *Alice* and those that are likely not.

The software patents that courts have struck down in the last year or so have generally been those the USPTO issued during the decades in which the Federal Circuit had, in effect, opined that all software-related inventions were patentable subject matter. Had *Alice* been decided in 1990 instead of 2014, many of the computer-implemented and software-related patent claims filed with the USPTO in those years might well have been rejected on subject matter grounds.

The context in which these patents have been falling has largely been litigation in which a challenged firm defends against a claim of patent infringement by attacking the validity of the asserted patent or in which a firm threatened with patent infringement asks a court to declare the asserted patent is invalid on subject matter grounds. Some computer-implemented and software-related patents have survived subject matter challenges, but about three-quarters of the challenges have been successful so far.

Among the many software patents that have fallen when challenged are these:

- Hulu successfully attacked an UL-

^a See <http://1.usa.gov/1gFDI2X>.



**MASTER OF SCIENCE IN
COMPUTER SCIENCE**

With a focus on network security, the MSCS positions students with advanced technical skills essential to today’s programmers.

- One-year or part-time formats
- Evening classes
- Internship and networking opportunities
- Competitive tuition
- Scholarships and financial aid
- Optional test scores

www.merrimack.edu/gradcs
graduate@merrimack.edu • 978-837-3563

**Merrimack
COLLEGE**
NORTH ANDOVER, MA

tramercial patent on a method and system for using advertising as a currency on the Internet in exchange for consumers' getting access to content.

► Costco defeated a patent directed to a computer-implemented system and method for online mortgage shopping that allowed users to assess their borrowing capabilities without revealing their identities that Mortgage Grader, Inc., had obtained from the USPTO.

► Capital One successfully challenged a patent owned by Intellectual Ventures for an online banking method designed to help users create and stick to a budget. The claims were for a method of storing a user profile of specific budgets for each category of transactions the user anticipated, and then transmitting a summary of transactions within each category, so the user could discern if he was on or off budget.

But courts have upheld some software-related patent claims. Intellectual Ventures, for instance, successfully defended a patent aimed at screening computer data for viruses within a telephone network before communicating the data to an end user. Not only was the computerized screening not something a human could do, but the specification described the need for at least three computers to be configured in a specific manner as illustrated by three flowcharts. To practice the claim, it was necessary to imitate the receiving computer's configuration to determine whether a virus in an executable file would infect the receiving computer.

Importance of *Williamson v. Citrix*

Software patents are also being challenged on the grounds the claims are too indefinite to satisfy patent standards. *Williamson v. Citrix* is a case in point. The plaintiff got a patent on a method and system for distributed learning in a virtual classroom environment in which one or more presenters connect with a geographically distributed audience.

(*Williamson* is a trustee of the now bankrupt At Home Corp. that had obtained this patent before the firm went belly up. Adobe, Cisco, IBM Corp., and Microsoft are among the other defendants in *Williamson's*

Complaints about "bad" software patents have been legion among computing professionals.

patent case against Citrix. The patent was an asset of the defunct corporation *Williamson* has been trying to monetize so that holders of bonds in the company could get some compensation for the unfortunate investments they made.)

A trial judge ruled in favor of Citrix and its fellow defendants on all of *Williamson's* patent claims. *Williamson* appealed this loss. Because the trial judge gave an unduly narrow interpretation to some claims, the Federal Circuit overturned the trial judge's ruling on those claims and sent the case back for further proceedings.

However, the Federal Circuit upheld the trial judge's invalidation of eight of *Williamson's* claims because they were too indefinite to satisfy patent standards. The court construed these eight claims as means-plus-function claims, even though the patent did not specifically use the usual means-plus-function language.

Here is some context: U.S. patent law allows patent applicants to claim a means for carrying out a specific function as long as the application discloses particular structures, materials, or acts in the patent specification (which explains the invention in relation to the prior art). This disclosure is necessary to support a means-plus-function patent because the structure, materials, or acts are what makes the claim definite enough so that someone who wanted to do the same function would be able to tell if he or she was infringing the patent or had adopted a non-infringing way to do the same thing.

Under prior Federal Circuit decisions, *Williamson* would have enjoyed a strong presumption the claims at issue were definite enough to survive a challenge because the patent did not use the conventional means-plus-function language. But the Federal Circuit in *Williamson* decided the claims at issue could nonetheless be construed as means-plus-function ones. And because the eight challenged claims lacked the necessary recitation of structures, materials, or acts, the claims were too indefinite to be valid. The Federal Circuit recognized it should not elevate form over substance. Courts can now interpret claims as being means-plus-function ones even if the magic means-plus-function words are missing.

This decision is important because patent lawyers have often sought to broaden the scope of software patents by claiming methods and systems for performing particular functions. By avoiding use of the usual means-plus-function language, the lawyers hoped to attain a very broad scope for the patent, while insulating the patentee from an indefiniteness challenge by omitting the conventional means-plus-function words from the patent claim.

Conclusion

Complaints about "bad" software patents have been legion among computing professionals. Courts seem to be responding to these concerns by developing some new tools to strike down these patents. The most significant tool thus far is the new *Alice* test for patentable subject matter. But the *Williamson* decision may prove to be another important tool with which courts can strike down software patents that are too broad and indefinite. Some may also be struck down for as obvious. Congress is also considering a new package of reforms to curb abuses of the patent system. No silver bullet will solve all software patent problems, but these measures are important steps in the right direction. ■

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

Copyright held by author.

Economic and Business Dimensions Electronic Health Records and Patient Safety

Examining the effects of electronic health records on the safety of patients in medical facilities.

U.S. HEALTHCARE HAS made huge investments in health information technologies (IT). The U.S. Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009 earmarked more than \$20 billion to foster electronic health records (EHRs) at U.S. hospitals and other medical facilities, and facilities have spent billions of their own to digitize patient records and clinical workflows. What benefits have accrued? Have EHRs lowered the cost and improved the quality of healthcare? In particular, what has been the effect of EHRs on patient safety?

There is some evidence that EHRs reduce costs over the long term and under the right conditions.^{2,a} But evidence is scant on the effect of EHRs on patient safety. An Institute of Medicine (IOM) 2012 study, *Health IT and*

a EMR detractors have asserted EMR implementations also impose high cost to hospitals and physicians—both direct financial costs of implementation and maintenance and for the medical providers, the indirect costs of increased documentation imposed by EMRs. For a polemical example of such an argument, see Charles Krauthammer, “Why Doctors Quit,” *The Washington Post* (May 28, 2015); <http://wapo.st/1FdVEX4>.

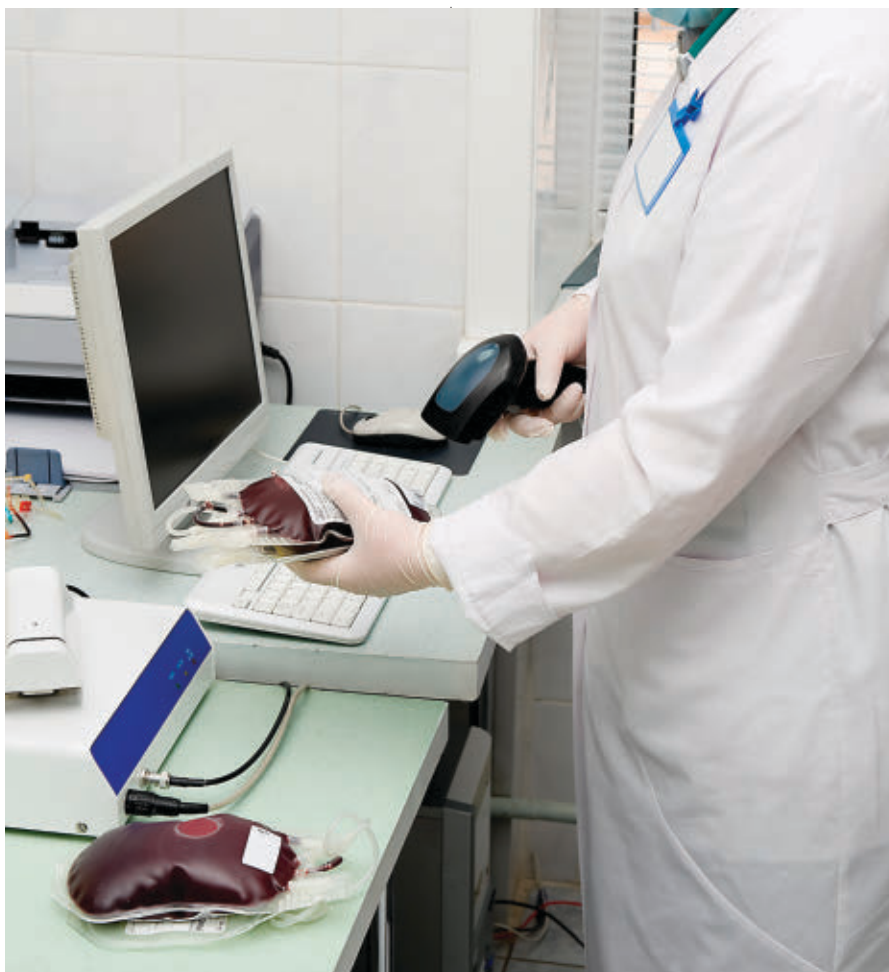


IMAGE FROM SHUTTERSTOCK.COM

Patient Safety, concluded, “current literature is inconclusive about the overall impact of health IT on patient safety.” This lack of evidence prompted an econometric study of patient safety at Pennsylvania (PA) hospitals. Patient safety improved for Pennsylvania hospitals that adopted EHRs: a 27% decline in overall patient safety events and a 30% decline in medication errors.^b

What Is Patient Safety?

Patient safety can be described as “freedom, as far as possible, from harm, or risk of harm, caused by medical management (as opposed to harm caused by the natural course of the patient’s original illness or condition).”⁴ A patient safety event (PSE) occurs if a patient is harmed or unnecessarily placed at risk of harm. Every year PSEs affect hundreds of thousands of patients in the U.S. and cost billions of dollars.

Medical errors and the harm they cause have been seen as unavoidable side effects of modern medicine or the result of incompetence. Lucian Leape says, “many errors are preventable and many are evidence of system flaws not character flaws.”⁷ The patient safety movement was brought to the medical mainstream by a report of the Institute of Medicine *To Err is Human*,⁵ with the goal to eliminate preventable patient harm through improved systems and find solutions to previously “unpreventable” errors.¹³

It is difficult to distinguish between known and acceptable risks and risks due to medical mismanagement. Concerns for patient privacy add to this problem. Finally, medical providers see PSEs on their watch as stains on their professional competence and may be hesitant to report them. Measuring PSEs is difficult. A popular approach is to use algorithms from the Agency for Healthcare and Research and Quality (AHRQ) to infer patient safety indicators (PSIs) from billing data. But these are viewed skeptically by the medical community. Peter Pronovost, a prominent Johns Hopkins physician, called them, “truly, near worthless,” during a U.S. Senate hearing on patient safety.¹²

It is difficult to distinguish between known and acceptable risks and risks due to medical mismanagement.

AHRQ algorithms for PSIs can generate false positives if medical conditions deemed markers for PSIs were present on admission, if there were coding errors, or if the coding lacks specificity.⁹ Coding is notoriously inconsistent across hospitals, or even within hospitals. Global trigger tools (GTTs), a set of defined rules applied in retrospect to identify trigger events that help trained analysts identify PSEs, are also viewed with skepticism. Their large random sample makes them a superior measure of PSEs, but they are difficult to use due to high costs in their manual review process. Of course, PSEs can be measured through voluntary or mandatory reporting, but this has its own problems related to underreporting.

The medical community has experimented with many interventions to reduce patient safety events: harm reduction initiatives, training programs, and checklist use programs (to reduce surgical errors). The latter provides a good example. The World Health Organization (WHO) asserts, “in medicine, as in aviation, checklists can help ensure consistency and completeness in carrying out complex tasks,” has encouraged surgical checklists, and has been working on other checklists for medical care.¹⁴ However, the impact of these interventions is not always clear. A study published in the *New England Journal of Medicine* (NEJM) found no significant reductions in PSEs in a random sample of North Carolina hospitals, even though these hospitals had implemented patient safety initiatives.⁶ Another NEJM published study found no relationship between reduced operative mortality and surgical safety checklists, although prior studies had found beneficial effects.¹¹ Showing

Calendar of Events

November 1–4

SenSys '15: The 13th ACM Conference on Embedded Network Sensor Systems, Seoul Republic of Korea, Co-Sponsored: ACM/SIG, Contact: Junehwa Song, Email: junesong@cs.kaist.ac.kr

November 2–3

COSN'15: Conference on Online Social Networks, Palo Alto CA, Sponsored: ACM/SIG, Contact: Anesh Sharma, Email: anesh.x.sharma@gmail.com

November 2–6

ICCAD '15: IEEE/ACM International Conference on Computer-Aided Design, Austin, TX, Co-Sponsored: Other Societies, Contact: Diana Marculescu, Email: dianam@cmu.edu

MSWiM'15: 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Cancun Mexico, Sponsored: ACM/SIG, Contact: J.J. Garcia-Luna-Aceves, Email: jj@soe.ucsc.edu

November 4–5

BuildSys '15: The 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, Seoul Republic of Korea, Co-Sponsored: ACM/SIG, Contact: David Culler, Email: culler@eecs.berkeley.edu

November 8–11

UIST '15: The 28th Annual ACM Symposium on User Interface Software and Technology, Charlotte, NC, Co-Sponsored: ACM/SIG, Contact: Celine Latulipe, Email: clatulip@uncc.edu

November 9–13

SIGUCCS '15: ACM SIGUCCS Annual Conference, St. Petersburg, FL, Sponsored: ACM/SIG, Contact: Leila Shabbender, Email: leila@princeton.edu

^b The study is described in detail in a longer paper available on SSRN: <http://ssrn.com/abstract=2503702>

clear benefits to patient safety from interventions has been difficult.

EHRs and Patient Safety

Our econometric study of EHRs and patient safety attempted to get around the measurement difficulties by using data from a mandatory patient safety reporting system. The Pennsylvania legislature mandated all PA hospitals to report patient safety events to the Patient Safety Authority, a PA state agency created by the “Medical Care Availability and Reduction of Error (MCARE) Act.” MCARE encourages more accurate reporting by guaranteeing that PSE data is confidential, reducing provider and hospital concerns about malpractice lawsuits or loss of reputation. Providers and hospitals are less concerned about malpractice lawsuits or loss of reputation. Such confidentiality might seem unfair to consumers, but providers and hospitals are unlikely to disclose PSE otherwise and the data creates opportunities for learning and quality improvement that can benefit consumers and that otherwise might not occur.

Electronic health records are health-care applications that digitize patient records and clinical workflows. EHRs might consist of a Clinical Data Repository (CDR) that stores patient data, a Clinical Decision Support System (CDSS) that assists providers by providing reference information and suggestions for care, a Computerized Provider Order Entry (CPOE) that enable providers to electronically place orders, and a Physician Documentation (PD) system that consolidates clinical notes across hospital departments. Errors in modern medicine occur because of work complexity, knowledge intensiveness, and the variety and volatility of circumstances.¹⁰ Tools that reduce these problems might reduce medical errors. Well-designed EHRs might improve patient safety by improving communications, making knowledge accessible, providing decision support, requiring key pieces of information for correct treatment, assisting with calculations, performing real-time checks, and assisting with monitoring.¹

To give an example, an elderly patient with multiple chronic diseases and taking several medications might benefit from a well-designed EHR (for example, with a reconciled medication list for the patient) when treated for new

The medical community has experimented with many interventions to reduce patient safety events.

medical conditions. The EHR can provide a physician with real-time access to the patient's data, notes from previous encounters, drug-drug interaction alerts and drug-disease interaction alerts. These might influence the physician when ordering new medications, or the hospital pharmacist to infer the correct route of administration when following physician notes to fill the prescription. Hospitals with well-designed EHRs might improve safety for such patients.

One can imagine an ideal empirical test for whether EHRs affect PSEs. It would involve a randomized control trial (RCT) in which a randomly chosen subset of hospitals would adopt and use EHRs while those in a control group would not, so systematic differences in PSEs might be attributed to adopting or not adopting EHRs. Because such an experiment is completely unfeasible and too costly to perform, we adopted the next best approach using observational data. Given that adoption of EHRs in U.S. hospitals occurred over time, we exploited within-hospital variation and controlled for time trends to estimate the effect of EHR on PSEs. This difference-in-difference research approach has been successfully employed to estimate the impact of EHRs on PSEs.

In our study, approximately 30% of the PA hospitals had adopted advanced EHRs in 2005, with adoption growing to approximately 70% by 2012.^c We found hospitals that adopted EHRs experienced a 27% decline in overall PSEs, with a 30% decline in medication PSEs. We think these results are significant, and they have been seen in other studies. Parente and McCullogh found a small beneficial effect of EHRs on three specific PSIs (infections, postoperative hemorrhage, and postoperative pulmonary embolism).⁸ Freedman et al. found

beneficial impact of CPOE on four post-operative PSIs that may be amenable to decision support.³ These two studies have results directionally similar to ours. EHR adoption and concomitant work flow changes can lead to patient harm but those problems do not challenge the potential value of EHRs in reducing PSEs. EHRs should not contribute to medical errors, but on average, EHRs seem to be improving patient safety. **■**

c Our hospital EHR adoption data was sourced from the Healthcare Information and Management Systems Society (HIMSS). We define advanced EMRs as CPOE or PD, following Dranove et al.²

References

1. Bates, D.W. and Gawande, A.A. Improving safety with information technology. *New England Journal of Medicine* 348, 25 (2003), 2526–2534.
2. Dranove, D., Forman, C., Goldfarb, A. and Greenstein, S. The trillion dollar conundrum: Complementarities and health information technology. *American Economic Journal: Economic Policy* (2014); <http://bit.ly/1itnxao>.
3. Freedman, S., Lin, H., and Prince, J. Information technology and patient health: An expanded analysis of outcomes, populations, and mechanisms. (June 3, 2014) SSRN Scholarly Paper, <http://bit.ly/10ViedS>.
4. Great Britain House of Commons Health Committee. *Patient Safety: Sixth Report of Session 2008–09, Vol. 1: Report, Together with Formal Minutes*. The Stationery Office, 2009; <http://bit.ly/10tSY09>.
5. Institute of Medicine. *To Err Is Human: Building a Safer Health System*. 1 edition. National Academies Press, Washington, D.C., 2000.
6. Landrigan, C.P. et al. Temporal trends in rates of patient harm resulting from medical care. *New England Journal of Medicine* 363, 22 (Nov. 25, 2010), 2124–2134; doi:10.1056/NEJMsa1004404.
7. Leape, L.L. Error in medicine. *JAMA* 272, 23 (Dec. 21, 1994), 1851–1857; doi:10.1001/jama.1994.03520230061039.
8. Parente, S.T. and McCullogh, J.S. Health information technology and patient safety: Evidence from panel data. *Health Affairs* 28, 2 (Mar. 1, 2009), 357–360; doi:10.1377/hlthaff.28.2.357.
9. Rosen et al. Validating the patient safety indicators in the veterans health administration. *Medical Care* 50, 1 (Jan. 2012), 74–85; doi:10.1097/MLR.0b013e3182293edf.
10. Spear, S.J. and Schmidhofer, M. Ambiguity and workarounds as contributors to medical error. *Annals of Internal Medicine* 142, 8 (Aug. 2005), 627–630.
11. Urbach, D.R. et al. Introduction of surgical safety checklists in Ontario, Canada. *New England Journal of Medicine* 370, 11 (Mar. 13, 2014), 1029–1038; doi:10.1056/NEJMsa1308261.
12. U.S. Senate Subcommittee on Primary Health and Aging. More than 1,000 preventable deaths a day is too many: The need to improve patient safety; <http://1.usa.gov/1JXulDM>.
13. Wachter, R. *Understanding Patient Safety*. Second Edition. McGraw-Hill Professional, New York, 2012.
14. World Health Organization. The checklist effect; <http://bit.ly/1KcLVas>.

Muhammad Zia Hydari (zia@cmu.edu) is Assistant Professor of Business Administration at the University of Pittsburgh, PA.

Rahul Telang (rtelang@andrew.cmu.edu) is a professor of Information Systems and the Ph.D. program chair at Heinz College, Carnegie Mellon University, Pittsburgh, PA.

William M. Marella (wmarella@ecri.org) is the MBA program director at Pennsylvania Patient Safety Authority, Harrisburg, PA.

This research was supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

Copyright held by authors.

Education

Programming in K–12 Science Classrooms

Introducing students to visual programming as a pathway to text-based programming.

INTRODUCING COMPUTATIONAL MODELING as the way of “doing science” in K–12 classes can greatly increase students’ participation in computing.^{3,4} But, how can K–12 science teachers, who may have little or no programming experience, teach such classes? And, how can teachers adopt computational modeling in a manner that not only deepens science learning, but also supports the development of authentic programming experiences?

Since 2012, we have been conducting studies in which 3rd, 4th, 5th, and 8th-grade (ages 8–10, and 13 years old) public school teachers and students, with no prior experience in programming, use programming as part of their regular science curricula throughout the academic year. The students we worked with were predominantly African American (100% in 3rd, 4th and 5th grades, and 70% in 8th grade), and came from low socioeconomic status backgrounds. The programming language used by these teachers is ViMAP, an agent-based visual programming language and modeling platform (see <http://www.vimapk12.net>) that uses NetLogo⁴ as its simulation engine. In agent-based programming, users create simulations by programming the behaviors of, and interactions between, computational agents. We designed ViMAP specifically: to enable yearlong curricular integration of programming as a means for modeling and simulating relevant scientific phenomena in K–12 classrooms



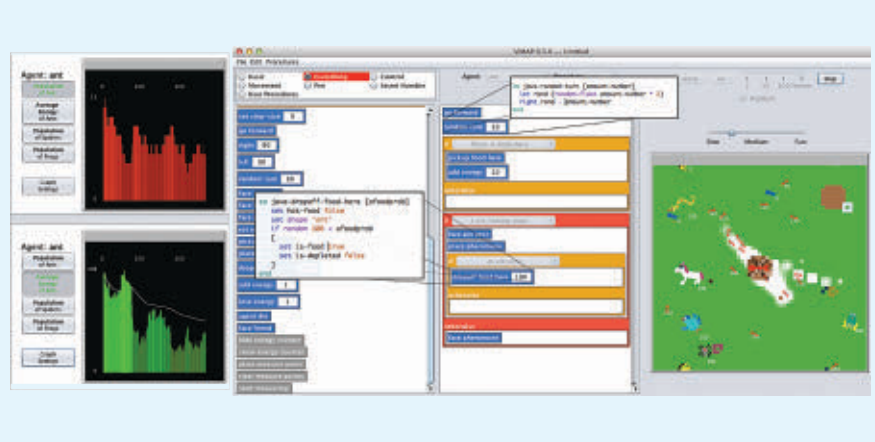
Fourth-grade students in Tennessee programming ViMAP-Ants models in order to conduct simulation experiments and develop mathematical explanations about ecological interdependence. The programming commands were developed for them by 8th-grade students.

using visual programming; and to create pathways to transition from visual to text-based programming in higher grades. Students in 3rd–5th grades used visual programming, while students in 8th grade used both visual and text-based programming.

In our studies, teachers and students built and refined ViMAP models twice a week throughout the academic year. Teachers led the classroom instruction. In weekly meetings, researchers pro-

vided ongoing professional support by collaboratively designing the curricular activities with the teachers, based on their state and district-mandated curricular needs. Curricular activities spanned kinematics (3rd and 4th grades) and ecology (3rd, 4th, and 8th grades). For kinematics, students programmed a single-agent NetLogo simulation using ViMAP (ViMAP-Motion), whereas in ecology, they programmed the behavior of multiple agents and agent types in

ViMAP-Ant-Foodweb simulation and programming commands developed by 8th-graders. Popped-out images show NetLogo procedures underlying the ViMAP commands created by the students. Left-to-right: Graphs of population and energy; library of ViMAP commands; a sample ViMAP program; and the NetLogo simulation controlled by the ViMAP program.



either a bird-butterfly-flower ecosystem (ViMAP-Bird-Butterfly), or ant colonies with multiple levels of predators (ViMAP-Ants).^a Students in 3rd, 4th, and 8th grades showed significant gains in our assessments of computational thinking and statewide-standardized tests in science and math. Students' gains in the standardized tests were significantly higher than the gains in other classrooms in the same schools that did not use ViMAP or any other form of programming with their science curricula.

We found teachers integrated programming in their curricula in the following ways:

Reframing programming as the design of mathematical measures. All of the teachers we worked with told us their students experienced difficulties in interpreting and constructing mathematical measures (for example, units of measurement and graphs) in their regular science classes. Manipulating units is emphasized in statewide standardized assessments, so it is an important learning goal. Our teachers reported that programming in ViMAP helped students overcome these challenges, because the activity of programming the behavior of agents requires the learners to define the event in discrete measures. The state of the simulation, at any instant, represents a single event in the form of spatialized representations of agent actions and interactions. To “run” the simulation, these events are repeated a number of

times specified by the user. By engaging in iterative cycles of building, sharing, refining, and verifying ViMAP models, students refine their understanding of what actions and interactions of agents represent an “event,” which are then displayed on graphs. The students are getting the opportunity to explore different kinds of units, and see their simulation measured in those units.

Embodied modeling supports agent-based thinking and programming. Programming an agent involves learning to think like the agent, because it can help students understand the relationship between their code and the simulated output. All the teachers saw embodied modeling as a valuable activity for teaching students how to think like an agent. Embodied modeling introduced the students to the relevant computational rules represented by the ViMAP commands, helped them debug their programs and deepened their understanding of the graphs. For example, in 3rd grade, students first conducted an embodied activity in which they acted

All the teachers saw embodied modeling as a valuable activity for teaching students how to think like an agent.

as butterflies and foraged for nectar, obeying simple rules that connected the butterflies' energy to their movement and foraging. Teachers often referred to the students' embodied modeling experiences while discussing the relevant ViMAP graphs, and while assisting students debug their ViMAP programs.

Programming commands and code visualization scaffolds help deepen conceptual understanding in science. Some of the programming commands in ViMAP were designed to directly represent scientific concepts. For example, while simulating ecosystems, students can use programming commands such as “make same color offspring” to specify the agents' reproductive behavior. Similarly, when students are building a ViMAP simulation of acceleration, they can use commands such as “change step-size,” which alters the turtle's speed—its step-size—by a specific amount. So, for students, reasoning about and discussing their ViMAP model and simulation involves thinking and talking about the relevant scientific concepts. ViMAP also offers scaffolds for code visualization: code-step-through or “highlight” functionality, and time-delay. This means that as a ViMAP program is running, the command currently being executed is highlighted in red, and users can also set a “time-delay” between the execution of command blocks. Students use these scaffolds to deepen their understanding of the relationships between their ViMAP programs and simulated outputs, and they learn the relevant scientific concepts by engaging in discussions with their peers and the teacher about how the commands in their programs affect the simulations and the graphs.²

Fostering a classroom culture of sharing and critiquing models. The teachers periodically brought their classes together by creating a culture of “sharing and critiquing” peer models, that is, their ViMAP programs, simulations and graphs. In this process, the class as a whole, normatively, developed criteria for what would count as a “good” computational model. In the course of four months of modeling motion, the 3rd-grade students and their teacher developed the following criteria: “a good model should accurately represent some important aspects of reality,” “a good model can be approximate,” and “a good model can be used

a The ViMAP software and associated curricula can be obtained from <http://www.vimapk12.net>.

to make good or accurate predictions.” These criteria originated in teacher-led class discussions as socially defined (voted by popular choice), but over time, became progressively more grounded in students’ mathematical explanations of relevant aspects of their ViMAP simulations. This led students to use progressively more sophisticated computational abstractions, such as loops and conditionals, in order to make their models predictive.

Transition from Visual to Text-Based Programming

Although visual programming lowers the barrier for entry into programming, learners who intend to pursue careers in computing may find the drag-and-drop nature of visual programming inauthentic, or find it difficult to transition to text-based programming.¹ We designed ViMAP explicitly to address this issue. Visual programming commands in ViMAP are defined as short NetLogo procedures (see the accompanying figure), which students can easily access and modify. In 8th-grade classes, after engaging in visual programming with ViMAP for approximately two months to build simulations of interdependence in ant ecosystems, the teacher and the students wanted to make deeper changes in the underlying text-based NetLogo code. But, given the limited instructional time, the teacher found it challenging to help students create new simulations in NetLogo using text-based programming. This required a lot of “overhead,” because the language syntax was often disconnected from the relevant scientific concepts.

To address this issue, the teacher then decided to return to the ViMAP-Ants unit, and asked the students to work in small groups to create new ViMAP commands by modifying and extending the underlying NetLogo code (see the figure). For the 8th-graders, this work was motivated by a capstone project of designing and creating a version of ViMAP-Ants in order to teach 4th-graders about food webs, which is a required curricular standard in 4th grade. The teacher introduced the students to relevant “chunks” (procedures) in the NetLogo code pertaining to specific ViMAP commands they were already familiar with. She led class discussions

The transition from visual to text-based programming by using text-based programming to create visual programming blocks can take students even deeper into both programming and science.

in which the students collaboratively interpreted and explained the significance of the computational abstractions in NetLogo code in terms of relevant scientific concepts represented in the ViMAP commands. Learning the syntax and new forms of abstractions (such as classes) in text-based programming therefore became deeply intertwined with the relevant concepts in ecology (for example, hierarchy of organisms in food webs). While students’ previous work using visual programming introduced them to computational abstractions such as loops, variables, and conditionals, text-based programming further deepened their computational thinking because it involved creating computational objects or classes, declaring new local variables, creating and modifying conditionals, editing and repurposing lists, and using random numbers. Students’ growth in computational thinking was further evident in a post-assessment activity, in which they successfully created new commands for a NetLogo simulation of a different ecosystem without teacher-provided assistance.

Conclusion

Long-term integration of computing in K–12 science is important not only for broadening participation in computing, but also because it can deepen students’ understanding of scientific concepts beyond what is possible in

the traditional classroom. At the same time, students develop the *key* scientific practice of modeling, especially, mathematical modeling. Agent-based computation, in particular, domain-specific, agent-based programming languages such as ViMAP, can serve a powerful role in this regard. Elementary and middle school science teachers with no prior programming experience can indeed successfully incorporate such programming platforms in their curricula by reframing programming as the means for designing, interpreting, and constructing units of measurement and graphs. Furthermore, when programming does become the *language* of science throughout the academic year, especially in higher grades, it is essential to introduce text-based programming, so that students can take on computational modeling challenges beyond the reach of visual programming. This transition can take students even deeper into both programming and science, by making it possible to use text-based programming to create visual programming blocks for modeling science. 

References

1. DiSalvo, B. Graphical qualities of educational technology: Using drag-and-drop and text-based programs for introductory computer science. *IEEE Computer Graphics and Applications* 36, 6 (June 2014), 12–15.
2. Farris, A.V. and Sengupta, P. Perspectival computational thinking for learning physics: A case study of collaborative agent-based modeling. In *Proceedings of ICLS 2014*, 3, (May 2014), 1102–1107.
3. Sengupta, P., Kinnebrew, J.S., Basu, S., Biswas, G., and Clark, D. Integrating computational thinking with K–12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies* 18, 2 (Feb. 2013), 351–380.
4. Wilensky, U., Brady, C.E., and Horn, M.S. Fostering computational literacy in science classrooms. *Commun. ACM* 57, 8 (Aug. 2014), 24–28.

Pratim Sengupta (Pratim@vimap12.org) is a professor of Learning Sciences at the University of Calgary, Alberta, Canada.

Amanda Dickes (Amanda.C.Dickes@vanderbilt.edu) is a Ph.D. student in the Learning Sciences Program at Vanderbilt University, Nashville, TN.

Amy Voss Farris (amy.voss@vanderbilt.edu) is a Ph.D. student in the Learning Sciences Program at Vanderbilt University, Nashville, TN.

Ashlyn Karan (ashlynk@gmail.com) is an 8th-grade science teacher at Liberty Collegiate Academy, Nashville, TN.

David Martin (kitmartin@gmail.com) is a Ph.D. student in the Learning Sciences Program at Northwestern University, Evanston, IL.

Mason Wright (masondw@umich.edu) is a Ph.D. student in the department of computer science at the University of Michigan, Ann Arbor.

Funding for the ViMAP software and associated curricula is provided by NSF CAREER Award #1150230.

Copyright held by authors.

Interview

An Interview with Fred Brooks

ACM Fellow and A.M. Turing Award recipient Fred Brooks reflects on his career.

ACM FELLOW FREDERICK (“Fred”) Brooks, recipient of the 1999 A.M. Turing Award, has made landmark contributions to computer architecture, operating systems, and software engineering. After earning a Ph.D. in Applied Mathematics and Computer Science from Harvard under the legendary Howard Aiken, he worked for IBM on several landmark computer systems, most notably the System/360 series that came to dominate main-frame computing for decades. He left IBM in 1964 to found the Computer Science Department at the University of North Carolina, from which he retired at the end of the Spring 2015 semester.

Noted software architect Grady Booch conducted an oral interview of Brooks in Cambridge, U.K., in September 2007. The complete transcript^a of this interview is available in the Computer History Museum’s oral history archive; presented here is a condensed and highly edited version designed to whet your appetite.

—Len Shustek

Falling in Love with Technology

I was born in Duke University Hospital because my father was teaching chemistry at the University of North Carolina, and that was the nearest hospital at the time. When I was about six months old he decided to change careers and



Fred Brooks with a statue created in his honor at the University of North Carolina Department of Computer Science.

go to medical school, so we settled in Greensboro, N.C. It was a great place to grow up because of the superb school system.

My favorite subject was physics, but I always had a fascination with business equipment. When the local corset factory went out of business, I bought filing cabinets for \$4 apiece and a Burroughs comptometer that I think

I paid \$35 for. I made my own McBee card keysort system^b for keeping track of my map collection. So I have always been fascinated with the kind of machines that process information. When I was 13, I read in a *Time* maga-

^a <http://www.computerhistory.org/collections/catalog/102658255>

^b An edge-notched card filing system invented in 1896; see http://en.wikipedia.org/wiki/Edge-notched_card

zine in the little town library about the Harvard Mark I computer and I knew from then on that was what I wanted to do.

Studying Computers at Harvard

I did my undergraduate major at Duke University in physics, but I also studied math, economics, and accounting as well as the humanities because that was also my interest. The head of the physics department was determined that his best students should go to Harvard to study physics, and two of us did. When I told my advisor there that I really want to study computers, he said, “Fred, you’re too late to get in on the ground floor, but you can catch the first landing.” That meant I had a chance to meet and really get to know J. Presper Eckert, John Mauchly, and Konrad Zuse. Howard Aiken was my thesis advisor. So I got to really know the pioneers even though I was not in that generation, and that has been a great joy.

I went into the computation lab, a very small group that was part of a division of engineering in applied physics. There was Aiken, who was the boss, and two young instructors, Ken Iverson and Bob Minnick. Every day the boss was in town the whole crowd gathered for coffee at 5 P.M. in the machine room, which had the Harvard Mark I on one side and the Harvard Mark IV on the other side, each about 60 feet long. We would talk for a half-hour or 45 minutes, and then he would go home to supper and we would go back to work.

Programming the Harvard Mark IV

We programmed the Mark IV starting the first semester. My present colleague Bill Wright and I undertook, for our first year project, to write a program that would analyze melodies and create synthetic melodies using an eightfold Markovian process. We chose common meter hymns because we could find a lot of them that had the same metrical structure. We transposed them all to the same key, then we analyzed the transition probabilities of the melodies. It took us three years to get done, but we got music you could pass off on any choir.

The Mark IV was programmed in decimal absolute. It had 230 registers,

When I was 13, I read about the Harvard Mark I computer and I knew from then on that was what I wanted to do.

10,000 words of program storage, and 4,000 words of backup drum storage. There wasn’t an assembler, but Aiken had designed a relay box that enabled you to encode it algebraically. Our first program for determining the Markov frequencies ran about 2,500 instructions. We were allowed two one-hour slots on the machine during the semester, so we did extensive desk checking. It is the only big program I ever wrote that ran right the first time.

Howard Aiken as an Advisor

At the end of my first year Aiken told Ken Iverson “I would like you to prepare a course on the application of computers to business.” Nobody had ever taught any such course anywhere in the world. I had had trouble with a course in boundary value problems and was not continuing my NSF fellowship, so I went to Ken and said, “Can I be your teaching assistant, because that’s right down my alley?” Out of that came our book *Automatic Data Processing*, which went through an edition based on the IBM 650 and then six years later an edition based on the 360. Ken was fully as important in my education as Aiken was.

For my dissertation, Aiken said “I want you to design a machine specialized for payrolls.” That sounded like a good topic, even though he was more interested in the machine and I was more interested in the design methodology—how you get from the requirements to the machine.

His hypothesis was that by specializing a machine for payroll you could make significant improvements in cost performance. That turned out not to be the case. By specializing for serial file processing you could get

an improvement, and that was essentially the machine I designed for my dissertation, but there were no further economies to be had by making it specialized for payroll.

Aiken was a very impressive person, and did something I cannot do with my students: he came to my office every day during the year I was working on my dissertation and wanted to see the prose that had appeared since the day before. And guess what? Some had appeared since the day before.

Helpful Summer Jobs

I was offered a summer job at Marathon Oil Company in Findlay, OH, using an IBM 650 to mechanize payroll. That was priceless experience. The complexities in business data processing are that you are trying to lump a wide variety of cases under a single process, so typically on the 650 when it was doing business applications, half of all instructions would be conditional branches. That is not true in scientific computing.

It helped me as a computer architect that I was also doing scientific computing. I spent a summer at North American Aviation doing missile tracking and database building. A summer at Bell Labs trying to identify which party on a four-party line was dialing this call. A summer at IBM in Endicott, NY, where I learned punched card machines and was in the physics department doing acoustics. Those four summers were a priceless part of my education, and they exposed me to radically different corporate cultures and radically different geographies.

Working on IBM's Supercomputer

Steve Dunwell, who was project manager for Stretch, came through Harvard in 1956 looking for people. His offer was very attractive: a chance to work on the world’s fastest computer. I leapt at the opportunity. My wife Nancy, whom I had met the first Sunday night at Harvard, worked on the transistor circuits, and I worked with the architects under Werner Buchholz.

IBM had been thinking about it for a year, and had just signed the contract with the National Security Agency to build Harvest, a variant of Stretch for breaking wire-wheel cryptographic codes. I was the only one of the archi-

fects proper who was native-born, so I could get the security clearance to work on it. Stretch, as the host, was about 15-feet long and 5-feet high and 5-feet deep. Harvest was a “plug-in card” that added another 20 feet of specialized electronics. It processed about 4 million bytes a second, and had about 250 bytes of instruction to set it up that would take half a day to write.

The purpose of Stretch was to make the fastest machine we could, cost no object. That is both liberating and tempting, but we did not succeed. Dunwell had set out to make a machine 100 times faster than the IBM 704. But the memories available were only six times faster, and the circuits were about 10 times faster. The idea was that by using more of everything we could get there, and you can’t. We only got to 50 times faster.

It was declared a disaster and withdrawn from the market. But later IBM recognized Stretch technology enabled the 7090s, 7080s, and 7074s, and that the concepts became crucial for the System/360. Tom Watson, to his credit, went back and got Steve Dunwell out of disgrace and made a special award to him recognizing the important influence the Stretch had on the company’s welfare.

Figuring Out IBM’s Future

The Data Systems Division, which was the middle of the market in computers, recognized they had a product problem with their many existing lines. The vice president for engineering appointed a committee to study what a successor product line architecture might look like. I had moved to the research division, but I chaired that committee.

At the end of the summer I was asked to come back to the Data Systems Division as manager of architecture. We undertook to design a new product line called the 8000 series, built around Stretch concepts. There was a small scientific computer, small and mid-sized business machines, a grown-up version for high performance use by insurance companies and utilities, and a grown-up version for scientific computing. We worked very hard, and in January of 1961 we had cost estimates and market forecasts, plus a plan that involved creating new markets by making these ma-

chines communications oriented.

In January we did an all-day presentation of the 8000 series with the brass up from Armonk and White Plains. The whole program was very well received, except for one fellow sitting in the back who just got glummer as the day went on. That was Vin Learson, and that is not who you want to get glummer, because he was executive vice president of the company. Well, that night he fired my boss. He brought in Bob Evans and told him “If it’s right, make it happen; if it’s wrong, turn it around.”

Bob spent three weeks looking into it, then took me out to dinner at a restaurant in Poughkeepsie and told me he had decided it was wrong. We were losing market share to the Seven Dwarfs.^c Everybody was out after us. We were obsolete. We were fundamentally address-size limited. We could not attach more memory to the 7090 or the 7080, and the applications were hungry for more memory.

He proposed to wait for a new semi-integrated circuit technology that was going to be three years down the road. The problem is, how do you hold the market in the meantime? My plan would get out there now, although it had some fundamental difficulties.

He was right and I was wrong. He had two parts of his vision. One was that we ought not to do a new product line for the Data Systems Division; we ought to do a new product line for the IBM company, big and little. The other was that we ought to make the new product line coincide with the new technology.

We fought back and forth. We went to the Corporate Management Committee in March, and I won. That did not slow Bob down a bit. Bob is unstoppable. He went to the Corporate Management Committee in May, and he won. It was over. This meant stopping all the 8000 series projects in the Poughkeepsie lab and reassigning all the people. His plan was to do temporizing machines—the 7094, the 7080 model 3, and so forth—to hold the market as best we could until we could get there with the new product line using the new technology.

I had gone to a retreat up at Saratoga Springs to spend a week ironing out who is going to do what, and to

make sure my boys landed on their feet in the reassignment. I was on the way out back to research when, to my utter amazement, Bob asked me to take the new product line. The crown jewels!

I was dubious. We had been fighting pretty hard. But Jerry Haddad, who was Bob’s boss at the time, said, “I never knew anybody that regretted working for Bob Evans.” So I gave it a try, and it went really well. We clicked, and we fought shoulder to shoulder, side by side, the rest of the way. Bob is one of four great bosses I had in my life.

Designing the System/360

Architecturally we started out in the data systems division pursuing a stack architecture, which is what Burroughs had done. It turns out that works great if you’ve got a transistor register for several levels, and it does not work great if you’ve got it all in memory, and you are having to pull and push all the time.

The problem is addressing. If you put addresses big enough on the littlest machine, your littlest machine is serial by byte and that means you spend a lot of cycles fetching address bytes that you are never going to use, and that compromises your performance. The stack architecture was a way of addressing that by not fetching as many addresses, but when we went through the performance and cost estimation in March 1962, the stack machine was working fine from the middle size up but was not competitive below because of all this pushing data in and out of memory. We had accepted the assignment of making the whole product line upward and downward compatible with one architecture.

So we scrapped it and had a design competition, which was Gene Amdahl’s idea. I think there were 12 or 13 internal teams, mostly of three or four people. I said the decision of the judge will be final. Gene’s team and Jerry’s team came in with essentially the same concept, which was to use the base registers that Philco had introduced in a machine a year or so earlier. Base registers meant we could get by with short addresses.

There was one very big difference. Gene’s machine was based on the existing 6-bit byte and multiples of that: 24-bit instructions and a 48-bit instruction or floating point. Jerry’s

^c Burroughs, Sperry Rand, Control Data Corporation, Honeywell, General Electric, RCA, and NCR.

machine was based on an 8-bit byte and 32-bit instructions, so 64-bit and 32-bit floating point. This is not a real happy choice. There are strong arguments each way, but you want your architecture to be consistent. You are not going to have an 8-bit byte and 48-bit instruction floating-point word.

It was our biggest internal fight. Gene and I each quit the company once that week, but Mannie Piore, the senior scientist in the company and a person of great wisdom, got us back together. I had made the decision for the 8-bit byte. Gene appealed to Bob; but Bob affirmed it. Of all my technical accomplishments, making the 8-bit byte decision is far and away the most important. The reason was that it opened up the lowercase alphabet. I saw language processing as being another new market area that we were not in, and could not get into very well as long we were doing 6-bit character sets.

When we announced the 360 in April 1964, I said to my team is “Tonight the lights will be on in the other guys’ offices”—those of our Seven Dwarfs

competitors—because the project had been kept pretty well secret. And the orders just started pouring in.

Operating Systems

In 1962 we formulated a plan to have four software levels, known as Romans I, II, III, IV. Because we had hardware compatibility across the line, the software levels had to be distinguished only by the memory size. W.B. McWhirter, who was the Data Systems Division president, took the software away from us, but they were busy, and they did not give the 360 software much attention.

Before the announcement I said to Bob, “Look, the machines are being released to the factory. The hardware part is done. The machines are on the track. Everything’s rolling. But the software is in an utter mess. Between now and September, when I will leave to help found the Computer Science Department at the University of North Carolina at Chapel Hill, let me go over there and bail, and just see what can be done with the boat.”

I changed teams, and we had a retreat off in the woods in February

1964. We came back with a totally different product plan that involved these memory levels, compatibility among them, a variety of language processors—two FORTRANs, a Report Program Generator, two levels of COBOL—and a modular operating system that would start at 16K. You have got to leave some space for the application programs, so we said the operating system has to be resident in 4KB. It turned out we could not do that, so we ended up making 32KB the minimum size machine for OS/360. We had a tape-based operating system, a card-based operating system, and a little disk operating system called Basic Operating System. Those got delivered on schedule in February 1965.

But the big operating system was in trouble. We patched and bailed, but by summer it was clear that we were still in trouble. Tom Watson invited me down to Armonk for one of these one-on-one luncheons in the executive dining room and said, “Why don’t you stay here?” I said that I really wanted to get back closer to the technical level. “Well, will you stay another year and keep on bailing out the operating system? If you

NEW BOOKS FROM THE MIT PRESS



HANDBOOK OF COLLECTIVE INTELLIGENCE

edited by Thomas W. Malone and Michael S. Bernstein

“... ambitious and invigorating. Whether looking to start your research career or simply curious to understand what collective intelligence is about, you will find no better starting point than this book.”

—Duncan J. Watts, Principal Researcher, Microsoft
Hardcover | \$30 | £20.95

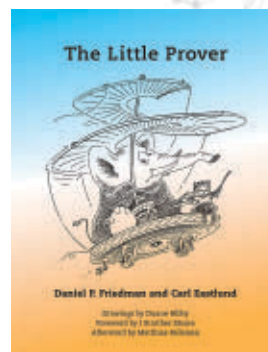


REGULATING THE CLOUD

Policy for Computing Infrastructure
edited by Christopher S. Yoo and Jean-François Blanchette

The emergence of the cloud as infrastructure: experts from a range of disciplines consider policy issues including reliability, privacy, consumer protection, national security, and copyright.

Information Policy series | Paperback | \$30 | £20.95



THE LITTLE PROVER

Daniel P. Friedman and Carl Eastlund

foreword by J Strother Moore
afterword by Matthias Felleisen
drawings by Duane Bibby

“... Friedman and Eastlund, two jolly characters, eschew talk of bugs and bombs, and strip [computational theorem proving] down to its essence as only a *Little* book can. Want proof? Look inside!”

—Shriram Krishnamurthi, Professor of Computer Science, Brown University
Paperback | \$34 | £23.95

mitpress.mit.edu

will, we'll send somebody to Chapel Hill to teach your courses. You go one week a month to get your department organized. And when you need a new computer in Chapel Hill, we will help." That was a good offer, so I said yes.

The workforce peaked at about 1,000. We did deliver the first version in April or May of 1965, but it was slow. It was November, really, before a respectable version was delivered.

The worst decision, which is documented in *The Mythical Man Month*,^d was to take the software away from the architecture group and give it to the operating system manager. The architecture manager had said, "If you leave it with me, it'll be the same amount late, it will cost the same, but it'll be right." He was right and I was wrong, and that was a multimillion-dollar mistake.

Languages

Assembly language, surprisingly enough, posed a lot of technical problems. There were two entirely different schools of thought. On the commercial side, the assembly language served as a platform on which high priests in corporations wrote macros and the troops programmed in the macro language. In the scientific community, the scientists would write their own macros and they programmed in macro-enhanced assembly language. The question of how to resolve all this led me into the thickest technical thickets I got into.

The other set of technical decisions had to do with PL/I. One serious question is: Do we do independent evaluation of expressions so that you can factor out common sub-expressions? The answer was yes, but that leads you to peculiarities such as one divided by three equals zero because of data typing.

The basic concept was to make a universal programming language that would meld and displace FORTRAN and COBOL. ALGOL was not really a factor, although part of our product plan included delivering an ALGOL compiler for government reasons. ALGOL was popular in Europe and had a lot of important concepts in it, particularly having to do with subroutine

There is still an awful lot we do not know about how to build complicated systems and keep them under control.

calls and parameter passing, which we had to master and adopt. An important step forward in PL/I was the provision of compile-time mode, just like macro assembler.

If we had been smart, we would also have done a schedule-time mode of PL/I instead of doing JCL, the Job Control Language. But we weren't smart. The worst mistake we made was JCL. Its existence was a mistake. Building it on a card format was a mistake. Building it on assembly language was a mistake. Thinking of it as only six little control cards instead of a language was a mistake, because it had no proper subroutine facilities, no proper branching facilities. That we did not see it as a language was the fundamental problem; we saw it as a set of control cards.

Except in the U.K., IBM waffled about its support for PL/I. They did the same thing with APL—on again, off again—when a wholehearted support would have made it happen. Now, could you have displaced FORTRAN? No, I now think that would have been impossible. I think you could have displaced COBOL, whose community is more coherent. It is more top-down oriented, and PL/I is a better substitute for COBOL than it is for FORTRAN.

How Big the Project Was

My hardware development budget was about \$100 million. That doesn't count all the I/O devices, the disks, the tapes, the new keypunches to accommodate 8-bit bytes and the larger character set, and the capital for the factories. The whole program cost billions. The software budget for the OS, not counting DOS, I think was somewhere in the neighborhood of \$400 million in 1964

dollars, which would be \$4 billion today roughly.

We recognized that building big systems is different from building programs. There is still an awful lot we do not know about how to build complicated systems and keep them under control. Tom Watson asked me a question at lunch: "You've managed the hardware program. You've managed the software program. What's the difference?" I said, "Well I can't answer that one but I'll think about it." That's where *The Mythical Man Month* came from. As I say in the book, managing software is more like managing hardware than most software people believe and it is less like managing hardware than most hardware managers believe.

On to Chapel Hill

I love to teach, and the atmosphere and the support at Chapel Hill in terms of my colleagues could not have been better.

One of the areas we picked for emphasis when we started the department was interactive three-dimensional computer graphics. I worked for 30 years with protein chemists, building tools to enable them to construct protein structures from crystallographic data. Then the virtual reality concepts came along as a natural extension of 3D interactive computer graphics. We have been doing that for years, looking mostly at serious applications as opposed to entertainment applications.

I would advise someone considering a career in the computer field to look at the intersection between the computer field and biology. The ablest young people today are often opting entirely to go totally into biology instead of computers, and biology offers the same promise today that the computer field did to me in 1953. On the other hand, the key to much biology is information: where it is hidden and how it is processed. For the computer scientist the interaction with biology and biologists is the golden opportunity. That is where the fun is going to be. 

Len Shustek (shustek@computerhistory.org) is the chairman of the Computer History Museum.

^d Frederick P. Brooks, Jr., *The Mythical Man Month*. Addison-Wesley, 1975, 1982, 47.

Viewpoint

When Technologies Manipulate Our Emotions

Implications of the Facebook emotions study.

ON THOSE RARE occasions when an academic research study receives frenzied media attention, it usually indicates the topic has touched on societal fears in some way. On June 17, 2014 the *Proceedings of the National Academy of Sciences* (PNAS) published a paper by researchers at Facebook and Cornell University presenting evidence for widespread emotional contagion in online social networks. The Facebook study manipulated the percentage of positive or negative posts appearing in Facebook user news feeds and reported that their manipulations had the anticipated effects of demonstrating emotional contagion at scale. The study contributes interesting results to an underserved area of research, but it triggered an understandable flurry of concern because of a failure to obtain consent from participants before attempting to influence their emotions. Public concern was also intensified by the fact that most people were unaware that Facebook already filters user news feeds (by necessity due to scale), that the company has such unprecedented reach, and that the manipulation involved something as personal as feelings and emotions.

The lack of awareness regarding information filtering supports an illusion of neutrality regarding technology design—the notion that computer programs are bereft of values or moral intent by default. This and other important issues have received less attention amid the flurry of criticism



pertaining to research ethics (most of the 120 papers Google Scholar identified as citing the study focused on ethics). Some open issues include the lack of transparency given restrictions to data access and the difficulty in systematic replication due to a lack of Facebook-level resources. This Viewpoint takes a different approach by discussing the implications of the study on technology design and the criteria on which technologists should base their design decisions.

The Facebook-Cornell study emerged from an attempt to better understand widespread emotion contagion in social networks. The authors experimentally studied how two filtering algorithms influenced the emotional expressions of a large number of users ($N=689,003$) by manipulating the likelihood of positive or negative posts appearing in the users'

news feeds. They then studied the emotional content of users' status updates, which were ostensibly influenced by the emotional content in their news feeds. Emotion feelings were measured by the inclusion of emotional terms in the posts, computed with the Linguistic Inquiry and Word Count system (LIWC)⁵ that provides psychologically grounded lists of positive and negative emotional terms (amongst other categories). After a week, those in the positivity-reduced condition (for whom the number of positive posts was reduced) used fewer positive (0.1%) and more negative (0.04%) emotional terms compared to a control condition where a similar proportion of posts were reduced at random (that is, without respect to emotional content). In contrast, when negative posts were reduced (negativity-reduced condition), users used more positive (0.06%) and

Emotional Contagion

It should be noted there is some justification to question the central claim that small changes in emotional word use reflect corresponding changes in *felt* emotions. There might be a number of other reasons why people change their emotional expressions aside from emotional contagion, including empathic mimicry, social politics, and compassion. Because the evidence for a contagion effect produced by the study is already statistically small (an effect of .0001 sigma), it might be the case that even if other factors only contribute minimally, it may still be enough to put the claim of emotion contagion in question. If, as the authors state, “[mimicry] may well have been part of the effect” then how small would the resulting effect be were mimicry excluded? For example, there was less than a 0.1% difference in use of positive words in the negativity-reduced condition. Could empathic mimicry be driving 0.05% of this effect? In order to provide sufficient evidence that the data from the Facebook-Cornell study shows that emotions themselves (rather than just emotional expression) can be contagious on Facebook, more work will need to be done to clarify the prevalence of various factors underlying the observed effects. Investigating the various and nuanced motivations behind observed behavior is essential to truly understanding the impact of software designs on users’ lives, yet it is, admittedly, a task not generally well suited to big data studies.

fewer negative (0.07%) emotional terms compared to the control condition. The authors’ interpretation of these findings was that users *felt* more negative and positive emotions in the respective conditions—thus emotion contagion occurred.

If we accept the authors’ conclusions that emotion contagion did in fact occur (see the accompanying sidebar), this leads us to a larger question for technology: Can design ever be emotionally neutral and if not, on what criteria should technologists base design decisions? Beyond the controversy surrounding the way the Facebook study addressed informed consent, it is important not to miss the valuable contribution made by this study. This study contributes to a critical area of modern inquiry: How does digital experience and the design of it affect our emotions? This is significant because one issue neglected in the media discourse is that design, be it of a filter, interface, or algorithm, is arguably never neutral. For example, newspapers use editorial guidelines to filter what information is published and search engines use algorithms to make these decisions. Every design decision must be based on some criteria. As researchers in Values-sensitive Design have made clear, the values and goals of designers and stakeholders will shape the design of any technology.³ Thus, the obligation to understand the impacts of our design decisions, and to be transparent about what influences them, becomes imperative.

If design is known to affect emotions, above and beyond this particular

study on emotion contagion (see Calvo and Peters¹ for a list of examples), how can we study these effects and how should we apply the knowledge gained? If software design is not neutral, how should a software designer make decisions about what is a “good” design—particularly when we are designing interfaces so closely linked to what we care most about: family, friends, and relationships? For instance, if it is in fact *impossible* for Facebook *not* to filter information due to scale, how should the filter criteria be determined? Should filters only ever be randomized and not optimized for the user experience? Or can we look deeper and seek to support greater transparency and user autonomy; what if designers allowed users to make more of these decisions themselves? For example, what if users could set the parameters for their news feed filter or aspects of their search algorithms on their own? Transparency and autonomy seem underexplored opportunities for respecting individual differences and safeguarding against paternalism or misuse.

We also posit that, as we look for criteria upon which to base technology design decisions, we should be turning to the research on psychological well-being—that design decisions should seek to promote (rather than hinder) *thriving* (an area we call Positive Computing⁴). It is important to note that well-being is not defined simply as an increase in positive emotions. Other determinants include empathy, compassion, self-awareness, engagement, autonomy, and connectedness accord-

ing to research in psychology. Negative emotions are also an important component of lasting well-being.² In this view, empathizing with a friend in need or receiving that empathy, may contribute more to one’s well-being than merely positive expression. Clearly, emotional impact represents a rich, nuanced, and complex space of inquiry into which we are still only scratching the surface.

Conclusion

There is still much left to be understood about how our emotional lives play out in digital experience and how the design of systems, interfaces, and interactions shape our emotional experience. By publishing studies like the one mentioned here, companies are helping contribute important knowledge, not just to the academic community, but also to those who care about the psychological impact of technology. We believe the controversy over the Facebook study is a useful reminder of how important it is to uphold ethical guidelines in research, and the important role technology plays in our emotional experience. However, we hope it will encourage rather than deter further research into understanding ourselves better and understanding how we, as computing professionals, can make design decisions that are of optimal benefit to society. ■

References

1. Calvo, R.A. and Peters, D. *Positive Computing: Technology for Wellbeing and Human Potential*. MIT Press, 2014.
2. Fredrickson, B.L. and Losada, M.F. Positive affect and the complex dynamics of human flourishing. *The American Psychologist* 60, 7 (2005), 678.
3. Friedman, B., Kahn, P.H., and Borning, A. Value sensitive design and information systems. In K.E. Himma and H.T. Tavani, Eds., *The Handbook of Information and Computer Ethics*. Wiley, 2008, 69–101.
4. Kramer, A., Guillory, J.E., and Hancock, J.T. Experimental evidence of massive-scale emotional contagion through social networks. In *Proceedings of the National Academy of Sciences* (2014): 201320040.
5. Pennebaker, J.W. *The Development and Psychometric Properties of LIWC2007*. Austin, TX, LIWC, 2007.

Rafael A. Calvo (rafael.calvo@sydney.edu.au) is a professor and the co-director of the Software Engineering Group at the University of Sydney, Australia.

Dorian Peters (dorian.peters@sydney.edu.au) is Creative Leader, Web and Interface Design for Learning, at the University of Sydney, Australia.

Sidney D'Mello (sdmello@nd.edu) is an assistant professor in the Departments of Psychology and Computer Science and Engineering at the University of Notre Dame, Notre Dame, Indiana.

Rafael A. Calvo is supported by Australian Research Council Future Fellowship. Sidney D'Mello is supported by the U.S. National Science Foundation (NSF).

Copyright held by authors.

ACM Books. In-depth. Innovative. Insightful.

The VR Book: Human-Centered Design for Virtual Reality

By Jason Jerald, PhD

The VR Book focuses on human-centered design of virtual reality (VR) experiences. Creating compelling VR experiences is an incredibly complex challenge. When VR is done well, these experiences can be brilliant and pleasurable, but when done badly, they can result in frustration and sickness. While there are many causes of bad VR such as the limitations of technology, much is centered on a lack of understanding of human perception, interaction, design principles, and real users.

This book focuses on human elements of VR, such as how users perceive and intuitively interact with various forms of reality, causes of VR sickness, creating useful and pleasing content, and how to design and iterate upon effective VR applications. It is not just for VR designers; it is for the entire team, as they all should understand the basics of perception, VR sickness, interaction, content creation, and iterative design.

Good VR design requires strong communication between human and machine, indicating what interactions are possible, what is currently occurring, and what is about to occur. A human-centered design principle, like lean methods, is to avoid completely defining the problem at the start and to iterate upon repeated approximations and modifications through rapid tests of ideas with real users. Thus, The VR Book is intended as a foundation for anyone and everyone involved in creating VR experiences including: designers, managers, programmers, artists, psychologists, engineers, students, educators, and user experience professionals.

Paperback ISBN: 9781970001129 Hardcover ISBN: 9781970001150
Ebook ISBN: 9781970001136 DOI: 10.1145/2792790



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

or contact ACM at
ACMbooks-Info@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



Morgan & Claypool Publishers
1210 Fifth Avenue, Suite 250
San Rafael, CA 94901, USA
Phone: +1-415-462-0004
Email: info@morganclaypool.com

Reliability in the face of rapid change.

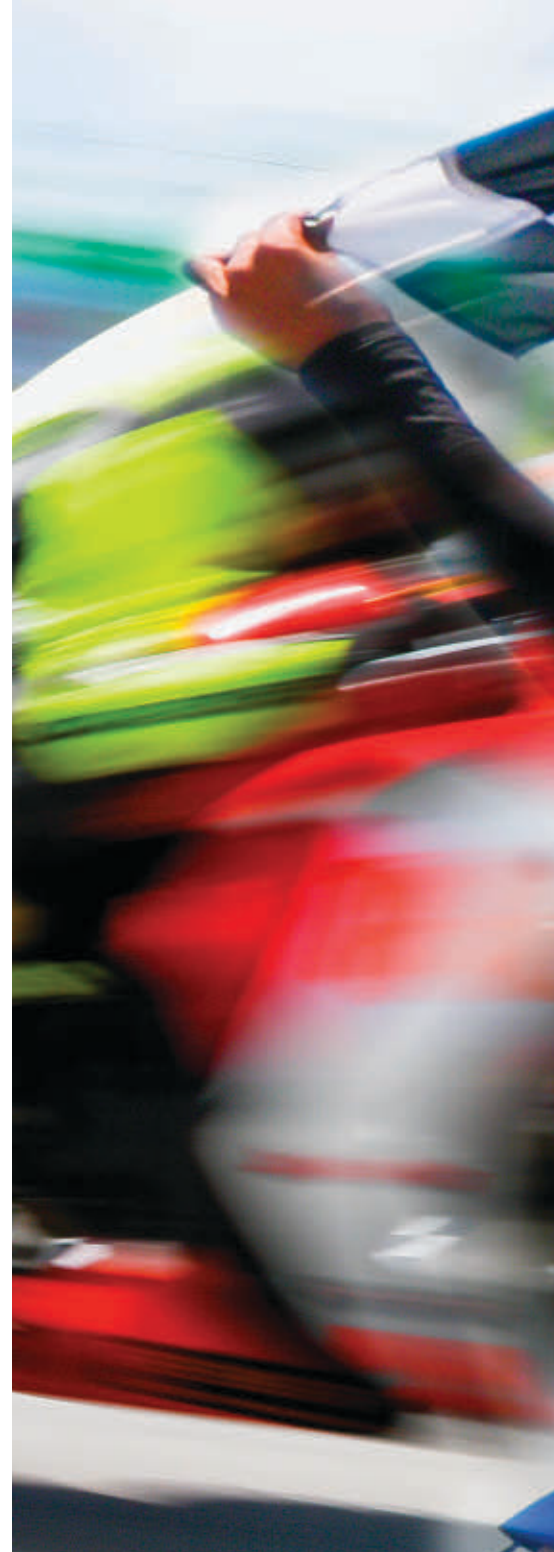
BY BEN MAURER

Fail at Scale

FAILURE IS PART of engineering any large-scale system. One of Facebook's cultural values is embracing failure. This can be seen in the posters hung around the walls of our Menlo Park headquarters: "What Would You Do If You Weren't Afraid?" and "Fortune Favors the Bold."

To keep Facebook reliable in the face of rapid change we study common patterns in failures and build abstractions to address them. These abstractions ensure best practices are applied across our entire infrastructure. To guide our work in building reliability abstractions, we must understand our failures. We do this by building tools to diagnose issues and by creating a culture of reviewing incidents in a way that pushes us to make improvements that prevent future failures.

While every failure has a unique story, many failures boil down to a small number of fundamental root causes.



Individual machine failures. Often an individual machine will run into an isolated failure that does not affect the rest of the infrastructure. For example, maybe a machine's hard drive has failed, or a service on a particular machine has experienced a bug in code, such as memory corruption or a deadlock.

The key to avoiding individual machine failure is automation. Automation works best by combining known failure patterns (such as a hard drive with S.M.A.R.T. errors) with a search



IMAGE FROM SHUTTERSTOCK.COM

for symptoms of an unknown problem (for example, by swapping out servers with unusually slow response times). When automation finds symptoms of an unknown problem, manual investigation can help develop better tools to detect and fix future problems.

Legitimate workload changes. Sometimes Facebook users change their behavior in a way that poses challenges for our infrastructure. During major world events, for example, unique types of workloads may stress

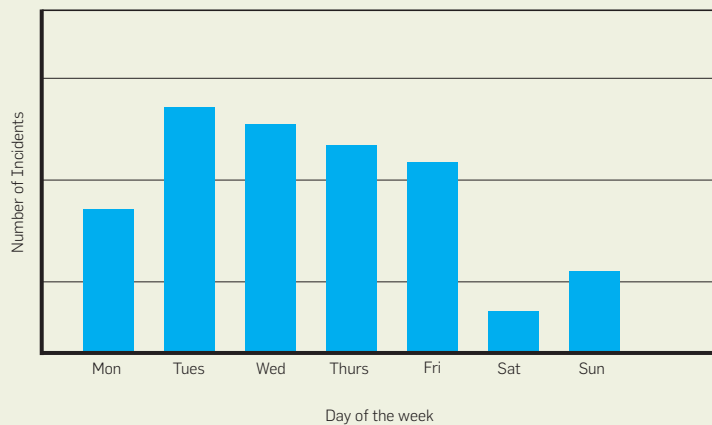
our infrastructure in unusual ways. When Barack Obama won the 2008 U.S. Presidential election, his Facebook page experienced record levels of activity. Climactic plays in major sporting events such as the Super Bowl or World Cup result in an extremely high number of posts. Load testing, including “dark launches” where a feature is activated but not visible to the user, helps ensure new features are able to handle load.

Statistics gathered during such

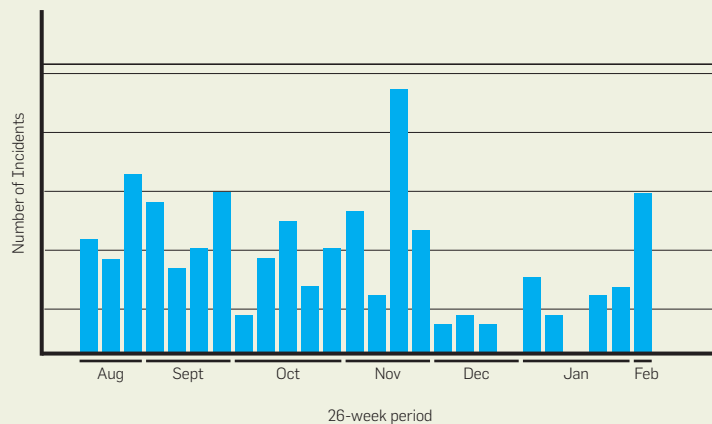
events often provide a unique perspective on a system’s design. Oftentimes, major events cause changes in user behavior (for example, by creating focused activity around a particular object). Data about these changes often points to design decisions that will allow smoother operation in subsequent events.

Human error. Given that Facebook encourages engineers to “Move Fast and Break Things”—another one of the posters that adorns the offices—

Figure 1. SLA violation events.



(a) Incidents across the week



(b) Number of incidents by week

one might expect many errors are caused by humans. Our data suggests that human error is a factor in our failures. Figure 1 includes data from an analysis of the timing of events severe enough to be considered a service-level agreement (SLA) violation. Each violation indicates an instance where our internal reliability goals were not met and caused an alert to be generated. Because our goals are strict most of these incidents are minor and not noticeable to users of the site. Figure 1a shows how incidents happened substantially less on Saturday and Sunday even though traffic to the site remains consistent throughout the week. Figure 1b shows a six-month period during which there were only two weeks with no incidents: the week of Christmas and the week when employees are expected to write peer reviews for each other.

These two data points seem to suggest that when Facebook employees are not actively making changes to infrastructure because they are busy with other things (weekends, holidays, or even performance reviews), the site experiences higher levels of reliability. We believe this is not a result of carelessness on the part of people making changes but rather evidence our infrastructure is largely self-healing in the face of non-human causes of errors such as machine failure.

Three Easy Ways to Cause an Incident

While failures have different root causes, we have found three common pathologies that amplify failures and cause them to become widespread. For each pathology, we have developed preventative measures that mitigate widespread failure.

Rapidly deployed configuration changes. Configuration systems tend to be designed to replicate changes quickly on a global scale. Rapid configuration change is a powerful tool that can let engineers quickly manage the launch of new products or adjust settings. However, rapid configuration change means rapid failure when bad configurations are deployed. We use a number of practices to prevent configuration changes from causing failure.

- *Make everybody use a common configuration system.* Using a common configuration system ensures procedures and tools apply to all types of configuration. At Facebook we have found teams are sometimes tempted to handle configurations in a one-off way. Avoiding these temptations and managing configurations in a unified way has made the configuration system a leveraged way to make the site more reliable.

- *Statically validate configuration changes.* Many configuration systems allow loosely typed configuration, such as JSON structures. These types of configurations make it easy for an engineer to mistype the name of a field, use a string where an integer was required, or make other simple errors. These kinds of straightforward errors are best caught using static validation. A structured format (for example, at Facebook we use Thrift)⁴ can provide the most basic validation. It is not unreasonable, however, to write programmatic validation to validate more detailed requirements.

- *Run a canary.* First deploying your configuration to a small scope of your service can prevent a change from being disastrous. A canary can take multiple forms. The most obvious is an A/B test, such as launching a new configuration to only 1% of users. Multiple A/B tests can be run concurrently, and you can use data over time to track metrics.

For reliability purposes, however, A/B tests do not satisfy all of our needs. A change deployed to a small number of users, but causing implicated servers to crash or run out of memory, will obviously create impact that goes beyond the limited users in the test. A/B tests are also time consuming. Engineers often wish to push out minor changes without the use of an A/B test. For this

reason, Facebook infrastructure automatically tests out new configurations on a small set of servers. For example, if we wish to deploy a new A/B test to 1% of users, we will first deploy the test to 1% of the users that hit a small number of servers. We monitor these servers for a short amount of time to ensure they do not crash or have other highly visible problems. This mechanism provides a basic “sanity check” on all changes to ensure they do not cause widespread failure.

► *Hold on to good configurations.* Facebook’s configuration system is designed to retain good configurations in the face of failures when updating those configurations. Developers tend naturally to create configuration systems that will crash when they receive updated configurations that are invalid. We prefer systems that retain old configurations in these types of situations and raise alerts to the system operator the configuration failed to update. Running with a stale configuration is generally preferable to returning errors to users.

► *Make it easy to revert.* Sometimes, despite all best efforts, a bad configuration is deployed. Quickly finding and reverting the change is key to resolving this type of issue. Our configuration system is backed by version control, making it easy to revert changes.

Hard dependences on core services.

Developers tend to assume that core services—such as configuration management, service discovery, or storage systems—never fail. Even brief failures in these core services, however, can turn into large-scale incidents.

► *Cache data from core services.* Hard dependencies on these types of services are often not necessary. The data these services return can be cached in a way that allows for the majority of services to continue operating during a brief outage of one of these systems.

► *Provide hardened APIs to use core services.* Core services are best complemented by common libraries that follow best practices when using these core services. For example, the libraries might provide good APIs for managing the cache or good failure handling.

► *Run fire drills.* You might think you are able to survive an outage of a core service, but you never know until you



To guide our work in building reliability abstractions, we must understand our failures. We do this by building tools to diagnose issues and by creating a culture of reviewing incidents in a way that pushes us to make improvements that prevent future failures.



try. For these types of outages we have had to develop systems for fire drills ranging from fault injection applied to a single server to manually triggered outages of entire data centers.

Increased latency and resource exhaustion. Some failures result in services having increased latency to clients. This increase in latency could be small (for example, think of a human configuration error that results in increased CPU usage that is still within the service’s capacity), or it could be nearly infinite (a service where the threads serving responses have deadlocked). While small amounts of additional latency can be easily handled by Facebook’s infrastructure, large amounts of latency lead to cascading failures. Almost all services have a limit to the number of outstanding requests. This limit could be due to a limited number of threads in a thread-per-request service, or it could be due to limited memory in an event-based service. If a service experiences large amounts of extra latency, then the services that call it will exhaust their resources. This failure can be propagated through many layers of services, causing widespread failure.

Resource exhaustion is a particularly damaging mode of failure because it allows the failure of a service used by a subset of requests to cause the failure of all requests. For example, imagine a service calls a new experimental service that is only launched to 1% of users. Normally requests to this experimental service take 1 millisecond, but due to a failure in the new service the requests take 1 second. Requests for the 1% of users using this new service might consume so many threads that requests for the other 99% of users are unable to run.

We have found a number of techniques that can avoid this type of build-up with a low false positive rate.

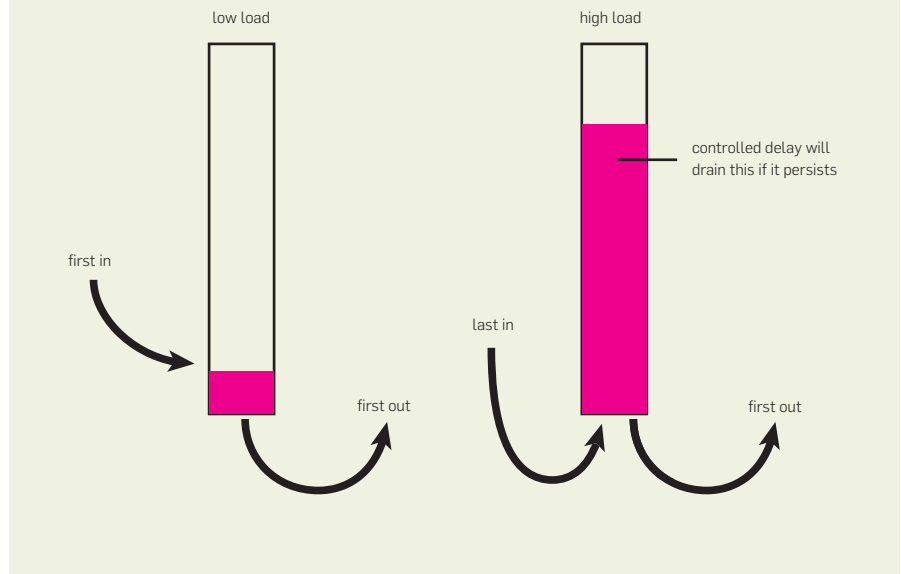
► *Controlled delay.* In analyzing past incidents involving latency, we found many of our worst incidents involved large numbers of requests sitting in queues awaiting processing. The services in question had a resource limit (such as a number of active threads or memory) and would buffer requests in order to keep usage below the limit. Because the services were unable to keep up with the rate of incoming requests, the queue would grow larger and larger

until it hit an application-defined limit. To address this situation, we wanted to limit the size of the queue without impacting reliability during normal operations. We studied the research on bufferbloat as our problems seemed similar—the need to queue for reliability without causing excessive latency during congestion. We experimented with a variant of the CoDel¹ (controlled delay) algorithm:

```
onNewRequest(req, queue):
  if (queue.lastEmptyTime() <
      (now - N seconds)) {
    timeout = M ms
  } else {
    timeout = N seconds;
  }
  queue.enqueue(req, timeout)
```

In this algorithm, if the queue has not been empty for the last N milliseconds, then the amount of time spent in the queue is limited to M milliseconds. If the service has been able to empty the queue within the last N milliseconds, then the time spent in the queue is limited to N milliseconds. This algorithm prevents a standing queue (because the `lastEmptyTime` will be in the distant past, causing an M -ms queuing timeout) while allowing short bursts of queuing for reliability purposes. While it might seem counterintuitive to have requests with such short timeouts, this process allows requests to be quickly discarded rather than build up when the system is not able to keep up with the rate of incoming requests. A short timeout ensures the server always accepts just a little bit more work than it can actually handle so it never goes idle.

Figure 2. LIFO (left) and Adaptive LIFO with CoDel.

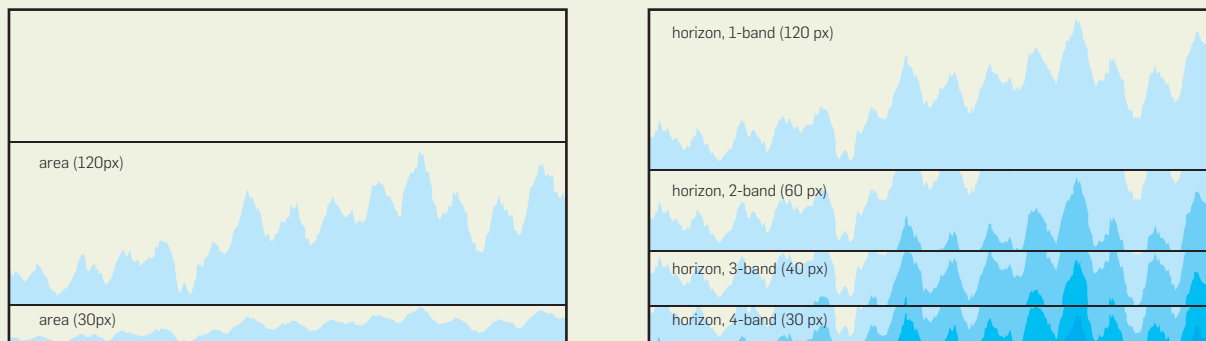


An attractive property of this algorithm is the values of M and N tend not to need tuning. Other methods of solving the problem of standing queues, such as setting a limit on the number of items in the queue or setting a timeout for the queue, have required tuning on a per-service basis. We have found a value of five milliseconds for M and 100ms for N tends to work well across a wide set of use cases. Facebook's open source Wangle library⁵ provides an implementation of this algorithm, which is used by our Thrift4 framework.

► *Adaptive LIFO (last-in, first-out).* Most services process queues in FIFO (first-in first-out) order. During periods of high queuing, however, the

first-in request has often been sitting around for so long the user may have aborted the action that generated the request. Processing the first-in request first expends resources on a request that is less likely to benefit a user than a request that has just arrived. Our services process requests using adaptive LIFO. During normal operating conditions, requests are processed in FIFO order, but when a queue is starting to form, the server switches to LIFO mode. Adaptive LIFO and CoDel play nicely together, as shown in Figure 2. CoDel sets short timeouts, preventing long queues from building up, and adaptive LIFO places new requests at the front of the queue, maximizing the chance they will meet the deadline set

Figure 3. Area charts (left) and horizontal charts (right).



by CoDel. HHVM,³ Facebook's PHP runtime, includes an implementation of the Adaptive LIFO algorithm.

► **Concurrency control.** Both CoDel and adaptive LIFO operate on the server side. The server is often the best place to implement latency-preventing measures—a server tends to serve a large number of clients and often has more information than its clients possess. Some failures are so severe, however, that server-side controls are not able to kick in. For this reason, we have implemented a stopgap measure in clients. Each client keeps track of the number of outstanding outbound requests on a per-service basis. When new requests are sent, if the number of outstanding requests to that service exceeds a configurable number, the request is immediately marked as an error. This mechanism prevents a single service from monopolizing all its client's resources.

Tools That Help Diagnose Failures

Despite the best preventative measures, some failures will always occur. During outages the right tools can quickly lead to the root cause, minimizing the duration of the failure.

High-density dashboards with Cubism. When handling an incident, it is important to have quick access to information. Good dashboards allow engineers to quickly assess the types of metrics that might be abnormal and then use this information to hypothesize a root cause. We found, however, that our dashboards grew so large it was difficult to navigate them quickly, and that charts shown on those dashboards had too many lines to read at a glance.

To address this, we built our top-level dashboards using Cubism,² a framework for creating horizon charts—line charts that use color to encode information more densely, allowing for easy comparison of multiple similar data series. For example, we use Cubism to compare metrics between different data centers. Our tooling around Cubism allows for easy keyboard navigation so engineers can view multiple metrics quickly. Figure 3 shows the same dataset at various heights using area charts and horizon charts. In the area chart version, the 30-pixel version is

difficult to read. On the other hand, the horizon chart makes it extremely easy to find the peak value, even at a height of 30 pixels.

What just changed? Since one of the top causes of failure is human error, one of the most effective ways of debugging failures is to look for what humans have changed recently. We collect information about recent changes ranging from configuration changes to deployments of new software in a tool called OpsStream. However, we have found over time this data source has become extremely noisy. With thousands of engineers making changes, there are often too many to evaluate during an incident.

To solve this problem, our tools attempt to correlate failures with relevant changes. For example, when an exception is thrown, in addition to outputting the stack trace, we output any configuration settings read by that request that have had their values changed recently. Often, the cause of an issue that generates many stack traces is one of these configuration values. We can then quickly respond to the issue—for example, by reverting the configuration and involving the engineer who made the change.

Learning from Failure

After failures happen, our incident-review process helps us learn from these incidents.

The goal of the incident-review process is not to assign blame. Nobody has been fired because an incident he or she caused came under review. The goal of the review is to understand what happened, remediate situations that allowed the incident to happen, and put safety mechanisms in place to reduce the impact of future incidents.

A methodology for reviewing incidents. Facebook has developed a methodology called DERP (for detection, escalation, remediation, and prevention) to aid in productive incident reviews.

► **Detection.** How was the issue detected—alarms, dashboards, user reports?

► **Escalation.** Did the right people get involved quickly? Could these people have been brought in via alarms rather than manually?

► **Remediation.** What steps were tak-

en to fix the issue? Can these steps be automated?

► **Prevention.** What improvements could remove the risk of this type of failure happening again? How could you have failed gracefully, or failed faster to reduce the impact of this failure?

DERP helps analyze every step of the incident at hand. With the aid of this analysis, even if you cannot prevent this type of incident from happening again, you will at least be able to recover faster the next time.

Move Fast by Breaking Fewer Things

A “move-fast” mentality does not have to be at odds with reliability. To make these philosophies compatible, Facebook's infrastructure provides safety valves: our configuration system protects against rapid deployment of bad configurations; our core services provide clients with hardened APIs to protect against failure; and our core libraries prevent resource exhaustion in the face of latency. To deal with the inevitable issues that slip through the cracks, we build easy-to-use dashboards and tools to help find recent changes that might cause the issues under investigation. Most importantly, after an incident we use lessons learned to make our infrastructure more reliable. ■

Related articles on queue.acm.org

Automating Software Failure Reporting

Brendan Murphy

<http://queue.acm.org/detail.cfm?id=1036498>

Scale Failure

George Neville-Neil

<http://queue.acm.org/detail.cfm?id=2147781>

Improving Performance on the Internet

Tom Leighton

<http://queue.acm.org/detail.cfm?id=1466449>

References

1. CoDel (controlled delay) algorithm; <http://queue.acm.org/detail.cfm?id=2209336>.
2. Cubism; <https://square.github.io/cubism/>.
3. HipHop Virtual Machine (HHVM); <http://bit.ly/1Qw68bz>.
4. Thrift framework; <https://github.com/facebook/fbthrift>.
5. Wangle library; <https://github.com/facebook/wangle/blob/master/wangle/concurrent/CoDel.cpp>.

Ben Maurer is the tech lead of the Web Foundation team at Facebook, which is responsible for the overall performance and reliability of Facebook's user-facing products. Before Facebook, he co-founded reCAPTCHA with Luis von Ahn.

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

Probabilistic algorithms are all around us. Not only are they acceptable, some programmers actually seek out chances to use them.

BY TYLER MCMULLEN

It Probably Works

PROBABILISTIC ALGORITHMS EXIST to solve problems that are either impossible or unrealistic (too expensive or too time consuming, and so on) to solve precisely. In an ideal world, you would never actually need to use probabilistic algorithms. To programmers who are not familiar with them, the idea can be positively nerve-racking: “How do I know it will actually work? What if it is inexplicably wrong? How can I debug it? Maybe we should just punt on this problem, or buy a whole lot more servers ...”

However, to those who either deeply understand probability theory or at least have used and observed the behavior of probabilistic algorithms in large-scale production environments, these algorithms are not

only acceptable, but it is also worth seeking out opportunities to use them. This is because they can help solve problems and create systems that are less expensive, more predictable, and can do things that could not be done otherwise.

The world is probabilistic—or at least it is way too complex for us to model 100% accurately. Networks, and especially the Internet, are also probabilistic. Whether or not a particular packet will get where it needs to go is not something we can have complete knowledge of. Knowing which paths it will take through the many networks it will traverse in getting from one side of the world to the other is even less certain. Systems that run across asynchronous/lossy networks are necessarily probabilistic. Like it or not, probability is all around us.

Uncertainty is the core of the problem that uninitiated developers have with probabilistic algorithms, leading them to believe the algorithms may work poorly or be impossible to debug. That idea, however, is not correct. Probabilistic algorithms incorporate an element of randomness (they are also referred to as *randomized algorithms*), which is quantifiable. It can be analyzed, allowing strong predictions about the algorithm’s behavior to be made. There is, in fact, very little uncertainty about how the algorithms will behave.

This article addresses two types of probabilistic algorithms: those that explicitly introduce randomness through a `rand()` call, and those that convert input data into a uniform distribution to achieve a similar effect. I will discuss specific algorithms that fall into these classes, including LogLog (and its better-known successor HyperLogLog) and Bimodal Multicast, and the problems they attempt to solve.

In a third type of probabilistic algorithm, randomness is intrinsic to the data. This applies to problems such as GPS tracking, self-driving cars, sensor networks, and missile-guidance systems. For example, sensor networks often try to make a statement about



the network as a whole despite having only partial data. In the case of self-driving cars and missile-guidance systems, the data is in continuous flux. By the time a program has processed the data, the current state has changed. So it is already wrong and that must be taken into account. This article does not discuss this type of algorithm, but for more information on the set of problems in this category and the algorithms to solve them, it is helpful to read up on Kalman filters¹¹ and estimation algorithms in general.

Count-Distinct

The count-distinct problem involves counting the number of unique items in a stream that may contain duplicates. More formally: find the cardinality of a multiset. Many problems fall into this category. For example: How

many unique IPs have connected to a server over the past hour? How many different words are in a huge corpus of text? How many different users visited a popular Web site in a day? Or even, how many unique URLs have been requested through an HTTP proxy?

The naive solution to this problem is easy. You could easily have written the code in your head before getting to the end of the previous sentence. Here is some Python code that illustrates the solution:

```
...
def count_distinct(stream):
    seen = set()
    for item in stream:
        seen.add(item)
    return len(seen)
...
```

As with many problems, the difficulties with count-distinct come

with scale. It may be easy and cheap to count a thousand or even a million unique users, IPs, URLs, or words, but how about 100 million? Still not too bad. How about 100 million per server across thousands of servers? Now it is starting to get interesting.

The problem now is: perform a count-distinct across 1,000 servers, which could be as many as 100 million unique items each, and find the cardinality of the union of their result. In other words: distributed count-distinct. Let's consider a naive solution.

Each server could keep a set of "seen" items, just as in the previous code example. When each finishes its individual count-distinct run, it sends the entire contents of that "seen" set to a central server, which can then take the union of all the sets and return the cardinality of the combined set.

```
...
def combined_cardinality(seen_sets):
    combined = set()
    for seen in seen_sets:
        combined |= seen
    return len(combined)
...
```

Again, the naive solution is delightfully simple and obvious, but it comes at a steep cost. If each server is seeing as many as 100 million unique items, then the size of that “seen” list is going to be significant. In the case of URLs, the average length is around 76 characters. This would put the list at approximately 7.6GB of data per server. Even if each of the URLs could be hashed down to a 64-bit integer, the size would be 800MB. That is much more manageable, but what if there are thousands of servers? Each of those servers sends its “seen” list to a central server. In the case of just 1,000 servers, that is 800GB of data to dump into the `combined_cardinality` function.

If you need to do this distributed count-distinct often, then you are either going to need to install one of those “big data” systems (and hire a team to run it) or find a different solution.

Enter HyperLogLog. This algorithm takes a probabilistic approach to the count-distinct problem. Its core is based on two observations: first, the probability of any particular bit being set in the binary representation of a random number is 50%; second, the probability of two independent events, A and B, both occurring is $P(A) \cdot P(B)$. Thus, if the probability that any one specific bit is set in that random number is 50%, then the probability any two specific bits are set is 25%, and the probability any *three* specific bits are set is 12.5%. You get the picture.

The other bit of Probability Theory 101 to remember is the expected number of trials until an event occurs is $1 / P(\text{event})$. Therefore, if $P(\text{one specific bit set})$ is 50%, then its expected number of trials is two. For two specific bits, the expectation is four; for three, eight; and so on.

Input values are generally not uniformly distributed random numbers, so you need a way of transforming the input values into values that resemble a uniform distribution—in other words, a hash function. A word



Whether or not a particular packet will get where it needs to go is not something we can have complete knowledge of. Knowing which paths it will take through the many networks it will traverse in getting from one side of the world to the other is even less certain.



of warning: in some applications of hash functions the distribution is not critical to the correctness of the system. HyperLogLog is very sensitive to this, however. If certain bits of the hash function output are significantly more likely than others to be set or unset, it completely throws off the math that underlies the algorithm. (I have had good success with MurmurHash3.¹³ To see how your preferred hash function stacks up, take a look at SMHasher,¹⁵ which does a great job of finding flaws in hash functions.)

The first step, then, is to take the input item and hash it to a set of bits. Then count the number of leading bits that are set and keep a record of the *maximum* number of leading bits that have been set so far. This would provide a rough estimate of the number of unique items processed. The probability of seeing one leading bit set is 50%, so you would “expect” the number of unique items is two. With three leading bits set, then, on average, you would see eight unique items.

The way to improve the estimate—and the unique idea behind HyperLogLog—is to divide the incoming items into a series of “buckets” based on their trailing bits. Remember the maximum leading number of set bits for each of the buckets: by dividing the incoming items into buckets like this, you can develop a much more accurate estimate of the total cardinality. If you have m buckets and n unique items, then on average each bucket will have seen n/m unique items. Therefore, taking the mean of those buckets provides a reasonable estimate of $\log_2(n/m)$, from which you can easily generate an estimate.

Further, HyperLogLog has the excellent property of allowing you to take separate HyperLogLog instances of the same number of buckets, as well as the maximum of each of their corresponding buckets, and end up with a HyperLogLog that is the *union* of those two previous HyperLogLogs.

That is a rough sketch of how the HyperLogLog algorithm works. The primary change HyperLogLog introduces is the use of a harmonic mean instead of an arithmetic mean. (For further information, read the original LogLog⁶ and HyperLogLog⁸ papers.)

Consider the naive implementation of the distributed count-distinct prob-

lem. Assume you have 1,000 servers and 100 million unique items per server. During every run of the algorithm, the central server will need to process 800GB of data. This also implies that 800GB of data will need to traverse the network. How does HyperLogLog change this?

According to the analysis done by the authors of the original paper, you can expect accuracy of about 2% while using approximately 1.5KB of space for the buckets. Each server keeps its own 1.5KB HyperLogLog, then sends it to the central server. For 1,000 servers, the central server is processing about 1.5MB of data per run of the distributed count-distinct. In other words, you are processing only about 0.0002% of the data you were processing for the naive solution.

This completely changes the economics of the problem. Suddenly you could run many of these, and run them more frequently. You would not need a big data system; you would not even need more than one server—all for the cost of a 2% skew in your estimates.

Nearest-Neighbor Search

The nearest-neighbor search problem involves finding items that are similar within a set—for example, books, music, products, photos, and videos. This is actually a twofold problem: What does *similar* mean, and how do you find items that are similar to each other?

The first half of the problem is known as *feature extraction*. It defines what is being compared about the items. For example, if you were to compare two books character by character, then the character and its location would be your features. In practice, that is not terribly useful. Instead, you might choose to extract the relative count of each unique word and use that as your set of features for the book. Imagine the following quote is a “book:”

One is still what one is going to cease to be and already what one is going to become. One lives one's death, one dies one's life.

—Jean-Paul Sartre

Extracting the counts of each word from this “book” would result in a “bag of words,” visualized in this way:

```
{ already: 1, and: 1, be: 1, become: 1,
  cease: 1, death: 1, die: 1, going: 2, is:
  3, life: 1, lives: 1, one: 7, still: 1, to: 3,
  what: 2 }
```

That is just one way of doing feature extraction. You could choose to extract pairs of words instead:

```
{ already_what: 1, and_already: 1,
  be_and: 1, become_one: 1, cease_to: 1,
  death_one: 1, ... }
```

In all cases, however, the output of feature extraction can be represented as a *vector*. A book with 10,000 unique words could be represented as one vector in a 10,000-dimensional space.

That might seem odd, but representing the features this way allows for some nifty maneuvers. For example, if you think of each book as a single point in an n -dimensional space, you could measure similarity by calculating the Euclidean distance⁷ between them. You could also measure the angle between the vectors. Cosine similarity³ works well for that technique.

There are many techniques for measuring similarity, but they all fall prey to an insidious problem: *the curse of dimensionality*.⁴ Succinctly, the curse of dimensionality is a group of phenomena that occurs when attempting to work with data in a very high-dimensional space due to the volume of space increasing exponentially with each dimension. The problem is the amount of work necessary to calculate the Euclidean distance or the cosine similarity of two vectors grows in relation to the number of dimensions. Doing those calculations on vectors with many dimensions is very time consuming.

Enter another probabilistic algorithm: LSH (locality-sensitive hashing). The idea behind LSH is basically the opposite of traditional hashing. In traditional hashing, two input values that are similar but not the same need to have very different outputs. In LSH, two inputs that are similar should have similar outputs, but in a much, much smaller amount of data. In practice, I have used LSH to convert million-dimensional vectors to 512-bit hashes, and it worked well.

Clearly, if high-dimensional vectors are being shrunk down to a small “signature,” then data will be lost. That is where the probabilistic part comes in.

There are a number of different ways to generate an LSH. The following

example is one of the simpler ways.

Let's say we have a corpus of book texts. We have already extracted a vocabulary with 100,000 entries. Thus, the feature vectors for each of these books will have 100,000 dimensions. Let's also say we have decided to make a 256-bit LSH function. Now we generate 256 completely random 100,000-dimension vectors. These random vectors are the basis of our LSH function.

The LSH function computes the dot product of the input vector and each of the random vectors that were generated ahead of time. For each random vector, if the dot product is positive, we record a 1 bit. If it is negative, we record a 0 bit. We end up with a 256-bit output hash.

The best way to think about this is the random vectors effectively “divide” the vector space. By answering whether the input vector is greater than or less than each of them, we can approximate its “angle” in that space. Therefore, what this LSH function actually does is produce a hash that can be used to approximate the cosine-similarity between any two of the hashes. The cosine-similarity of two books will be proportional to the Hamming distance between the hashes of them.

Reliable Broadcast

Reliably broadcasting messages among a large number of peers across a high-latency lossy network (such as the Internet) is a difficult problem. It is, in fact, the exact problem that we had to solve at Fastly in order to create our Instant Purging system. Normally when faced with this problem, engineers will attempt to solve it with a centralized, single source of truth. That system then either pushes updates to all servers or is pulled from by all servers. Relying on that single source of truth, however, also makes it a single source of failure. If that centralized system goes away, or is inaccessible to all or a subset of the servers, then it is game over for the system's reliability.

Are there alternative solutions? At first glance, this problem appears to be one that could be solved with *atomic broadcast*.⁵ One of the tenets of atomic broadcast, however, is the property of total order. In other words, if you have two messages, A and B, every server always sees them in the same

order. While that property can be useful to some applications, it introduces a *head-of-line blocking* problem.¹⁰ If message A comes before message B, but message A is delayed getting to a server, then every server must wait to receive B until that one server receives A. In our particular application, the order is irrelevant, and the potential additional latency caused by head-of-line blocking is unacceptable.

We kept looking for alternatives and eventually decided on an algorithm called Bimodal Multicast, a multiphase protocol that reliably and probabilistically distributes messages over a network of servers. The first phase consists of unreliably broadcasting a message from one peer to all others. The mechanism for that broadcast is not terribly important—you could use IP multicast if it is available, for example. What is important is the messages get out to all servers as quickly as possible, even if they have a chance of being lost along the way.

The second phase of this algorithm is a *gossip protocol*⁹ used for *anti-entropy*. Every so often each server independently chooses at random another server in the network. It sends that chosen server a digest of its current state, essentially a summarized version of everything that server has seen until this point. The specific format of that digest is not a dictated part of the protocol, but the idea is to send the summary in such a way that when the server receives the digest it can determine which messages it has missed. It will then respond with the list of messages it has not yet received, and the gossip-initiating server will resend those messages.

This system is probabilistic in that servers choose which other servers to gossip with at random. It is possible the server they choose to gossip with in some round may be missing the same messages they are. In that case, they will not immediately recover the missing messages. Even though the way missing messages propagate through the system is random, it is well understood and follows a classic “epidemic” pattern. The speed and probability with which a lost message will make its way through the network can be tuned by changing the frequency of gossip as well as the length of time that servers remember old messages.

The version of this system that is at the core of our purging system is tuned such that the probability of losing messages permanently is astronomically unlikely—a likelihood of about $10^{-312\%}$. At the current rate of purges, we would expect to lose a message as a result of the randomness in the algorithm once every 3×10^{300} years. In other words, “with high probability” is entirely sufficient, as long as you know what that probability is.

Many problems that are difficult to solve precisely become quite feasible with probabilistic algorithms. Consistent hashing² is often applied in load balancing and distributed databases. The Mincut algorithm¹² can be applied to problems in network routing. Bloom filters¹ exist in many databases and network applications. Quicksort¹⁴ is practically the standard sort algorithm in use today. Even basic run-of-the-mill hash tables are actually probabilistic. A programmer’s day-to-day work is filled with probabilistic algorithms and data structures. They tend not to be noticed because they work.

Indeed, probabilistic algorithms are all around us. They can help solve hard problems without resorting to racks upon racks of machines. They can help solve problems that are otherwise practically unsolvable. Probabilistic algorithms should be at the forefront of the minds of those designing and building large systems. C

Related articles on queue.acm.org

Metamorphosis: the Coming Transformation of Translational Systems Biology
Samantha Kleinberg and Bud Mishra
<http://queue.acm.org/detail.cfm?id=1629775>

AI in Computer Games
Alexander Nareyek
<http://queue.acm.org/detail.cfm?id=971593>

Hazy: Making it Easier to Build and Maintain Big-data Analytics
Arun Kumar, Feng Niu, and Christopher Ré,
<http://queue.acm.org/detail.cfm?id=2431055>

References

1. Bloom filters. Wikipedia; https://en.wikipedia.org/wiki/Bloom_filter.
2. Consistent hashing. Wikipedia; https://en.wikipedia.org/wiki/Consistent_hashing.
3. Cosine similarity. Wikipedia; https://en.wikipedia.org/wiki/Cosine_similarity.
4. Curse of dimensionality. Wikipedia; https://en.wikipedia.org/wiki/Curse_of_dimensionality.
5. Defago, X., Schiper, A., Urban, P. Total order broadcast and multicast algorithms: Taxonomy and survey. *ACM Computing Surveys* 36, 4 (2004), 372–421; <http://dl.acm.org/citation.cfm?doid=1041680.1041682>.

6. Durand, M., Flajolet, P. LogLog counting of large cardinalities. *European Symposium on Algorithms* (2003); <http://www.ic.unicamp.br/~celio/peer2peer/math/bitmap-algorithms/durand03loglog.pdf>.
7. Euclidean distance. Wikipedia; https://en.wikipedia.org/wiki/Euclidean_distance.
8. Flajolet, P., Fusy, E., Gandouet, O. and Meunier, F. HyperLogLog: The analysis of a near-optimal cardinality estimation algorithm. *Conference on Analysis of Algorithms* (2007); <http://algo.inria.fr/flajolet/Publications/FlFuGaMe07.pdf>.
9. Gossip protocol. Wikipedia; https://en.wikipedia.org/wiki/Gossip_protocol.
10. Head-of-line blocking problem. Wikipedia; https://en.wikipedia.org/wiki/Head-of-line_blocking.
11. Kalman, R.E. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering* 82 (series D), 1960, 35–45; <http://www.cs.unc.edu/~welch/kalman/kalmanPaper.html>.
12. Mincut algorithm. Wikipedia; https://en.wikipedia.org/wiki/Karger%27s_algorithm.
13. MurmurHash3; <https://code.google.com/p/smhasher/wiki/MurmurHash3>.
14. Quicksort. Wikipedia; <https://en.wikipedia.org/wiki/Quicksort>.
15. SMHasher; <https://code.google.com/p/smhasher/>.

Further Reading

These articles provide a more thorough explanation of the techniques for generating locality-sensitive hashes:

Andoni, A. and Indyk, P.
2008. Near-optimal hashing algorithms approximate nearest neighbor in high dimensions. *Commun. ACM* 51, 1 (2008), 117–122; <http://people.csail.mit.edu/indyk/p117-andoni.pdf>.

Andoni, A., Indyk, P. Patrascu, M.
On the optimality of the dimensionality reduction method. *Foundations of Computer Science*, 2006, 449–458; <http://people.csail.mit.edu/mip/papers/eps2n/eps2n.pdf>

Andoni, A., Indyk, P. Nguyen, H.L., Razenshteyn, I.
Beyond locality-sensitive hashing, 2013; <http://arxiv.org/pdf/1306.1547.pdf>.

Datar, M., Indyk, P., Immorlica, N., Mirrokni, V.S.
Locality-sensitive hashing scheme based on p-stable distributions, 2004; <http://www.cs.princeton.edu/courses/archive/spring05/cos598E/bib/p253-datar.pdf>.

Gionis, A., Indyk, O., Motwani, R.
Similarity search in high dimensions via hashing. *Proceedings of the 25th International Conference on Very Large Databases*, 1999; 518–529; <http://www.cs.princeton.edu/courses/archive/spring13/cos598C/Gionis.pdf>.

Indyk, P., Motwani, R.
Approximate nearest neighbors: Toward removing the curse of dimensionality. *Proceedings of STOC98*, 604–613; <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.249&rep=rep1&type=pdf>.

Tyler McMullen (<https://twitter.com/tbmcmullen>) is CTO at Fastly, responsible for the system architecture and the company’s technology vision. McMullen built the first versions of Fastly’s Instant Purging system, API, and real-time analytics.

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



We may be on the cusp of a new revolution in Web development.

BY TAYLOR SAVAGE

Componentizing the Web

THERE IS NO task in software engineering today quite as herculean as Web development.

A typical specification for a Web application might read: The app must work across a wide variety of browsers. It must run animations at 60fps. It must be immediately responsive to touch. It must conform to

a specific set of design principles and specs. It must work on just about every screen size imaginable, from TVs and 30-inch monitors to mobile phones and watch faces. It must be well engineered and maintainable in the long term.

With the latest Web technologies, this list grows: the Web application must work offline. It must be able to send push notifications. It must sync in the background.

Of course, not all new Web projects must meet this complete set of requirements—some may be more full-fledged single-page apps, others more publishing- or e-commerce-focused—but this diversity of applications makes the Web developer's job even more difficult.

The real challenge arises when the expectations of the Web application meet the realities of the Web platform. The raw materials that are available to build Web applications have not remotely kept up. The building blocks of HTML—such as “<div>,” “<p>,” “<h1>,” and “”—are great for doc-

ument markup, but they are not sufficient to build today's complex, app-like interfaces and websites.

As a result, the Web development community has evolved a vast array of frameworks to tackle the problem of building sane interfaces out of the basic elements the platform provides. The scope and diversity of available Web frameworks is immense—a thriving ecosystem that only a platform as powerful and flexible as the Web could support. As of this writing, more than 60 frameworks are listed on TodoMVC (Figure 1),¹⁰ a showcase for framework usage. This flourishing framework ecosystem is truly amazing.

Or is it?

Though many Web frameworks are incredible feats of engineering, are extraordinarily popular, and have large ecosystems in their own right, there are a few key problems with the multi-framework model that limit Web developer productivity.

Frameworks are wide-reaching, hard dependencies. The inherently

Figure 1. Partial list of frameworks showcased in TodoMVC.

...
 agularjs
 ariatemplates
 atmajs
 backbone
 backbone_marionette
 batman
 canjs
 chaplin-brunch
 closure
 componentjs
 cujo
 derby
 dijon
 dojo
 duel
 durandal
 emberjs
 enyo_backbone
 epitome
 exoskeleton
 extjs_deftjs
 firebase-
 angular
 flight
 foam
 gwt
 jquery
 kendo
 knockback
 knockoutjs
 lavaca
 maria
 meteor
 mithril
 montage
 mozart
 ...

global nature of HTML, CSS, and JavaScript is one of many factors that have driven frameworks to be all-encompassing factories, rather than pay-as-you-go toolkits. Choosing a Web framework is typically the first major technological decision in a new Web project. Because frameworks tend to be overarching, almost every line of code or item of markup must be written with the framework in mind. Angular, for example, provides complex view management and routing, dependency injection, internationalization and accessibility features, low-level animation support, and more. Meteor provides entire frontend and backend stacks, from a UI library down to database drivers. These are incredibly full-featured application platforms in and of themselves, but they lead to lock-in from day one and are extremely challenging to migrate off of. If you choose to switch frameworks, you will probably be starting from scratch.

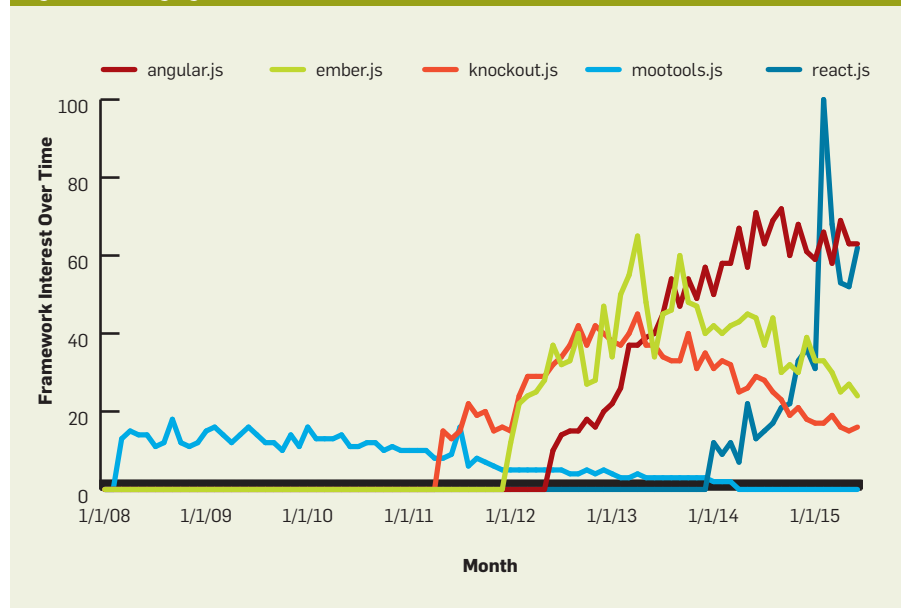
Fashionability. Web frameworks come and go like seasons. Figure 2 shows the rise and fall in popularity of five JavaScript libraries, according to Google Trends. The Web development community is constantly craving the next new thing, and rightly so—Web application requirements and device and browser capabilities are evolving so rapidly the tools must evolve rapidly to keep up. Unfortunately this means yesterday's hot

new framework might be today's old news, and the massive investment a company makes in a technology can quickly become out of date. To stay relevant in the job market, Web developers must keep up with the latest technologies, running on an ever-accelerating treadmill of Hacker News framework announcements, tutorials, and starter kits.

Lack of interoperability. A sane component model is critical to scalable interface development. Frameworks typically define their own component models for organizing and rendering interfaces. Angular has directives, and React and Ember each have their own notion of "Component." Yet a component built in one framework's model has no meaning outside that framework—you cannot use an Angular component in an Ember application, short of including multiple overlapping and redundant dependencies. Coupled with the lock-in that frameworks often require, the lack of interoperability makes writing universally reusable, encapsulated components nearly impossible on the Web.

The holy grail, the "Goldilocks" solution, would be if Web developers could pick the application architecture best suited for the problem they were trying to solve, and could reuse interface components across projects. The Web development community could have thriving ecosystems for each framework and organizational

Figure 2. Changing interest in various frameworks over time.



philosophy, as well as a shared, universal ecosystem of components that could be used in any Web application regardless of framework. Large organizations could share sets of globally maintained components that conformed to a consistent style, but that all teams could use regardless of their stacks.

This Web development utopia seems impossible to achieve, or at least technically infeasible, because it would require broad agreement among different frameworks on a consistent treatment for components. The Web, however, has precedent for these universal components—HTML elements themselves.

Consider the `<Select>` Element

The `<select>` element provides a simple dropdown menu. All frameworks understand and can leverage `<select>`—it is baked right into the platform. It works across all browsers with a generally predictable interface. It does one job, and does it well.

Moreover, `<select>` has an API surface area that makes it particularly easy to work with and valuable to use, as shown in Figure 3a–f.

It is *composable* (Figure 3a). `<select>`, composed with “`<option>`” for its items, generates a fully formed dropdown menu.

It is completely *declarative* (3b). A wide variety of different features can be applied using attributes in the markup.

It is *flexible* (3c). Depending on its children and attributes, it can provide different interfaces and functionality.

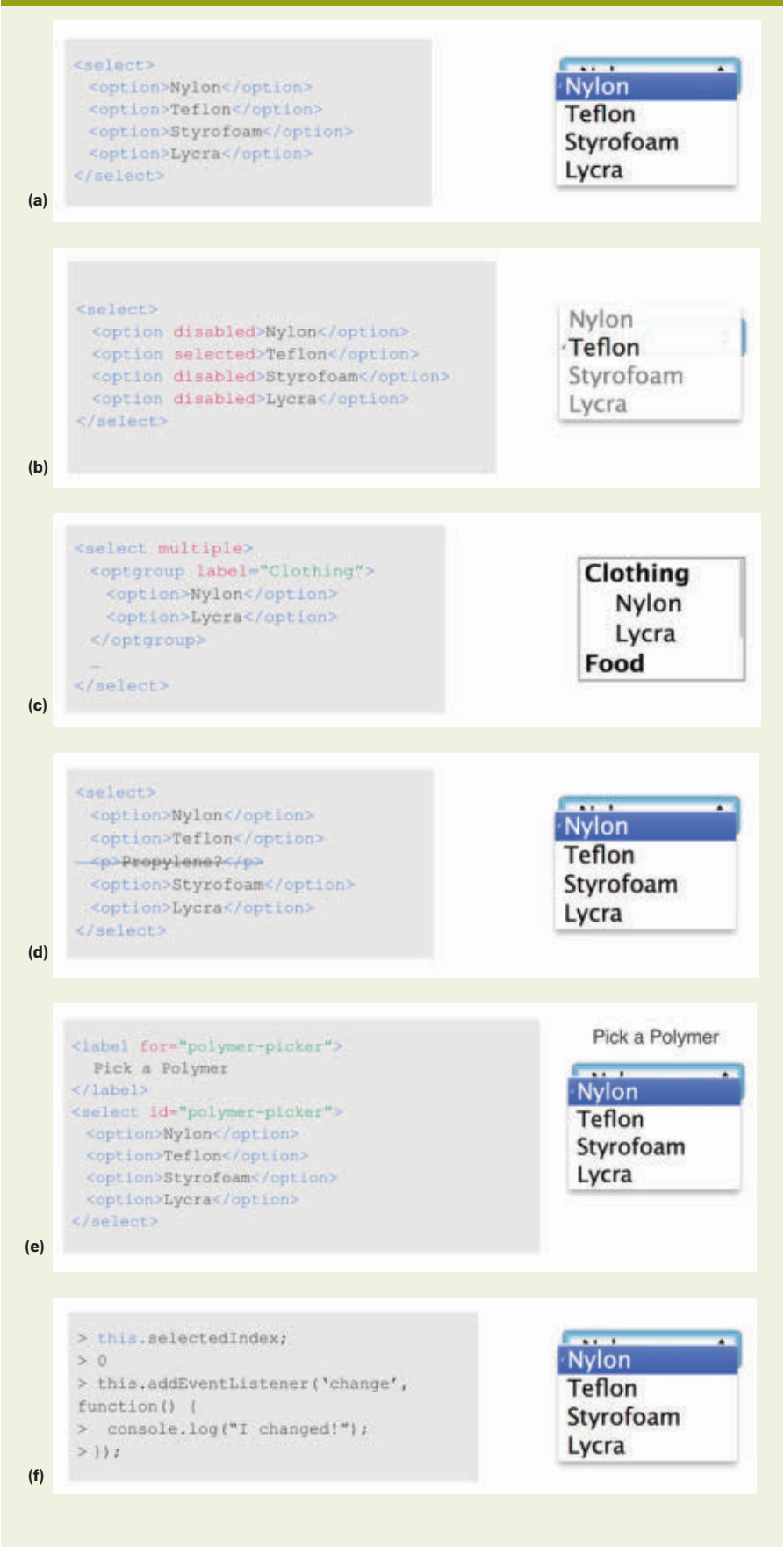
It is *forgiving* (3d). An incorrect child does not crash the application, but is simply ignored.

It is internally *accessible* (3e). Once focused, it provides all the necessary handles in order to be accessible.

Beyond what it provides declaratively, `<select>` can be *scripted* (3f). It emits events that can be listened to and acted on, and it has an imperative API that can be leveraged.

Finally, `<select>` is extremely simple to use, and can be used in just about any context. With minimal declarative markup, a developer gets this extremely powerful behavior. Regardless of the framework, this simple markup and DOM API are well understood and usable.

Figure 3. (a) A composable select element. (b) Declarative attributes. (c) Flexible interface. (d) Forgiving syntax. (e) Accessibility to styles. (f) Scriptable events.



The “<select>” element is certainly useful, but the platform only provides a limited set of such elements, and that set is woefully out of date. There are two potential ways to expand this power: increasing the number of platform-defined elements, or providing the primitives for developers to create their own elements, with all the power of native elements.

With the Extensible Web Manifesto,¹⁴ browser vendors decisively landed on the latter approach, favoring providing the primitives needed to expand Web platform features over providing higher-level abstractions directly in the platform.

What would it take to build an element like this? What sort of primitives and features of the platform might a developer need in order to build as eminently reusable an element as <select>?

A few key pieces of <select> make it useful. It has a declarative API in the form of attributes, an imperative API off its DOM node, a composition model in terms of the child elements it supports, and a standard visual interface.

Thus, to create a similar custom element, a developer would need to:

Define its API: Give the element a name, and give it imperative methods as well as declarative attributes that can be used to affect its behavior.

Define its template: Provide the element with some basic visual layout if it requires any local UI.

Encapsulate it from the document: The element’s internals should be invisible from the document. That is, adding an element to a document should not have unintended side effects.

Define its composition model: Specify what kind of children the element can accept and how it manages its children.

Manage its dependencies: A custom element should be able to use other elements in its local UI, and so should be able to specify and load the definitions of any elements it depends on.

Of course, you could build such encapsulation and template features at the framework level, and many frameworks do. But to realize the dream of broadly interoperable components, these features need to be provided at the platform level, so components built with them can be reused, just like <select>.

Enter Web Components

Web Components is the umbrella term for a handful of new W3C specs that give developers the primitives needed to build such interoperable, platform-level features. The individual specs that make up Web Components map almost directly to the specific features needed to create a truly interoperable element.

Define Its API: The Custom Elements spec² describes how an element might be given a name, define an API surface area, and respond to different events in its life cycle. The registration of a custom element boils down to a simple call as illustrated in Figure 4.

This allows the developer to specify a tag for the element and pass in a prototype for all instances of the element. At this point, all instances of “<my-element>” in the document are upgraded from HTMLUnknownElement to the prototype of the element that was passed in. Note the specific syntax may evolve slightly from the time of writing—for the latest spec, see <http://w3c.github.io/webcomponents/spec/custom/>.

Life-cycle callbacks defined in the custom elements spec give the element author more granular control over the element at particular stages of its life cycle. These callbacks include “createdCallback,” called when the element is created and has been registered; “attachedCallback,” called when the element is inserted into the document; “detachedCallback,” called when the element is removed from a document; and “attrib-

uteChangedCallback,” called when an attribute of the element is set, changed, or removed.

Between the second document.registerElement argument and the attributeChangedCallback, element authors can specify an element’s imperative and declarative APIs.

Define its template: Many elements have some included UI, such as buttons, inputs, selects, headers, and lists. To build a truly platform-level element, element authors should be able to specify the UI for their own element. Rather than define a new DSL or expose C++ hooks, Web developers should be able to use the language of the Web itself—HTML and CSS—to define an element’s template.

True templates have a few key properties. The template’s mere existence should not have side effects for the document, the template should have to be explicitly selected in order to be used and so should be encapsulated away from the main document, and it should be inert until it is actually cloned and used.

The problem of templating on the Web is one all UI frameworks face. A number of workarounds have evolved to provide this behavior, but none meet all the criteria for a true template. Some attempts at templating use a block of markup in the main document with “display: none;” to hide it until it is cloned and used, but this can have layout and performance side effects. Sticking HTML inside a <script> tag is another common approach for templating, but this can lead to security issues with “.innerHTML” and is clunky to manipulate prior to being actually initialized as DOM.

The HTML5 <template> element⁹ provides the complete set of features one would expect for a true template. It is parsed but not rendered, and inert until used, and its contents are encapsulated from the main document in the form of a document fragment. This allows element authors to use a true template to define the look and feel of a custom element, to be cloned and used each time a custom element is created and inserted into a document.

Encapsulate it from the document: Templates give element authors a

Figure 4. A simple call for a custom element spec.

```
var MyElement = document.registerElement('my-element', {
  prototype: Object.create(HTMLElement.prototype)
});
```

way to associate markup with an element, but once the element and its associated markup are inserted into a page, there should be a way to keep them isolated from the document itself. For example, an HTML5 <video> tag has a play button associated with it, but the user of the tag should not have to worry about accidentally styling or selecting the play button when using CSS or “document.querySelector” within their main document.

The Shadow DOM spec⁷ provides the mechanism for this crucial encapsulation. It introduces the notion of a “Shadow Root”—a separate, scoped tree that lives in the DOM but is protected from accidental interference by CSS selectors or DOM manipulation methods. Shadow DOM is the encapsulation primitive that lets elements be used without fear of side effects—either the element accidentally leaking style to its host document, or the host document accidentally leaking effects or style to the element.

Shadow DOM is a subtle and complex but incredibly important primitive. Fortunately, as of this writing, all major browser vendors either have shipped or are working on implementing Shadow DOM.

Define its composition model: To work like a native HTML element, a custom element must be able to accept and manipulate children. The Shadow DOM specification introduces the concept of “Distribution”⁸—the ability to specify insertion points within a shadow root where specific children can be “distributed” into the main document.

This allows a custom element author to define what kinds of children an element accepts and how it interacts with them. Authors can use this to specify how a “select” element might look for and project its “option” children. Distribution essentially provides another API surface area for an element, in the form of children it accepts.

Manage its dependencies: With the ability to define its own internal markup via a template and shadow root, one could imagine a custom element relying on other custom elements for its internal UI. Since multiple elements may depend on the same custom element, there



The scope and diversity of available Web frameworks is immense—a thriving ecosystem that only a platform as powerful and flexible as the Web could support.



must be some way for elements to declare their dependencies and for the browser to load and de-duplicate such shared dependencies.

The HTML Imports specification³ provides this mechanism—a way for an element author or Web developer to load HTML-based dependencies in HTML. As of this writing, the spec authors are working to reconcile this HTML loading and de-duping mechanism with the forthcoming ES6 module loading and de-duping mechanism.

With these four crucial new features—Custom Elements, Templates, Shadow DOM, and HTML Imports—Web developers finally have the platform-level primitives needed to create truly reusable custom elements, with all the power of native HTML elements.

How to Leverage Web Components

The question remains: How can individuals and organizations benefit from the capabilities provided by Web Components?

The most immediate use case for Web Components is in building user interfaces. General best practice in software engineering dictates that systems be isolated and componentized. This guideline can now be directly applied to building Web UI at the platform level, with custom elements as the components.

The first step in building a new Web-based product that uses Web components as interface elements would naturally be to build the set of custom elements the whole product would share. Because custom elements can encapsulate their own look and feel, this element-creation step might include building a visually consistent set of buttons, data tables, menus, layout templates, and other UI components as elements to be used across the application or suite of applications.

Considering the unit of work for a frontend engineering team to be a component, rather than defining the unit of work as a screen or flow, starts to unlock the organizational power of Web Components. By freeing up each individual to focus on a single custom element at a time, teams can minimize both visual inconsistency and duplicate work. All appearances of an

element within an interface—all the buttons used in an application, for example—are reuses of a single custom element, and are consistent in terms of look and feel.

Individual engineers also may benefit from minimizing the cost of switching context between building application logic and UI. By spending the time up front to ensure perfect element design independent of how the application works, pixel-perfection goes from being last-minute polish to being a necessary stage in application development. The app creation stage that follows this component creation stage is also much more streamlined. From the first line of app logic code written, the application looks and feels complete. It becomes much easier to get a sense of the final application throughout the development process, helping catch user experience flaws and begin useful QA processes early in the development cycle.

The efficacy of the custom-element-based interface model is felt even more strongly on a team's second project. Since they have already spent the time to construct a pixel-perfect set of interface elements, the costs of building the second interface are dramatically lowered. This frees the team to focus on new features and overall performance, rather than reinventing UI. Any improvements to the elements that come out of the second project can be seamlessly incorporated into the first project. Each product becomes a capital investment that pays dividends over the entire lifetime of the custom element set.

The engineering practice of reusing elements is also innately different than copying and pasting front-end layout from an old project into a new one. Custom elements, designed in isolation from the beginning, are specifically meant to be flexibly reused. Raw layout code or markup in old projects was rarely designed to be used in a different context, and it can be full of bugs and quirks. Overarching stylesheets lack encapsulation, and can quickly build up cruft and become difficult to maintain. If an element needs to be updated to fit a new use case that becomes a capital



A single consistent set of UI elements helps ensure synergy and efficiency between design and engineering teams. Reused elements ensure brand consistency, guaranteeing the exact same look and feel everywhere.



investment in the custom element that benefits all future users. If raw layout code needs to be updated, it often becomes one additional hack on top of a patchwork of non-systematic edits that rapidly decays into spaghetti code.

A single consistent set of UI elements also helps ensure synergy and efficiency between design and engineering teams. Reused elements ensure brand consistency, guaranteeing the exact same look and feel everywhere. A visual brand will thus be enforced not only at the design level but also at the implementation level. A set of elements provides a living style guide, and makes it much easier to align engineering with design: design no longer occurs in a vacuum, as each visual piece and tweak can be quickly incorporated into a component and tested in situ. Visual design overhauls are also much easier to achieve. Custom elements only have to be redesigned once and the new style can be quickly implemented everywhere, often with a straightforward upgrade to the element definition.

Organizations also benefit from separating custom element creation from element usage. Some engineers will simply be better suited to achieving pixel-perfect designs for elements, because they have an eye for animation, knowledge of the platform quirks, and a passion for visual detail. By having these engineers focus on creating custom elements to be used in many applications, their skill can be leveraged companywide.

Good UI design and performance is often as much art as science, and can take an artist's touch to get just right. Custom UI elements allow this artistic achievement to be broadly shared and utilized. Custom elements help make easy things easier, and difficult things easily repeatable. The broad and long-term advantage to having a single aligned set of UI elements makes it an obvious early investment for an engineering organization.

Perhaps most important, because custom elements based on the Web Component specs are built with the platform rather than with a specific framework, they can be reused regard-

less of what structural framework the next project is built with. The capital investment in a set of visually consistent elements persists far longer than a set of components built for one specific framework technology.

Of course, with great power comes great responsibility. Native HTML elements, built by browser vendors, have accessibility features baked in. The responsibility to make a custom element accessible falls on the element author. Just as a native HTML element would be incomplete if it was not naturally accessible, a custom element must bake in accessibility features as much as possible. To a certain extent, ecosystem dynamics should reward elements that are naturally accessible, but the element author community has a responsibility to explicitly prioritize accessibility from day one. Custom elements are not a magic accessibility wand—high-quality elements will be internally accessible, but application authors must also get accessibility right at the application level.

The Web Component Ecosystem

Beyond the benefits for a single team or organization, one can imagine the network effects that might be possible with an ecosystem of fully interoperable Web components.

Suites of custom elements could be created to make building full-fledged applications on the Web easier: different “UIKits” for different types of Web applications. Custom elements could be built for specific use cases, such as elements that make blogs easier to create, or e-commerce sites more effective and easier to use, or data visualization easier to achieve. Custom elements could breathe new life into the movement for a truly semantic Web, by making it possible to intertwine form and function into a single element. Catalogs could help organize a burgeoning ecosystem of elements, crowdsourcing element ratings, and reviews.

Such an ecosystem, predicated on platform-level interoperability, would require broad adoption of platform-level APIs that give Web developers the ability to create custom elements. The Web Components specs are a substantial undertaking. Since

their introduction in 2011 they have generated lively discussion and have evolved based on feedback. Though there is general agreement on the value of Web components, there are two important contentious pieces of the specs being worked out by implementers—reconciling HTML Imports with the forthcoming ES6 module system, and ironing out specifics of Shadow DOM behavior. You can follow and join the conversation on the public-webapps mailing list.⁶

Currently, the Template element is part of the living HTML spec⁹ and is broadly supported by modern browsers. HTML Imports, Shadow DOM, and Custom Elements have been seeing growing cross-browser enthusiasm, especially after recent meetings to address the more contentious pieces. They have been shipped in their entirety beginning with Chrome 36. Microsoft Edge recently announced¹² it is starting development on the HTML Template element, and stated positive views on the latest evolutions of the remaining specs. Firefox is shipping implementations under a flag, and recently published an in-depth article¹³ on the history of Web components, with the hopeful conclusion we are nearing broad cross-browser support.

Fortunately, developers can start building with the complete Web Components APIs today, using the comprehensive set of Web components polyfills.⁴ These JavaScript implementations support Custom Elements, HTML Imports, and Shadow DOM across the last two versions of major browsers starting with IE10, Safari 7, and the evergreen browsers Chrome and Firefox. Web Components-based libraries such as X-Tag,¹¹ Polymer,⁵ and Bosonic¹ rely on some of the polyfills for broad browser support, and include optimizations around the heavier parts of the polyfills to achieve production-ready performance.

Web developers today have it tough. But with the consistency of a sane, platform-level component model coupled with the wild, expansive power of the Web ecosystem, we might be on the cusp of a revolution in Web development. Happy componentizing!

Related articles on queue.acm.org

Web Services: Promises and Compromises

Ali Arsanjani, Brent Hailpern, Joanne Martin, and Peri Tarr

<http://queue.acm.org/detail.cfm?id=639315>

From COM to Common

Greg Olsen

<http://queue.acm.org/detail.cfm?id=1142043>

A Conversation with Roger Sessions and Terry Coatta

<http://queue.acm.org/detail.cfm?id=1095416>

References

1. <http://bosonic.github.io/>.
2. <http://w3c.github.io/webcomponents/spec/custom/>.
3. <http://w3c.github.io/webcomponents/spec/imports/>.
4. <https://github.com/webcomponents/webcomponentsjs>.
5. <https://www.polymer-project.org/1.0/>.
6. <https://lists.w3.org/Archives/Public/public-webapps/>.
7. <https://w3c.github.io/webcomponents/spec/shadow/>.
8. <https://w3c.github.io/webcomponents/spec/shadow/#distributions>.
9. <http://www.w3.org/TR/html5/scripting-1.html#the-template-element>.
10. <https://github.com/tastejs/todomvc>.
11. <http://x-tags.org/>.
12. Leithead, T. and Eicholz, A. Microsoft Edge and Web Components; <https://blogs.windows.com/msedgedev/2015/07/15/microsoft-edge-and-web-components/>.
13. Page, W. The state of Web Components; <https://hacks.mozilla.org/2015/06/the-state-of-web-components/>.
14. W3C Extensible Web Community Group. Extensible Web Manifesto, 2013; <https://extensiblewebmanifesto.org/>.

Taylor Savage is a product manager on the open Web platform team and lead PM on the Polymer project at Google. Prior to Polymer, he worked as project manager on new features for Google search.

**A metro map can tell a story,
as well as provide good directions.**

**BY DAFNA SHAHAF, CARLOS GUESTRIN,
ERIC HORVITZ, AND JURE LESKOVEC**

Information Cartography

“RAISE YOUR HAND if you don’t quite understand this whole financial crisis,” said David Leonhardt’s *New York Times* article, March 2008. The credit crisis had been going on for seven months and extensively and continuously covered by every major media outlet in the world. Despite that coverage, many readers felt they did not understand what it was about.

Paradoxically, pervasive media coverage may have contributed to the public’s lack of understanding, a phenomenon known as information overload. Recent technology advances allow us to produce data at bewildering rates, while the surge of the Web has brought down the barriers of distribution. Yet despite this accelerating data deluge, knowledge and attention remain precious and scarce commodities. Writers, researchers, and analysts spend countless hours gathering information and synthesizing meaningful narratives, examining and inferring relationships among pieces of information. Subtleties and relationships in an evolving story are easy to lose in an echo chamber created by the modification and reuse of content, as fueled by incentives to attract indexers, eyeballs, and clicks on advertisements.

The problem of automatically extracting structured knowledge from large datasets is increasingly prevalent.

Several methods have sought to summarize and visualize narratives.^{2,28,29} However, most work only for simple stories that are linear in nature. In contrast, complex stories exhibit a nonlinear structure; stories spaghetti into branches, side stories, dead ends, and intertwining narratives. To explore them, users need a map to guide them through unfamiliar territory.

We previously introduced a methodology for creating structured summaries of information we call “metro maps.” The name is metaphoric; just as cartographic maps have been relied on for centuries to help us understand our surroundings, metro maps help us understand the information landscape. In this article, we explore methods we have developed for automatically creating metro maps of information.^{25–27}

Metro maps consist of a set of lines with intersections or overlaps. Most important, they explicitly show the relationships among different pieces of information in a way that captures a story’s evolution. Each metro stop is a cluster of articles, and lines follow coherent narrative threads. Different lines focus on different aspects of the story; for example, the map in Figure 1 was automatically generated for the query “Crimea.” The map outlines the

» key insights

- **Though human attention and comprehension can be overwhelmed by the data deluge, automatic methods can extract structured knowledge and provide maps of complex information landscapes to help people understand ideas, connections, and storylines.**
- **Properties of good maps are difficult to formalize; important characteristics include coherence of storylines, coverage of diverse and important topics, and relationships among pieces of information.**
- **These principles can be used to synthesize meaningful narratives from large datasets across multiple domains, including news stories, research papers, legal cases, and works of literature.**

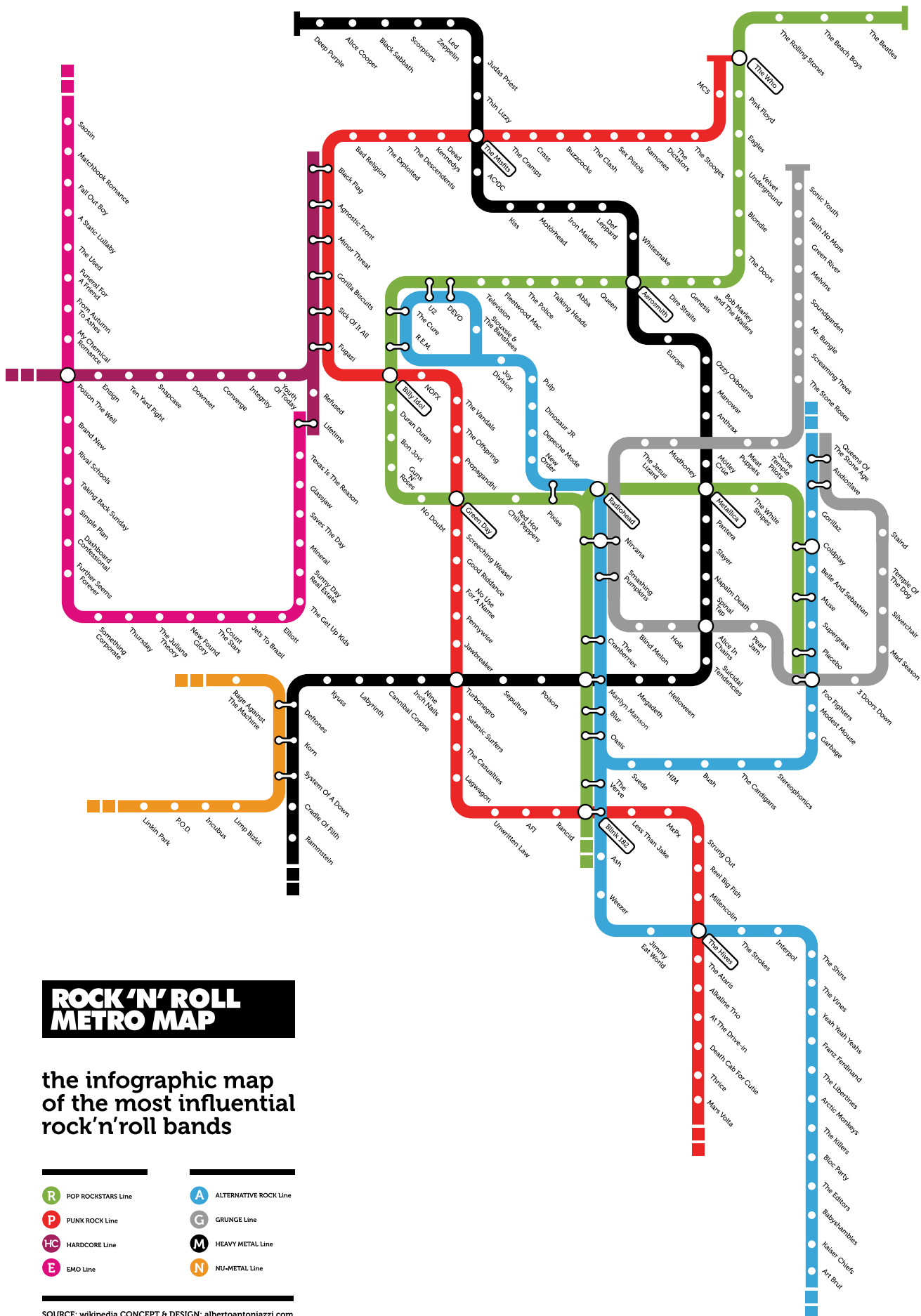
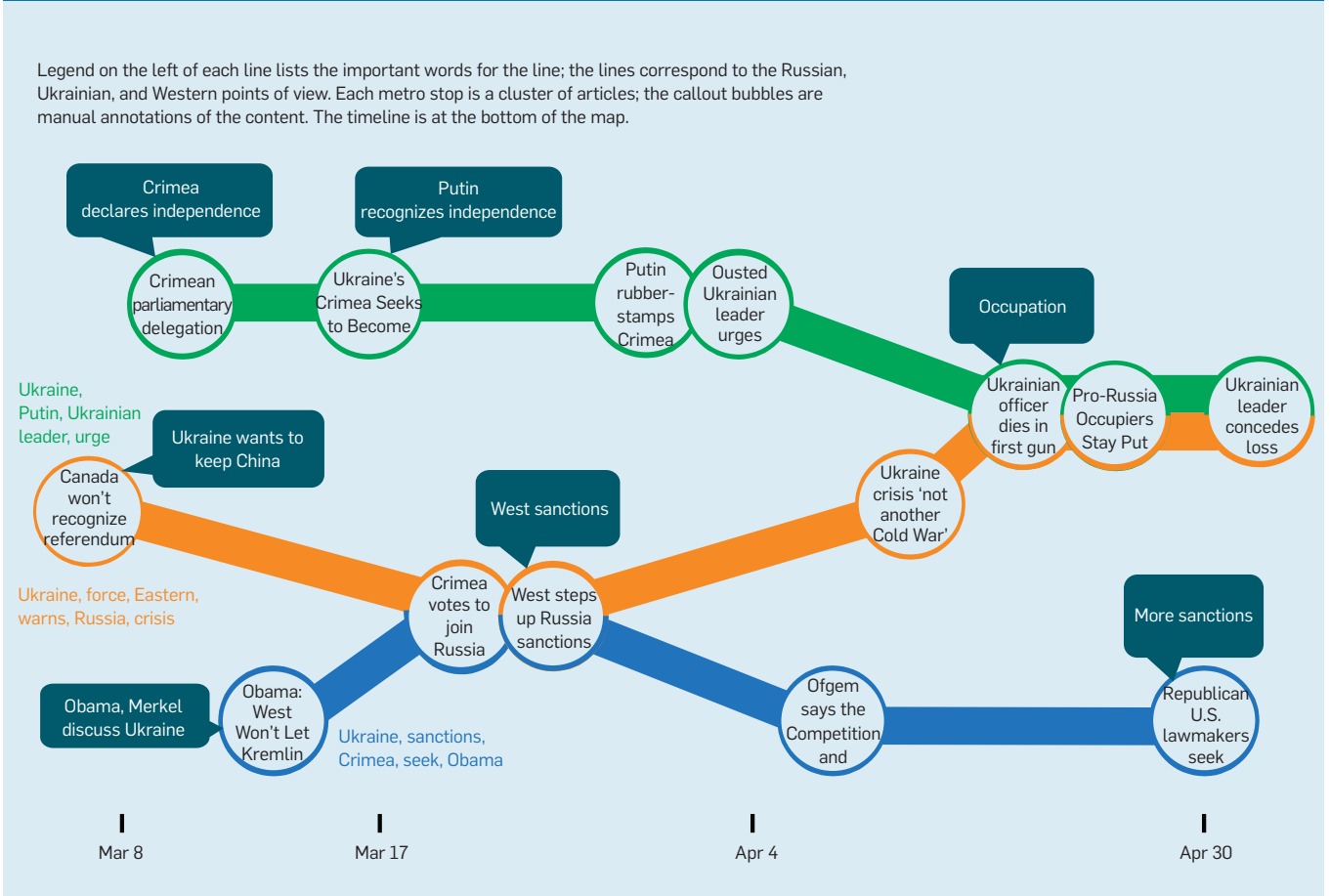


Figure 1. Sample output: metro map of the 2014 Crimean crisis.



2014 Crimean crisis, with the three lines corresponding to the Russian, Ukrainian, and Western points of view. The legend to the left of each line shows the important words for the line. The timeline appears at the bottom of the figure. The Russian (green) line starts in March, with the Crimea parliament voting to join Russia and Vladimir Putin recognizing Crimean independence. The Ukrainian (orange) line starts with Ukraine's former prime minister urging the West to stop Russian aggression. The Ukrainian line then joins the Western (blue) line to discuss the West's attempts to support Ukraine. Finally, the Russian and Ukrainian lines intersect when pro-Russia groups took over police stations in Ukraine.

Our representation is motivated by the strong empirical evidence that map representations help users gain and retain knowledge; for example, mind maps and knowledge maps have been shown to increase memory recall in students,^{11,23} as well as motivation and concentration.¹⁵ We have also found map visualizations enable

users to digest information. We also integrated capabilities for supporting user interaction into the methodology, letting users guide formulation of the maps.

We demonstrate that metro maps can help people understand information in many areas, including news stories, research areas, legal cases, even works of literature. Metro maps can help them cope with information overload, framing a direction for research on automated extraction of information, as well as on new representations for summarizing and presenting complex sets of interrelated concepts.

Finding a Good Map

We start by formalizing the characteristics of good maps and formulating their construction as an optimization problem. We then provide efficient, scalable methods with theoretical guarantees for constructing maps. Our description of the characteristics is intentionally abstract. Later, we demonstrate how to adapt these abstract notions to various domains.

An objective function. Before we can come up with an algorithm for computing good maps, we must craft an objective function, which is especially important for maps, where the objective is not clear, a priori. In the following sections, we motivate and formalize several (sometimes conflicting) criteria. In the next section, we present a principled approach to constructing maps that optimizes trade-offs among these criteria.

First, recall our goal. Given a set of documents, we seek to compute a metro map that summarizes and organizes the documents. A metro map consists of a set of metro lines, each an ordered sequence of stops, where a stop is a subset of articles. Each line follows a coherent narrative thread, and different lines focus on different aspects of the story. Intersections across lines reveal the ways different storylines interact; for example, we computed the map in Figure 1 over news articles containing the word "Crimea" from March to April 2014. Each stop is a cluster of articles. The map includes three story-

lines, following the Russian, Ukrainian, and Western points of view.

Coherence. A first requirement is that each metro line tells a coherent story; following the articles along a line should give the user a clear understanding of the evolution of a story.

Consider a chain of clusters, where a cluster is a set of documents. For the sake of the presentation, we focus on singletons, with each cluster a single document. In order to define coherence, a natural first step is to measure similarity between each two consecutive articles along the chain. As a single bad transition can destroy the coherence of an entire chain, we measure the strength of the chain by the strength of its weakest link.

However, this simple approach can produce poor chains. Consider, for example, chains A and B. Both have the same endpoints, yet Chain A is significantly less coherent. Note the transitions of Chain A are all reasonable when examined out of context; the first two articles are about debt default, the second and third are about Republicans, and so on. Despite these local connections, the overall effect is incoherent.

Now take a closer look at the two chains. Figure 2 shows word appearance along both chains; for example, the word “Greece” appears throughout Chain B. It is easy to spot the associative flow of Chain B. Words appear for short stretches; some words appear, then disappear and reappear. Contrast this with Chain A, where stretches are longer and transitions are smoother. This observation motivates our definition of coherence.

We transform the problem into a linear programming optimization problem, where the goal is to choose a small set of words and score the chain based solely on these words. To ensure the strength of each transition, the score of a chain (given a set of active words) is the score of the weakest link; see Shahaf and Guestrin²⁴ for details.

The score of a single link might depend on the domain. In Shahaf et al.²⁶ we showed how to compute a score, given article content alone. In Shahaf et al.,²⁵ we showed how to take advantage of links among articles.

Coverage. Coherence is crucial for

good maps, but is it sufficient? Pursuing an answer, we found maximally coherent lines for the query “Bill Clinton.” The results were discouraging. While the lines were indeed coherent, they were not important. Many lines revolved around narrow topics (such as Clinton visiting Belfast). Moreover, as there was no notion of diversity, multiple lines included redundant information. This example suggests selecting the most coherent lines does not guarantee a good map. Instead, the key challenge is balancing coherence and coverage; in addition to being coherent, lines must also cover diverse topics important to the user.

We define a set of elements the map can cover. The elements can depend on the domain; in the case of news articles, we select words (such as “Obama” and “China”),²⁶ so a high-coverage map discusses many important words. In the case of a scientific corpus, we select papers²⁵ so a high-coverage map touches a large chunk of the corpus.

We calculate a coverage function, measuring how well each document covers each element. We extend it to a set function, measuring how well a set of documents covers each element. In order to encourage diversity, this function is submodular; if the map covers an element well already, adding another document covering that element well thus provides little extra coverage. Lack of ex-

tra coverage encourages us to pick documents that cover new topics instead.

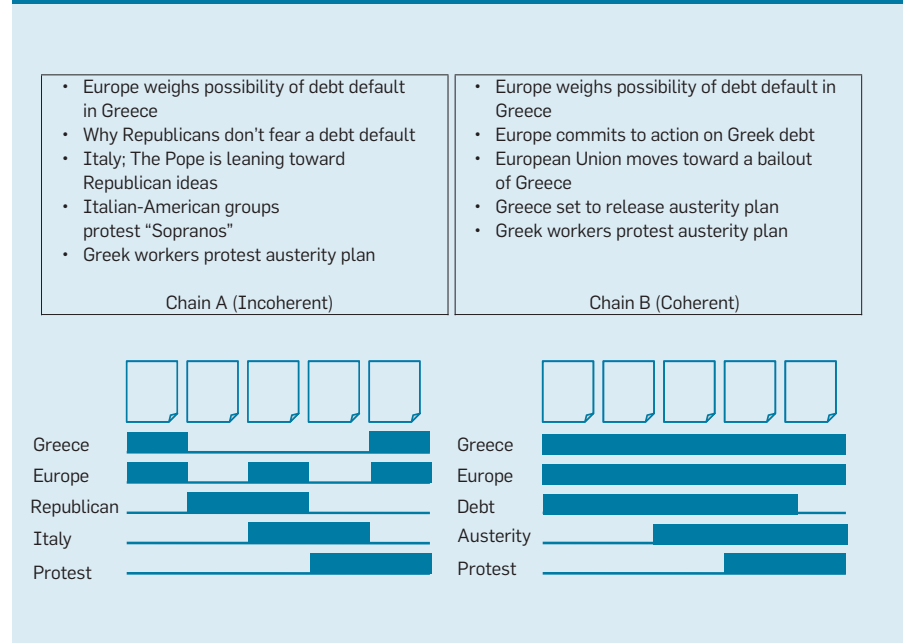
We next introduce weights for each element, indicating the element’s importance. The weights bias the map toward covering important elements while also offering a natural mechanism for personalization. In Shahaf et al.,²⁶ we discussed learning weights from user feedback, resulting in a personalized notion of coverage.

Connectivity. Finally, a map is more than a set of lines, with information in its structure as well. Our final property is thus connectivity. A map should convey the underlying structure of the story and how different aspects of the story interact.

Intuitively, different stories have different structures. Some stories are almost linear, while others are much more complex. In order to capture the structure of a story, we compute the minimum number of lines that cover all metro stops. This objective prefers long storylines whenever possible; linear stories become linear maps, while complex stories maintain their interweaving threads.

Tying it all together. We now formulate the problem of finding a good metro map, given a set of documents. We need to consider trade-offs among the properties discussed earlier: “cluster quality,” “line coherence,” “map structure,” and “coverage under bud-

Figure 2. Word patterns in Chain A (left) and B (right); bars correspond to word appearance in the articles listed above.



get”; for example, maximizing coverage leads to a disconnected map, as there is no reason to reuse a cluster for more than one line. Maximizing coherence often results in repetitive, narrow-scope chains. It is thus better to treat coherence as a constraint; a chain is either coherent enough to be included in the map, or it is not. Coverage and structure, on the other hand, should both be optimized. We define the map objective like this:

Problem 1 (Metro maps: Informal)
 A map must satisfy
 High coverage (o1)
 High structure quality (o2)
 Subject to
 Minimal level of line coherence (c1)
 Minimal cluster quality (c2)
 Maximal map size (c3)
 See Shahaf et al.²⁷ for a formal statement of the algorithm and optimization.

Algorithm
 We now briefly review the main ideas behind the algorithm, which starts by computing a set of documents from a query. We then segment the articles into time windows and compute good clusters for each window (constraint c2 in Problem 1) using a community-detection algorithm on word co-occurrence graphs.²⁷ These clusters serve as metro stops.
 Once we have clusters, we can proceed to computing coherent lines (con-

Figure 3. A metro map for the query “Boston” in May 2013.

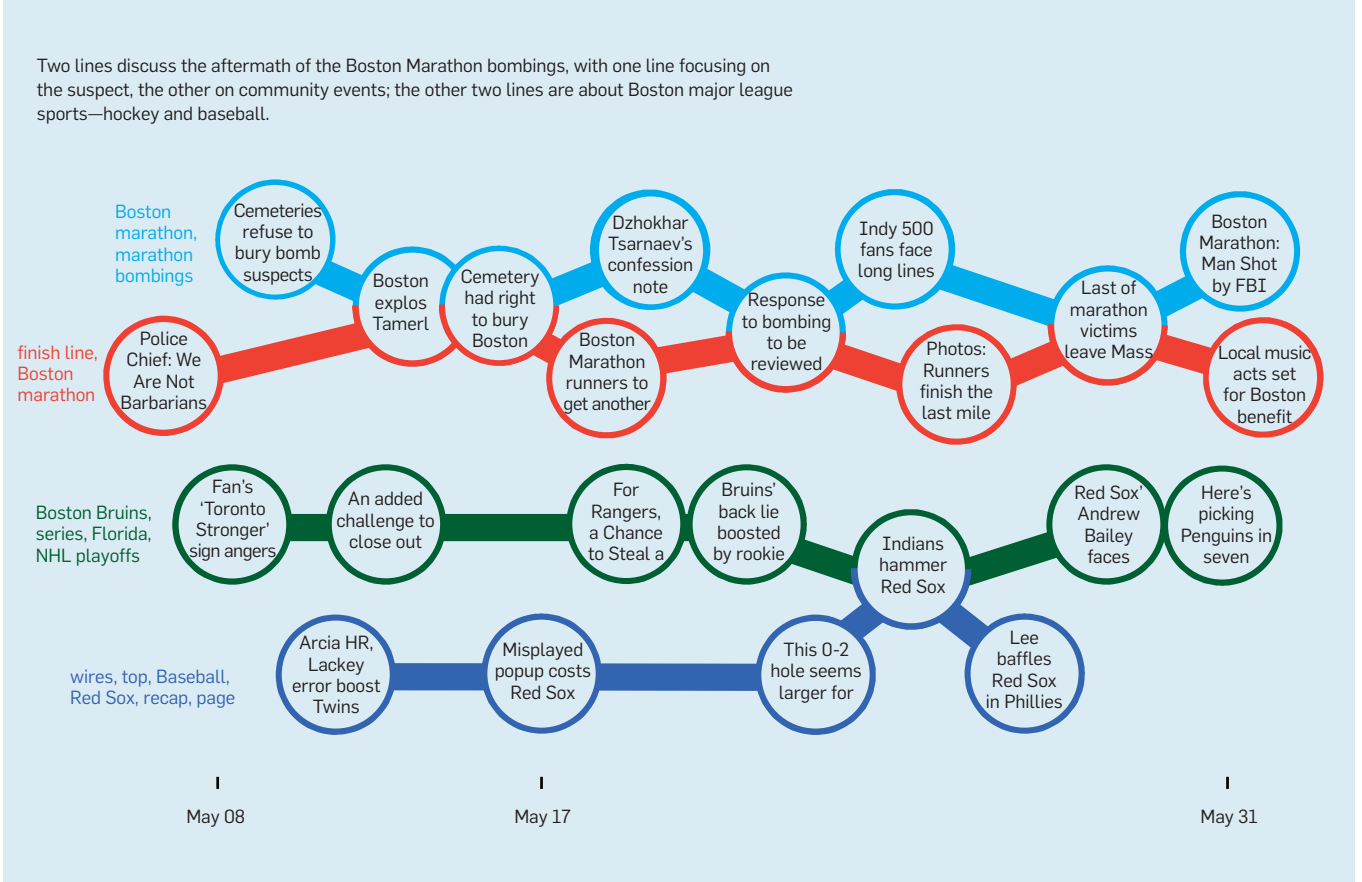
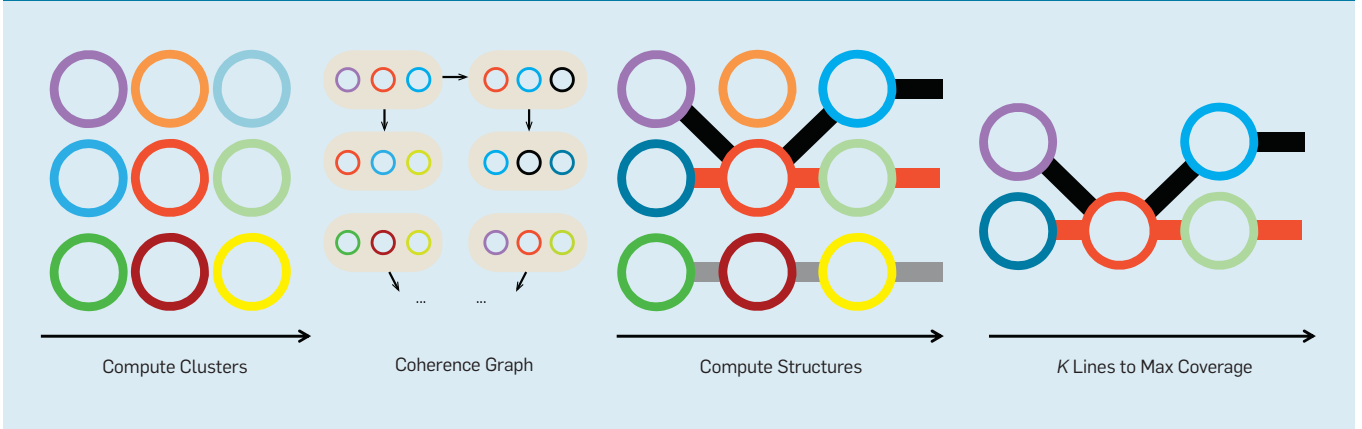


Figure 4. Overview of the algorithm. We compute clusters, encode coherent lines in a graph, and use the graph to compute the structure of the story. We then pick K lines from the structure that maximize coverage.



straint $c1$). Ideally, we could enumerate all possible candidate lines, but that is often infeasible. We instead propose a divide-and-conquer approach, constructing long lines from shorter ones. It allows us to compactly encode many candidate lines as a graph; the nodes of the graph correspond to short coherent lines, and edges indicate lines that can be concatenated and remain coherent. Paths in the graph thus correspond to coherent lines.

After encoding all coherent lines, we identify the underlying structure of the story, optimizing a connectivity objective ($o2$) that prefers longer story-lines whenever possible. The objective, while difficult to optimize exactly, is submodular and can be efficiently approximated within guarantees.

A story may be very complex, but the user's attention span is limited. To keep maps manageable, our final step is to restrict their size. We pick up to K lines that obey the structure and maximize coverage ($c3$ and $o1$). We rely on submodular optimization again to optimize the map within theoretical guarantees. Example outputs of the algorithm are in Figure 1 (Crimea) and Figure 3 (Boston).

Complexity and running time. Given a query set of documents D , we first run a linear-time algorithm, compiling D to a sequence of word co-occurrence graphs. More important, the size of the graphs does not depend on D but on the size of our vocabulary W . Our dependency on the size of D is linear, and our algorithm scales well (see Figure 4); see Shahaf et al.²⁶ and Shahaf et al.²⁷ for theoretical guarantees. Our main bottleneck is the coverage step, which is polynomial of a high degree in $|W|$. A parallel implementation and lazy evaluations achieve the same approximation guarantees while often leading to dramatic speedups.

In practice, our system often takes less than a minute for query sets including tens of thousands of documents. Note while our system could in principle support even larger query sets, the use case we have in mind for maps rarely necessitates it. We speculate that very broad queries (“U.S.”) would be less common than narrower ones (“health care reform”).

Parameters. Tuning is required for several parameters to obtain a good metro map. In particular, each con-



The key challenge is balancing coherence and coverage; in addition to being coherent, lines must also cover diverse topics important to the user.



straint in Problem 1 has an associated parameter that is manually adjusted on training queries. Another important parameter is m , the user's “history window,” or number of previous articles in the line the user can remember. Higher m results in more coherent chains but is more computationally expensive. In practice, we choose the highest m we can afford computationally.

Applications

In the previous section, we discussed metro maps in the news domain, but maps are easily applied to other domains. The main principles—coherence, coverage, and connectivity—are the same, but one can use domain knowledge to improve the objective. In the following sections, we discuss four applications: news, science, legal documents, and books.

News. News media play a pivotal role informing the public of social, cultural, and political issues. Understanding news enables the public to make key life decisions (such as choosing a place to live or a political orientation). The consequences of acting without understanding the big picture can be adverse. However, with the increasing amount of content published every day, readers can easily miss the big picture amidst a flood of data.

Approach. We used our algorithms to compute maps about news events, assembling multiple news datasets covering hundreds of thousands of posts from Internet news sources; for a demonstration of the system see <http://metromaps.stanford.edu/>.

Evaluation. Quantitatively evaluating metro maps is difficult. There is no established gold standard for doing it, and ground truth is difficult to define. Since the goal of the maps is to help people navigate information, we conducted a user study to better understand the value of the methodology. The study took place in 2011 at Carnegie Mellon University in Pittsburgh, PA, aiming to test whether the maps we generate are useful for humans seeking information about complex topics.

To demonstrate a deep understanding of a topic, we asked users to explain it to others. We recruited 15 undergraduate students, asking them to write two paragraphs, one summarizing the Haiti earthquake and one summarizing

the Greek debt crisis. We randomly assigned them a metro map or the Google News result page. We computed the maps from a corpus of more than 18,000 articles from the *New York Times*. We employed crowdworkers on Mechanical Turk (<http://www.mturk.com/>) to evaluate the paragraphs. In each round, we gave the crowdworkers two paragraphs (map user vs. Google News user) and asked them to assess which one provided a more complete and coherent picture of the story. After removing spam, we had 294 evaluations for Greece and 290 for Haiti; 72% of the Greece comparisons preferred map paragraphs, but only 59% of the Haiti comparisons preferred map paragraphs. After examining the Haiti paragraphs, we found most map users focused solely on the major storyline (on distributing aid). Based on the results of the study, we speculate maps are more useful for stories without a single dominant storyline.

Science. As the number of scientific publications soars, even the most enthusiastic reader can have trouble staying on top of the evolving literature. We are motivated by the idea of creating valuable literature exploration tools that can help people entering a new field (such as new graduate students or experts reaching beyond their traditional discipline's borders).

Approach. We extended our techniques to the scientific domain, aiming to test whether maps can help researchers understand the state of the art of a field. We modified the objective slightly (see Shahaf et al.²⁵), taking advantage of the citation graph. Our dataset included more than 35,000 papers from ACM conferences and journals.

Figure 5 outlines part of a map we computed for the query “reinforcement learning,” depicting multiple lines of research, including Markov decision processes, robotics and control, bounds and analysis, exploration-exploitation trade-offs, and multiagent cooperation. Note the lines in the figure do not intersect. Intersection in the scientific domain is difficult; a theory line and an application line can be highly related, yet no single document belongs to both. We thus modified our objective to allow for a softer kind of connectivity, where lines can interact through citations; for example, the map shows how the Markov Decision Process line affects the



Maps are designed to display connections between multiple pieces of information.



multi-agent and robotics lines and how the exploration-exploitation line interacts with the analysis line. These relationships are gray dashed paths, with relevant citation text nearby.

Evaluation. To test our maps, we recruited 30 graduate students from Carnegie Mellon University, asking them to conduct a quick literature survey in reinforcement learning, an area they had not studied. In particular, we asked them to update a survey paper from 1996 by identifying up to five research directions that should be included in the updated survey and listing a few relevant papers for each direction. We recorded their browsing histories and took a snapshot of their progress every minute. We limited their time to 40 minutes to simulate a quick first pass on the papers. All participants used Google Scholar,^a a search engine that indexes scholarly literature. In addition, we gave 15 of them a metro map. Allowing them to query Google Scholar's entire set of publications makes the task both more realistic and more difficult for maps.

An expert graded the output of all participants. We wanted them to find good papers, as well as identify important research areas. We thus measured precision (fraction of retrieved papers that are relevant) and subtopic recall (fraction of relevant research areas retrieved). Map users outperformed Google-only users in every parameter, with an average score of 84.5%, discovering on average 1.62 seminal papers. Google users achieved a score of 74.2%, finding 1.2 seminal papers on average. The map users' average recall score was 73.1%, compared to Google's 46.4%.

Further analysis of the snapshots we took throughout the study provides anecdotal evidence of the map's utility. Google users visited more pages and listed more papers on average. However, when looking at the average ratio, only one of 4.5 pages visited by Google users was added to their list, while map users added one of 3.8 pages. That is, the map users appeared to be more focused; they may have visited fewer pages but found them satisfactory. In addition, several map users started by composing a short list of research directions, then progressively added papers to each direction

^a <http://scholar.google.com>

throughout the session. Google users, in contrast, did not exhibit such “big picture” behavior.

Legal documents. Law is built on the evolution of ideas with links to key precedents. Legal scholars and lawyers routinely do research on legal corpora, dealing with an avalanche of information. Despite the related informational challenges, legal documents and review processes remain largely untouched by technology. We sought to explore the value of metro maps to help lawyers argue a case, envisioning a system that would help them find related cases, understand how the law has evolved (and why it changed), and prepare a case strategy accordingly.

Approach. Our data consisted of U.S. Supreme Court decisions as supplied by Ravel Law.^b Unlike news articles and scientific papers, Supreme Court decisions can be lengthy, reaching hundreds of pages. The simple text-processing methods of the earlier sections could not separate the wheat from the chaff for legal scholars.

^b <http://www.ravellaw.com>

Addressing this challenge, we focused on anchor text, or text surrounding citations. Identifying highly cited paragraphs allowed us to focus on the important parts of each case; for example, *Women’s Community Health v. Cohen* cites *Roe v. Wade*, saying, “[T]he Supreme Court held that the constitutional “right of privacy ... is broad enough to encompass a woman’s decision whether or not to terminate her pregnancy” (Id. 410 U.S. at 153 93 S.Ct. at 727). We used the anchor text to compute our input set of documents and applied our map algorithm.

Evaluation. As a reality check, we computed a map for the query “commerce clause,” which appears in Article I, Section 8 of the U.S. Constitution, saying the Congress has power “To regulate Commerce with foreign Nations, and among the several States, and with the Indian Tribes.” It is an important clause that has been thoroughly discussed in multiple courts.

Figure 6 (left) shows the map we computed and shown to lawyers from Ravel Law, asking for their interpretation. They browsed through it and manually labeled each line. As a reality check, we

then computed the words that made each line coherent for our algorithm. Figure 6 (right) shows the comparison. The coherent words correspond well to the lawyers’ annotation; for example, the purple line deals with the question of whether Congress may abrogate the Eleventh Amendment immunity of the states. This line was labeled “eleventh amendment, state sovereignty” by the lawyers, and its coherent words were “immunity,” “sovereignty,” “amendment,” and “eleventh.”

We further asked the lawyers to explain each line. As an example, consider again the purple line in Figure 6. The line starts with *Ford Motor v. Dept. of Treasury*. The Court held the Eleventh Amendment denies to the federal courts authority to entertain a suit by private parties against a state without the state’s consent.

In the next stop (a case known as *Parden v. Terminal Railway*), the Court discussed whether a state owning a railroad could successfully plead sovereign immunity in a federal court suit by its employees. In *Employees v. Dept. of Public Health*, the Court noted even

Figure 5. Part of the map computed for the query “reinforcement learning.”

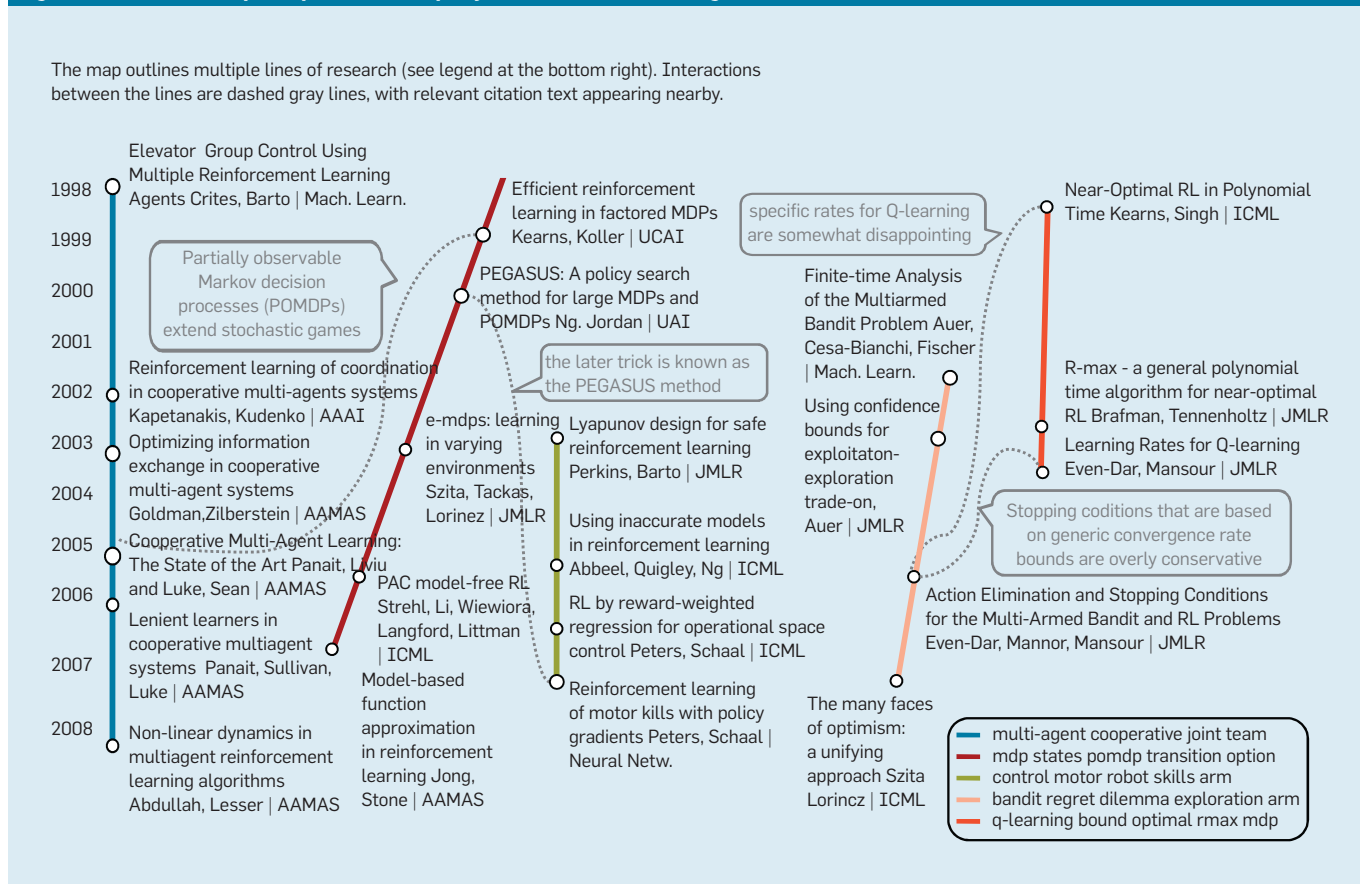
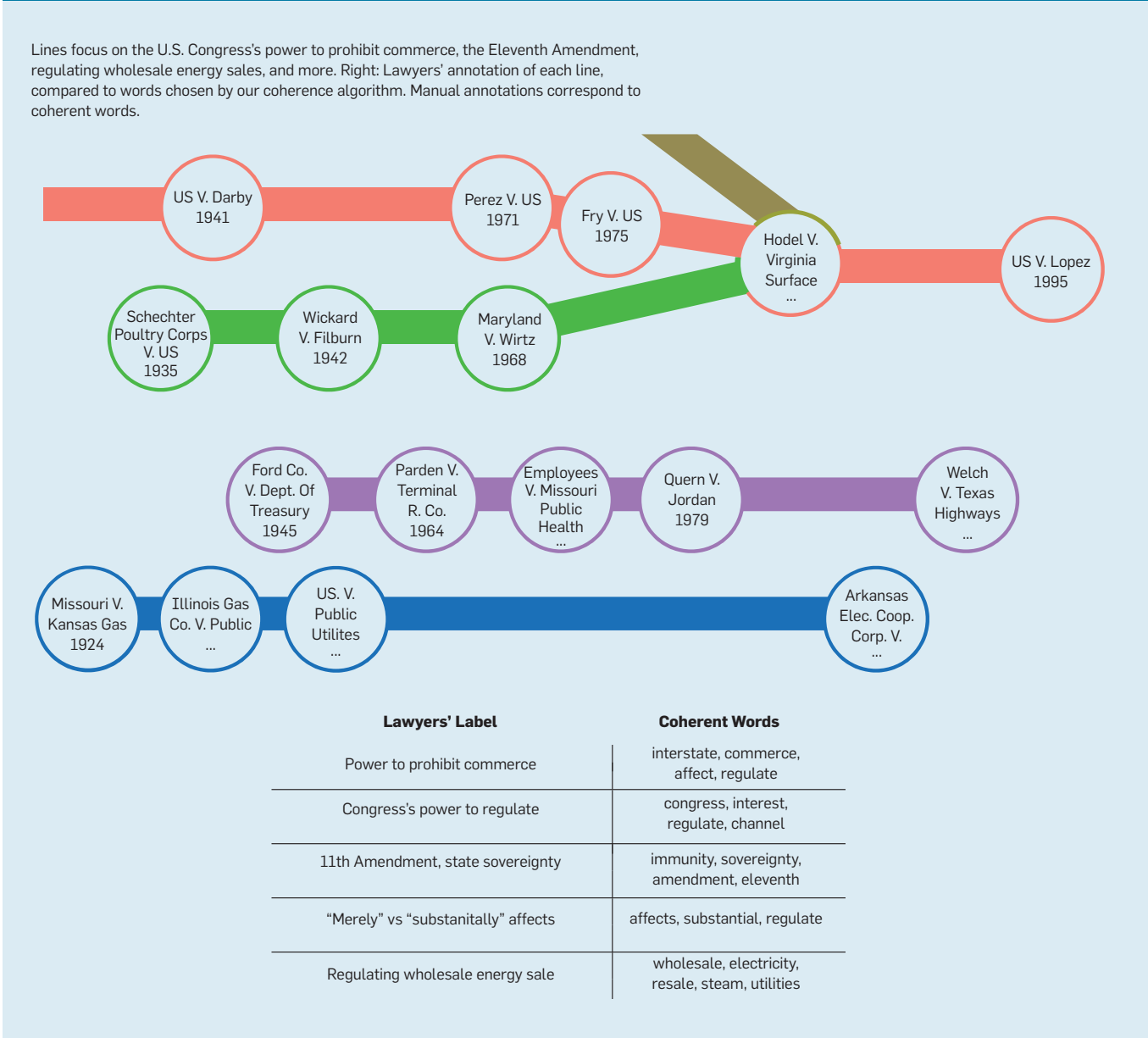


Figure 6. Detail of a map over legal documents for the query “Commerce Clause.”



if the Court reverses itself, Congress can provide authority for suits in state courts to implement federal statutory rights, thus doing away with common law sovereign immunity of the states. Next, in *Quern v. Jordan*, the Court held Congress had not intended to include states within the term “person” for the purpose of subjecting them to suit. Finally, in the last stop (a case known as *Welch v. Texas Highways*), the Court overruled an aspect of *Parden* regarding states participating in federal spending programs.

Likewise, the lawyers could explain all other lines, expressing their confidence in the benefit of maps to the legal community.

Books. Narratives are important in many areas, including literary criticism, political science, and linguistics. Despite it, we know little about their structure. We thus decided to apply metro maps to elucidate the structure of complex books. Our first test case was *The Lord of the Rings*, an epic fantasy novel of more than 480,000 words, internally divided into six books. It includes a long list of characters that can be difficult to follow for even the most dedicated reader.

Approach. Since our maps operate on a collection of documents, we partitioned the book into three-page segments, treating each segment as a document. Applying the maps algorithm to this collection of segments, we

encountered an interesting problem with our notion of coherence, originally developed in the context of news. Journalists often remind their readers of previous events, and thus coherence could be inferred through repetition. In contrast, book authors do not recap events that happened several pages ago but rather rely on the memory of engaged readers. We therefore relied on other hints for coherence.

Noting a character's point of view is often a coherent narrative, we thus decided to focus on named entities. We identified the characters present on each page and looked for storylines with co-occurring characters.

Evaluation. Figure 7 shows a seg-

ment from *The Lord of the Rings* map that reveals important structural information: The story begins in the Shire (leftmost cluster) where Gandalf meets the hobbits. In the pages associated with this cluster, Gandalf advises Frodo to take the ring away from the Shire. In the second cluster, Frodo leaves, accompanied by Sam, Merry, and Pippin. They take the Strider, later revealed to be Aragorn, as guide and protector.

In the next cluster, Aragorn leads the hobbits to Rivendell where the Council of Elrond meets. The Council decides the Ring must be destroyed, and a “Fellowship of the Ring” is formed, including Sam, Merry, Pippin, Aragorn, Gandalf, Gimli, Legolas, and Boromir.

The council cluster splits into three lines, as in Figure 7, corresponding to the fellowship splitting up when orcs kidnap Merry and Pippin (green line). Aragorn, Gimli, and Legolas pursue the orcs (purple line), while Frodo and Sam continue on their own, capturing Gollum (blue line), and head toward Mordor, the region controlled by Sauron and his servant, Saruman (yellow line).

This example, while preliminary, demonstrates the potential of maps for sorting through complex plotlines.

Using Maps

We have discussed the process of creating maps and now shift to the user, exploring possible uses of maps. We rely on the traditional information-retrieval framework, characterizing a user by an information need. Users formulate their information needs and submit queries to a system. If not satisfied with the results, they may interact with the system through reformulated queries until they are satisfied. In this section, we discuss information needs and interaction scenarios.

Information needs. Maps are not intended to replace search engines; many search-engine queries are extremely focused, and corresponding information needs are often satisfied with a simple phrase. In contrast, maps are designed to display connections between multiple pieces of information. In terms of Broder’s taxonomy,⁷ maps are mostly useful for informational queries, and of little use for navigational and transactional queries.

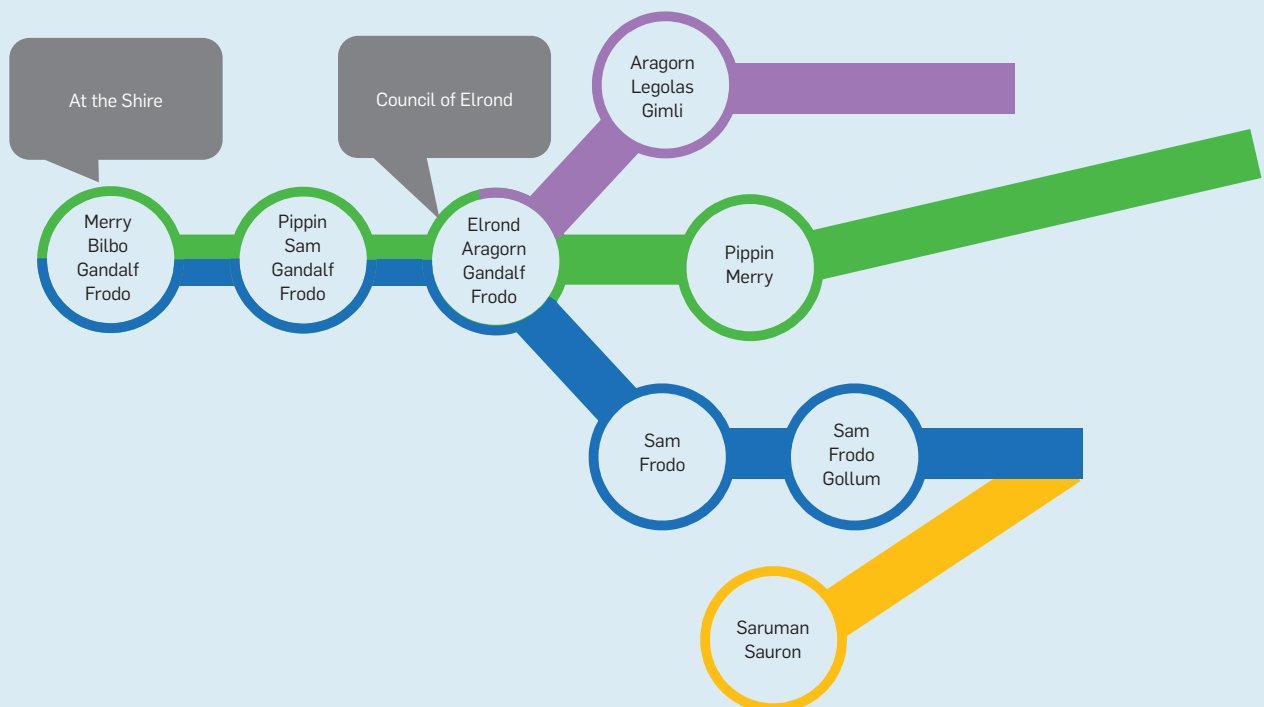
We would like to characterize the information needs of map users. Informational queries are driven by a user’s need to learn something. In order to characterize the different types of learning, we

rely on Bloom’s taxonomy,³ which identifies six cognitive categories characterizing the processes of learning, from recalling facts to making judgments.

We distinguish between two categories of information needs for map users. “Learn” corresponds to the lower three levels of Bloom’s taxonomy; the user’s goal is to acquire knowledge. A user in the learn category might be interested in surveying a concrete topic, which may be new to the user or a familiar one the user wishes to monitor. Alternatively, the user may wish to explore and navigate around a starting point. Navigation is a promising application for maps, with many news sites today including a “related articles” function. Maps can augment it, allowing users to see the article in a broader context. Note the learn category does not include fact lookup or question answering. As described in Shahaf et al.,²⁶ maps are less useful for this type of query.

The “investigate” category corresponds to the higher levels of Bloom’s taxonomy, where users aim to produce outcomes. In it, users aim to transform existing data into useful patterns, seeking gaps in current knowledge. They analyze and synthesize different pieces

Figure 7. Detail from *The Lord of the Rings* map revealing the structure of the story; one can see how hobbits and Gandalf start their journey, gather together in Elrond’s council, and then split up, with callout bubbles as manual annotations.



of information, looking for plausible generalizations that could result in new insights. In particular, such users might be interested in contrasting and comparing multiple maps.

Interaction. Interaction is crucial to the success of metro maps. Users often know precisely what they want to find, but it is not easy for them to distill their ideas into a few keywords. Maps should thus allow interaction. Many models of interaction can be naturally integrated with metro maps. We rely on user feedback to learn preferences and adjust our maps accordingly. In the following sections, we discuss two interaction mechanisms we have implemented: “zooming” and “word feedback.”

Zooming. Some users are interested in a quick, high-level overview of a topic, while others wish to delve into the details. We thus want our maps to be zoomable. As maps are richly expressive, there are multiple ways to interpret zoom interactions. We have implemented three interpretations: zooming could affect time resolution; cluster resolution could make clusters split and merge; and users could focus on a particular metro line.

Word feedback. When a user interacts with a map, the map should ideally support feedback of the form “Tell me more about the E.U.’s reaction to the crisis” or “I am not interested in the Red Sox.” Labeling entire maps, or even single documents, is not rich enough to support this interaction model. Also, there is no way to indicate a user likes something not on the map.

We propose “feature-based feedback” instead to provide a natural way to support the queries discussed earlier; the user could increase the importance of the word “E.U.” and decrease the importance of “baseball” to achieve the desired effect. We use a discriminative semi-supervised learning method that incorporates such training affinities between features and classes.⁹ Using it, we define a personalized, session-sensitive notion of coverage that accounts for user feedback.

Related Work

To the best of our knowledge, automatic construction of metro maps is novel. Nevertheless, extensive work has been done on myriad related directions, from topic detection and tracking to summarization and temporal text mining.



Given a query, our algorithms generate concise structured sets of storylines that maximize coverage of salient pieces of information.



Our work differs from previous work in several important ways. First, our system has structured output, so not only does it pick nuggets of information, it explicitly shows connections among them. Prior work was limited largely to list-output models. In the summarization task,^{2,20,22} the goal is often to summarize a corpus of texts by extracting a list of sentences. Other methods^{18,31,30} discover new events but do not attempt to string them together.

Numerous efforts at information retrieval go beyond lists to provide richer views, including different notions of storylines.^{1,2,28,29} Graph representations are common across a variety of related problems,^{10,14,17,19} from topic evolution to news analysis. However, in all these methods there is no notion of graph paths as coherent storylines. Rather, graph edges might be selected because they pass a threshold or belong to a spanning tree.

Still other methods^{5,6} consider coherence at the path-level, in the sense that they aggregate a similarity score across all chain documents. However, they do not consider the order of the documents or the strength of the weakest link and may assign high coherence to chains despite bad transitions. Guaranteeing strong transitions across chains facilitates knowledge acquisition and comprehension.

Multiple tools exist for summarizing and visualizing literature; see Börner⁴ for a compendium. Unlike our clusters of documents, many of these systems^{12,16} use a single concept as a unit of analysis. This granularity is too fine to be useful by a non-expert. Other tools with granularity similar to ours often focus on visualizing citations or co-citations.^{8,13} Again, edges between documents are based on local computation, and there is no notion of coherent lines of research.

Finally, visual metaphors similar to metro maps have been used before to display abstract knowledge; for example, Nesbitt’s map shows interconnecting ideas running through his Ph.D. thesis.²¹ However, these maps were all constructed manually, as opposed to ours being generated automatically.

Conclusion

We have outlined our studies to date on methods that extract information and construct summarizing metro maps.

Given a query, our algorithms generate concise structured sets of storylines that maximize coverage of salient pieces of information. Most important, metro maps explicitly show the relationships between lines. We have applied metro maps to help people understand news stories, research areas, legal cases, and works of literature. We conducted promising pilot user studies over real-world datasets in several domains. The results suggest metro maps help users acquire knowledge more efficiently.

Our work also has several limitations that would be interesting to address in the future; for example, our notion of coherence assumes word repetition, so our system cannot handle extremely short documents (such as Twitter posts). Moreover, our shallow features make coherence metrics prefer chains of articles from the same source. We also plan to make our system more robust to noise; while the method is not very sensitive to the removal of a few articles from the dataset, near-duplicates affect coverage weights, biasing the algorithm toward covering them. Sensitivity to small changes in the query date range might be addressed by automatically finding an optimal segmentation of the timeline.

We also plan to experiment with richer forms of input, output, and interaction mechanisms and integrate higher-level semantic features. Much of the current work was devoted to crafting an objective function; in the future, we wish to learn or revise an objective function directly from user feedback. Another interesting direction is the point-of-view mechanism, letting users see a topic through the eyes of another person (such as a Democrat asking for the Republican point of view).

This line of work can lead to tools that help people navigate and understand ideas, trends, connections, and storylines amidst an information explosion.

Acknowledgments

We would like to thank Rok Sosic, Andrej Krevl, Dima Brezhnev, Caroline Suen, Jeff Jacobs, Heidi Wang, Thomas von der Ohe, Tom Camenzind, Rohan Puttagunta, and Raiyan Khan. This research has been supported in part by NSF IIS-1016909,

CNS-1010921, IIS-1149837, DARPA SMISC, DARPA GRAPHS, ARL AHP-CRC, Okawa Foundation, Docomo, Boeing, Volkswagen, Intel, the Brown Institute for Media Innovation, and the Alfred P. Sloan Fellowship. 

References

- Ahmed, A., Ho, Q., Eisenstein, J., Xing, E., Smola, A.J., and Teo, C.H. Unified analysis of streaming news. In *Proceedings of the 20th International Conference on the World Wide Web* (Hyderabad, India, Mar. 28–Apr. 1). ACM Press, New York, 2011.
- Allan, J., Gupta, R., and Khandelwal, V. Temporal summaries of new topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, LA, Sept. 9–13). ACM Press, New York, 2001.
- Bloom, B.S., Engelhart, M.D., Furst, E.J., and Hill, W.H., Eds. *Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain*. Longman, White Plains, NY, 1956.
- Borner, K. *Atlas of Science: Visualizing What We Know*. MIT Press, Cambridge, MA, 2010.
- Boyack, K.W. and Klavans, R. Creation of a highly detailed, dynamic, global model and map of science. *Journal of the Association for Information Science and Technology* 65, 4 (Apr. 2014), 670–685.
- Braam, R.R., Moed, H.F., and Van Raan, A.F. Mapping of science by combined co-citation and word analysis, I: Structural aspects. *Journal of the Association for Information Science and Technology* 42, 4 (May 1991), 233–251.
- Broder, A. A taxonomy of Web search. *ACM SIGIR Forum* 36, 2 (Sept. 2002), 3–10.
- Chen, C. Citespace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the Association for Information Science and Technology* 57, 3 (Feb. 2006), 359–377.
- Druck, G., Mann, G., and McCallum, A. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Singapore, July 20–24). ACM Press, New York, 2008.
- Faloutsos, C., McCurley, K.S., and Tomkins, A. Fast discovery of connection subgraphs. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Seattle, WA, Aug. 22–25). ACM Press, New York, 2004.
- Farrand, P., Hussain, F., and Hennessy, E. The efficacy of the ‘mind map’ study technique. *Medical Education* 36, 5 (May 2002), 426–431.
- Fox, E.A., Neves, F.D., Yu, X., Shen, R., Kim, S., and Fan, W. Exploring the computing literature with visualization and stepping stones and pathways. *Commun. ACM* 49, 4 (Apr. 2006), 52–58.
- Garfield, E. and Pudovkin, A.I. The histcite system for mapping and bibliometric analysis of the output of searches using the ISI Web of Knowledge. In *Proceedings of the 67th Annual Meeting of the American Society for Information Science and Technology* (Providence, RI, Nov. 12–17). Association for Information Science and Technology, Silver Spring, MD, 2004.
- Gillenwater, J., Kulesza, A., and Taskar, B. Discovering diverse and salient threads in document collections. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (Jeju Island, Korea, July 12–14). Association for Computational Linguistics, Stroudsburg, PA, 2012, 710–720.
- Hall, R.H. and O'Donnell, A. Cognitive and affective outcomes of learning from knowledge maps. *Contemporary Educational Psychology* 21, 1 (Jan. 1996), 94–101.
- Hossain, M.S., Gresock, J., Edmonds, Y., Helm, R., Potts, M., and Ramakrishnan, N. Connecting the dots between PubMed abstracts. *PLoS One* 7, 1 (Jan. 2012), e29509.
- Jo, Y., Hopcroft, J.E., and Lagoze, C. The web of topics: discovering the topology of topic evolution in a corpus. In *Proceedings of the 20th International Conference on the World Wide Web* (Hyderabad, India, Mar. 28–Apr. 1). ACM Press, New York, 2011, 257–266.
- Kleinberg, J. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery* 7, 4 (Oct. 2003), 373–397.
- Nallapati, R., Feng, A., Peng, F., and Allan, J. Event threading within news topics. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management* (Washington, D.C., Nov. 8–13). ACM Press, New York, 2004, 446–453.
- Nenkova, A. and McKeown, K. A survey of text summarization techniques. In *Mining Text Data*, C.C. Aggarwal and C. Zhai, Eds. Springer, 2012, 43–76.
- Nesbitt, K. Getting to more abstract places using the metro map metaphor. In *Proceedings of the Eighth International Conference on Information Visualisation* (London, U.K., July 14–16). IEEE, 2004, 488–493.
- Radev, D., Otterbacher, J., Winkel, A., and Blair-Goldensohn, S. Newsiness: Summarizing online news topics. *Commun. ACM* 48, 10 (Oct. 2005), 95–98.
- Rewey, K.L., Dansereau, D.F., and Peel, J.L. Knowledge maps and information processing strategies. *Contemporary Educational Psychology* 16, 3 (July 1991), 203–214.
- Shahaf, D. and Guestrin, C. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, D.C., July 25–28). ACM Press, New York, 2010.
- Shahaf, D., Guestrin, C., and Horvitz, E. Metro maps of science. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Beijing, China, Aug. 12–16). ACM Press, New York, 2012.
- Shahaf, D., Guestrin, C., and Horvitz, E. Trains of thought: Generating information maps. In *Proceedings of the 21st International Conference on the World Wide Web* (Lyon, France, Apr. 16–20). ACM Press, New York, 2012, 899–908.
- Shahaf, D., Yang, J., Suen, C., Jacobs, J., Wang, H., and Leskovec, J. Information cartography: Creating zoomable, large-scale maps of information. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Chicago, IL, Aug. 11–14). ACM Press, New York, 2013, 1097–1105.
- Swan, R. and Jensen, D. TimeMines: Constructing timelines with statistical models of word usage. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Boston, MA, Aug. 20–23). ACM Press, New York, 2000.
- Yan, R., Wan, X., Otterbacher, J., Kong, L., Li, X., and Zhang, Y. Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Beijing, China, July 24–28). ACM Press, New York, 2011, 745–754.
- Yang, Y., Ault, T., Pierce, T., and Lattimer, C. Improving text categorization methods for event tracking. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Athens, Greece, July 24–28). ACM Press, New York, 2000, 65–72.
- Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B., and Liu, X. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems* 14, 4 (July/Aug. 1999), 32–43.

Dafna Shahaf (dshahaf@cs.stanford.edu) is a postdoctoral fellow in the Computer Science Department at Stanford University, Stanford, CA.

Carlos Guestrin (guestrin@cs.washington.edu) is an associate professor in the Department of Computer Science & Engineering at the University of Washington, Seattle, WA.

Eric Horvitz (horvitz@microsoft.com) is a Distinguished Scientist and Director at Microsoft Research, Redmond, WA.

Jure Leskovec (jure@cs.stanford.edu) is an assistant professor in the Department of Computer Science at Stanford University, Stanford, CA.

© 2015 ACM 00010782/15/11 \$15.00



Watch the authors discuss their work in this exclusive Communications video.
<http://cacm.acm.org/videos/information-cartography>

DOI:10.1145/2739042

Explaining motivations for online comments, this study looks to help establish a positive, nonthreatening online comment culture.

BY SO-HYUN LEE AND HEE-WOONG KIM

Why People Post Benevolent and Malicious Comments Online

WITH THE PROLIFERATION of smart devices and mobile and social network environments, the social side effects of these technologies, including cyberbullying through malicious comments and rumors, have become more serious. Malicious online comments have emerged as an unwelcome social issue worldwide. In the U.S., a 12-year-old girl committed suicide after being targeted for cyberbullying in 2013.²⁰ In Singapore, 59.4% of students underwent at least some kind of cyberbullying, and 28.5% were the targets of nasty online comments in 2013.¹⁰ In Australia, Charlotte Dawson, who at one time hosted the “Next Top Model” TV program, committed suicide in 2012 after being targeted with malicious online comments. In Korea, where damage caused by malicious comments



is severe, more than 20% of Internet users, from teenagers to adults in their 50s, posted malicious comments in 2011.⁹

Recognizing the harm due to malicious comments, many concerned people have proposed anti-cyberbullying efforts to prevent it. In Europe, one such campaign was called The Big March, the world's first virtual global effort to establish a child's right to be safe from cyberbullying. The key motivation behind these campaigns is not just to stop the posting of malicious comments but also to motivate people to instead post benevolent comments online. Research in social networking has found benevolent comments online are not alone but coexist in cyberspace with many impulsive and illogical arguments, personal attacks, and slander.¹⁴ Such comments are not made in isolation but as part of attacks that amount to cyberbullying.

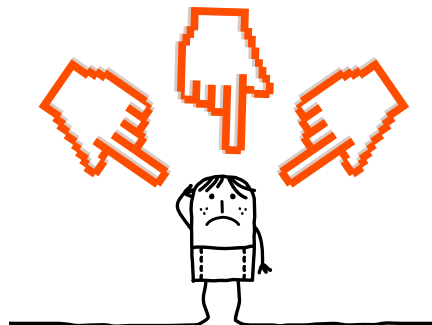
Both cyberbullying and malicious

» key insights

- People post benevolent comments to encourage and help others, feel self-satisfaction, and promote the social good; they post malicious comments to express anger, resolve feelings of dissatisfaction, and even have “fun.”
- One problem with online comments is lack of responsibility due to anonymity; one response might be ethical education for Internet users and use of real identities with stronger regulation.
- Developers could apply text filters to their systems to detect forbidden and malicious texts and post the identities of users most active in posting benevolent comments on their sites.

comments are increasingly viewed as a social problem due to their role in suicides and other real-world crimes. However, the online environment generally lacks a system of barriers to prevent privacy invasion, personal attacks, and cyberbullying, and the barriers that do exist are weak. Social violence as an online phenomenon is increasingly pervasive, a phenomenon manifesting itself through social divisiveness.

Research is needed to find ways to use otherwise socially divisive factors to promote social integration. However, most previous approaches to online comments have focused on analyzing them in terms of conceptual definition, current status, and cyberbullying that involves the writing of malicious comments.^{1,8,13,16,21,22} Still lacking is an understanding of why people post malicious comments in the first place or even why they likewise post benevolent



post comments. Here, we outline an approach toward shaping a healthier online environment with fewer malicious comments and more benevolent ones that promote social integration.

Methodology

As an exploratory study, we took an interview approach. Unlike previous studies where the research typically reflected the perspective of elementary, middle, or high school students, we included in-depth interviews with a broader range of age groups. Questions dealt with reasons for benevolent and malicious comments, problems associated with online comments, and suggestions for addressing the problems.

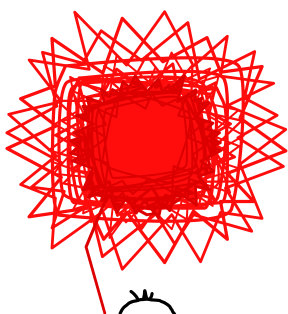
As a qualitative study, we adopted the convenience-sampling approach for selecting interviewees. For qualitative researchers, it is the relevance of interview subjects to the research topic rather than their representativeness that determines how they select participants.⁴ The interviewees should be able to explain the reasons or motivations for such postings. We thus checked whether interview subjects had posted comments online.

Our 110 interview subjects ranged from students in their teens to adults in their 50s. The number was determined by confirming theoretical saturation,¹⁸ indicating no additional relevant responses, or codes, emerged from additional data collection and analysis. By grouping 10 interview subjects at each stage, we were able to analyze interview data based on the coding at each stage. After conducting interviews

over 11 stages with the 110 subjects, we could no longer find new codes. For this reason, we limited ourselves to the 110 subjects, of whom three did not complete their interviews. We thus included 107 subjects in the analysis (see Table 1). The average interview time per participant was 30 to 40 minutes. We gave gift certificates for books to participants to encourage sincerity.

Using the open coding approach,¹⁹ we subjected transcripts of the interviews to content analysis, permitting inclusion of a large amount of textual information and systematic identification of its properties.⁷ Coding was performed by two researchers, one of whom, to avoid potential bias, was not involved in data collection. With open coding, each coder examined the interview transcripts line by line to identify codes within the textual data. We then grouped the codes into categories.

The inter-rater agreement scores for the entire coding process averaged 0.79, with Cohen's Kappa scores averaging 0.78, indicating an acceptable level of inter-rater reliability.⁹ Inter-rater disagreements were reconciled through discussion between the two raters. We then grouped the identified codes into broader categories that reflected commonalities among the codes. We further counted the frequency of relevant responses, or codes, for each category.



comments that promote social integration. Unlike previous studies that focused on cyberbullying itself as a socially divisive phenomenon, this study, which we conducted in Korea in 2014, involved in-depth interviews with social media users in regard to both malicious and benevolent comments. To combat the impropriety represented by the culture of malicious comments and attacks, our study sought to highlight the problem of malicious comments based on the reasons people

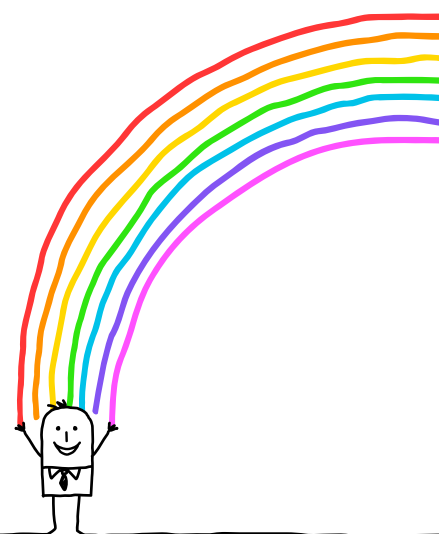


Table 1. Demographics of interview participants.

Demographic Variables		Frequency (%)
Gender	Male	49 (45.8%)
	Female	58 (54.2%)
Age (years)	<19	11 (10.3%)
	20-29	41 (38.4%)
	30-39	35 (32.7%)
	40-49	10 (9.3%)
	>50	10 (9.3%)
Job	Employed	30 (28.0%)
	Professional	24 (22.4%)
	Student	40 (37.4%)
	Homemaker	7 (6.6%)
	Other	6 (5.6%)
Total		107 (100%)

Table 2. Reasons for benevolent comments.

Reason	Example Responses	Frequency (%)
Encouragement	To console someone, encourage someone, praise someone, support someone	49 (36.3%)
Self-satisfaction	For one's satisfaction, one's sense of well-being	29 (21.5%)
Providing advice or help	To help someone, due to belief that one's faith or opinion can influence others	16 (11.8%)
Supporting other comments	Agreeing with or sympathizing with others' benevolent comments	12 (8.9%)
Actualizing society's good	To spread benevolent comments, prevent malicious comments, protect those receiving malicious comments	10 (7.4%)
Online context norm	Mob psychology, sense of unity	7 (5.2%)
Information sharing and communication	To communicate with others, share information with others, inform of one's praiseworthy behavior	7 (5.2%)
Others	People are essentially good-hearted, want to maintain offline relationships, want to receive a gift certificate	5 (3.7%)
Total		135 (100%)

Results

Table 2 outlines the reasons for posting benevolent comments online. Five of the seven main ones accounted for 85.4% of the total: encouragement (36.3%); self-satisfaction (21.5%); providing advice or help (11.8%); supporting other benevolent comments (8.9%); and actualizing the social good (7.4%). Many respondents said they write benevolent comments to “encourage” or give hope or courage to someone and think such an attitude can yield positive change. People also post benevolent comments to “provide advice and help” to others. Ranked next as the main reasons for posting benevolent comments were “support for other benevolent comments” and “actualizing

society's good.” They revealed themselves by agreeing with others' benevolent comments, following others in the selected online context (“online context norm”), or preventing malicious comments and spreading benevolent comments. Posting benevolent comments based on such reasons made the people doing the posting feel “satisfaction,” motivating them to continue to post further benevolent comments.

Table 3 outlines the reasons for posting malicious comments. Five of the seven main ones accounted for 85.0% of the total: resolving a feeling of dissatisfaction (28.1%); hostility (20.3%); low self-control (15.7%); supporting other malicious comments (8.9%); and fun (8.5%). Many interview

subjects said people write malicious comments to express “anger and feelings of inferiority” and to get attention. “Hostility toward others” suggests malicious commenters attack and slander a certain person's blog or bulletin board and spread groundless rumors elsewhere. “Low self-control” suggests people post malicious comments irresponsibly for “fun” and to release “stress.” “Supporting other malicious comments” and “online context norm,” or following the general pattern of the selected online context, demonstrate how often people post malicious comments by simply following others.

The figure here outlines the identified problems with online comments (“anonymity” ranked highest at 42.8%) and suggestions for addressing them. Writing comments online allows people to participate without a certification process, creating an environment in which they can criticize, curse, and malign others in the course of expressing their opinions. “Lack of responsibility” was cited by 30.3% of interview subjects. Most people posting malicious comments do not appreciate the potential seriousness of their effect or actually belong to a category involving some kind of violence; they impulsively write comments without a sense of responsibility for their effect. Moreover, “online context climate,” or the environmental conditions of the online context, represented 11.0% of the interview subjects, implying that as people become less caring and more disdainful of others, their internal dissatisfaction can be expressed through malicious comments. There also appears to be a “lack of regulation and punishment” for malicious comments and a rise of “commercial selfishness” (such as programs that trigger witch hunts and prompt advertising by commenters).

Among suggestions from interview subjects on how to deal with malicious online comments, improvement of awareness through “educational programs and campaigns” ranked highest, with 39.7%, meaning organized education for Internet users was seen as a way to ensure a civil public dialogue. A more proactive type of Internet ethical education targeting teenagers should be a priority in schools, as well as at

home, to teach commenters to appreciate the seriousness of malicious comments and the potential for cyber violence. Such programs should highlight Internet users' responsibility for their online comments. They should also promote the idea of writing benevolent comments as a way to limit malicious comments and increase benevolent comments. Regarding the seriousness of anonymity in online comments, the "use of real identity" ranked next, with 29.3%. Many interview subjects suggested using real identities (such as real names or photos) as a way to reduce malicious comments in the online context.

Another suggestion from interview subjects was "more and stronger regulation," with 20.8%. Enforcing official punishment may be difficult online, but inadequate punishment is one of the reasons for malicious comments. In contrast, many interview subjects endorsed the idea of stronger regulation and legal punishment for malicious comments. Yet another sugges-

tion was "role of management," with 6.4%. Many interview subjects highlighted the role of managers of social media providers in monitoring and deterring online comments, especially malicious ones. Social media management should have a role in resolving how to reduce malicious comments and promote benevolent comments in a social media context; for example, a management team might develop an algorithm for filtering online comments and identifying malicious writers, then implement a supporting system. Other suggestions included "a reward program for reporting malicious comments," "clarifying the

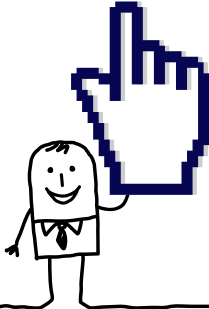
range of privacy," "abolishing any program that triggers a witch hunt," and "managing the culture of the online community." Despite the value of our findings, it would be useful to further test their robustness by replicating the study in countries other than Korea, in light of cultural differences.

Discussion

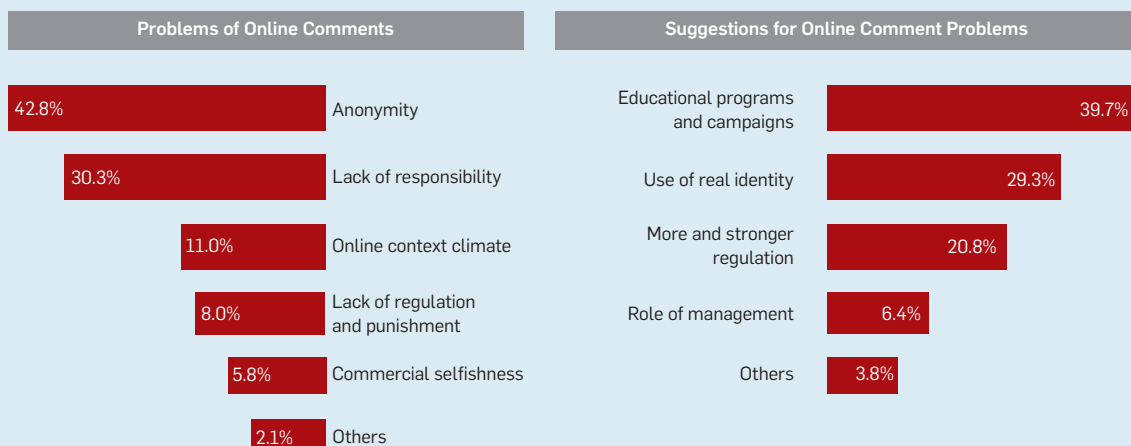
Motivations for malicious comments identified in the study involved targeting people's mistakes. Conversely, most benevolent comments involved encouragement and compliments to help people in difficult or risky situations, showing malicious comments

Table 3. Reasons for malicious comments.

Reason	Example Responses	Frequency (%)
Resolving a feeling of dissatisfaction	To express one's dissatisfaction, show a feeling of inferiority or frustration, desire to attract others' attention, jealousy	43 (28.1%)
Hostility toward others	To hurt others, criticize others, show hatred toward someone	31 (20.3%)
Poor self-control	Thoughtlessness, lack of responsibility, having a weak personality	24 (15.7%)
Supporting other comments	Agreeing that blame is deserved, agreeing with others' malicious comments	19 (12.4%)
Fun	Just for fun, enjoyment	13 (8.5%)
Releasing stress or alleviating tension	To release stress, express anger generated offline	10 (6.5%)
Online context norm	A general atmosphere of writing malicious comments, a social atmosphere where people cannot trust one another	8 (5.2%)
Others	Differences in outlook or perspective, stimulating content of articles, ignorance of content, judgmental	5 (3.3%)
Total		153 (100%)



Problems and suggestions for online comments.





is a primary reason for degradation of online social networks. Moreover, the abolition of anonymity and intensification of punishment in social media can be effective in reducing malicious comments and rumors. However, potential violation of freedom of expression also risks trivializing the online social network itself. Because anonymous forms of freedom of expression have always been controversial in theoretical and normative spheres of social research,^{5,6,11,12,17} careful consideration of any limiting of comments is necessary before a ban might be contemplated.

To establish an environment of healthy online commentary, a quick measure is needed of the damage caused by the spread of false or toxic information and formation of socioeconomic support for the potential victims. Measures are required to minimize damage when toxic information is posted in social media and investigate legal steps that might be needed to pursue awards for financial damage. Besides imposing legal punishment on people posting malicious comments, victims' mental pain is much more urgent. A well-organized legal support program for such damage is necessary.

Also needed are ways to enforce social media users' right to control their personal information, as well as to verify distorted information. Such systems would involve monitoring to detect distorted information, services for filtering information, an "information confirmation system," and laws supporting the rights of users concerning their consent and choices in how their personal information is used. Regarding malicious comments, Internet portal sites should suggest preventive measures (such as systems to report malicious comments, disclose commenters' IP addresses, and create lists of prohibited words).

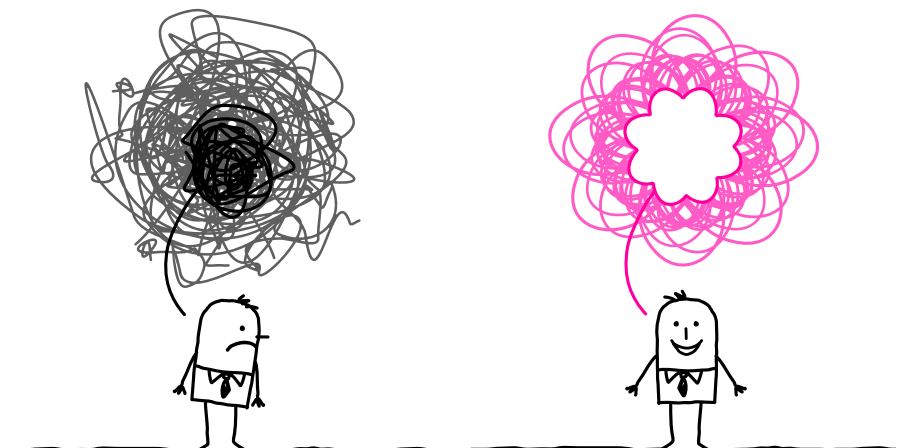
Education in socially appropriate and legal use of social media is necessary to minimize the social, cultural, and economic gaps among approaches and applications prevalent today. Education along these lines should emphasize not only the role of the producer but also of information users and instill a sense of responsibility for information. Such an effort could be accompanied by encouraging benevolent comments. Educational programs and campaigns could also be directed at motivating the posting of benevolent comments. Our study found Internet users post benevolent comments mainly to encourage and help others, often making them because other Internet users have already done so, and gain a sense of satisfaction from their action. Efforts to develop social norms of posting benevolent comments should also consider the reasons identified here for positive posts. People, especially teens, tend to take collective action in the use of social media.¹⁵ Our study further found that people post benevolent comments and malicious comments due to the online context, as in Table 2 and Table 3. It is therefore important for all online sites that accept comments to develop social norms of posting benevolent comments through educational programs and campaigns.

Conclusion

Unrestricted by time and space, online communication has led to increasing numbers of both benevolent and malicious comments, with the latter including impromptu and irrational personal abuse and defama-

tion. Because malicious comments can provoke pain and even violence in cyberspace, they have emerged as a serious social issue, including allegedly causing their targets to commit suicide. To combat the abuse represented by the culture of malicious comments and attacks, our study investigated their sources and role in social disintegration. Our study also suggested ways to address the problem and increase the number of benevolent comments that can contribute to social integration and harmony.

The study has several implications for research, as it was among the first to comprehensively consider the reasons for posting comments, identify related problems, and explain how to address them. Previous research explored general reasons (such as to redirect feelings, revenge, jealousy, and boredom) for cyberbullying based on data collected from high school students,²² but there was a lack of understanding among researchers as to the motivators that lead to posting malicious comments and benevolent comments, respectively. Our study thus adds value to the literature by explaining the reasons for malicious and benevolent comments and how they differ. Although previous research classified types of cyberbullying and investigated the consequences of cyberbullying,^{10,16} missing was an understanding of the problems related to online comments in general. This study thus adds value to the literature by identifying the relevant problems and advancing our understanding of the phenomenon. Meanwhile, several studies have discussed strategies



school guidance counselors and parents might use to prevent cyberbullying and proposed coping strategies for students.^{2,16} Extending the previous research, we have contributed by explaining how to manage the problems of online comments as a way to reduce malicious ones and promote benevolent ones.

Our results also suggest how social media service providers, educational institutions, and government policy-makers might be able to establish a positive, nonthreatening online comment culture. Educators must understand why students post malicious, as well as benevolent, comments. They can consider updating their curricula



or teaching content by adding ethical issues related to online comments. That is, based on the reasons we identified, they can help guide their students (such as for self-satisfaction and society's advancement through benevolent comments), what to post (such as support), and how to post (such as self-expression). They can likewise also teach why not to post (such as social problems), what not to post (such as rage), and how not to post (such as poor self-control). Students especially should be educated in a way that instills a sense of responsibility for their postings. If they come to have a sense of responsibility and perceive that posting malicious comments is a form of violence, cyberbullying would likely be reduced. Educators might also consider launching campaigns that promote the posting of benevolent comments, establishing social norms of conduct that would reach many other people online and be accepted by them.

Policymakers should understand government regulations and corre-

sponding legal punishment can be useful in regulating cyberbullying, especially in the form of malicious online comments. Our results further suggest cyberbullying, especially through such comments, should be regulated at the government level. Many people also believe legal penalties for posting malicious comments should be strengthened. Both regulation and punishment have a role to play in reducing and even preventing malicious online comments.

For social media service providers, including information systems developers, our results suggest they should consider requiring real identities for postings. When people access social media and post comments anonymously, they think less about what they post. Requiring true identities would cause them to be more careful and responsible. Our results also suggest providers of social media services can apply text filters to their systems. Because certain texts are used repeatedly in cyberbullying or malicious comments, providers should be persuaded to develop a system to detect certain texts and alert them as to when to possibly take action against the people posting them. Such a filtering function could reduce the number of all kinds of malicious comments. Conversely, social media service providers should consider posting lists that rank users most active in posting benevolent comments on their sites. Because people generally enjoy self-expression, these rankings could motivate more people to post positive comments as a way to develop a new social norm in which malicious comments are unwelcome and the people posting them are scorned.

Acknowledgment

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2012S1A3A2033291). 

References

1. Arıcak, T., Siyahhan, S., Uzunhasanoglu, A., Sarıbeyoglu, S., Ciplak, S., Yilmaz, N. and Memmedov, C. Cyberbullying among Turkish adolescents. *Cyberpsychology and Behavior* 11, 3 (2008), 253–261.
2. Bhat, C.S. Cyber bullying: Overview and strategies for school counsellors, guidance officers, and all school personnel. *Australian Journal of Guidance and Counselling* 18, 1 (2008), 53–66.
3. Fleiss, J.L. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, Inc., New York, 1981.

4. Flick, U. *An Introduction to Qualitative Research*. Sage, Thousand Oaks, CA, 1998.
5. Froomkin, M. Flood control on the information ocean: Living with anonymity, digital cash, and distributed databases. *Journal of Law and Commerce* 15, 2 (1996), 395–453.
6. Froomkin, M. Legal issues in anonymity and pseudonymity. *The Information Society* 15, 2 (1999), 113–127.
7. Holder, I. *The Interpretation of Documents and Material Culture*. Sage, Thousand Oaks, CA, 1994.
8. Kim, S.K. A study on the restriction of freedom of expression on the Internet. *The Journal of Comparative Law* 11, 3 (2011), 47–72.
9. Korea Internet and Security Agency. *Internet Ethical Culture Research*, Seoul, Korea (2012); <http://isis.kisa.or.kr/board/index.jsp?pageId=040100&bbsId=7&itemId=786>
10. Kwan, G.C.E. and Skoric, M.M. Facebook bullying: An extension of battles in school. *Computers in Human Behavior* 29, 1 (2013), 16–25.
11. Long, G.P. Who are you? Identity and anonymity in cyberspace. *University of Pittsburgh Law Review* 55, (1993), 1177–1213.
12. Marsh, T.D. In Defense of Anonymity of the Internet. *Res Gestae* (Apr. 2007), 24–32.
13. Park, H.J. A critical study on the introduction of the cyber contempt. *Anam Law Review* 28, 1 (2009), 315–347; http://kiss.kstudy.com/journal/thesis_name.asp?tname=kiss2002&key=2751961
14. Poster, M. *Cyber Democracy: Internet and Public Sphere*. University of California, Irvine (1995); <http://www.hnet.uci.edu/mposter/writings/democr.html>
15. Seo, H., Houston, J.B., Knight, L.A.T., Kennedy, E.J., and English, A.B. Teens' social media use and collective action. *New Media & Society* 16, 6 (2014), 883–902.
16. Slonje, R., Smith, P.K., and Frisen, A. The Nature of cyberbullying and strategies for prevention. *Computers in Human Behavior* 29, 1 (2013), 26–32.
17. Stieglitz, E.J. Anonymity of the Internet: How does it work, who needs it, and what are its policy implications? *Cardozo Arts and Entertainment Law Journal* 24, 3 (2007), 1395–1424.
18. Strauss, A. and Corbin, J. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage, Newbury Park, CA, 1990.
19. Strauss, A. and Corbin, J. *Basics of Qualitative Coding*. Sage, Thousand Oaks, CA, 1998.
20. *The Guardian*. Florida cyberbullying: Girls arrested after suicide of Rebecca Sedwick, 12. *The Guardian* (Oct. 15, 2013); <http://www.theguardian.com/world/2013/oct/15/florida-cyberbullying-rebecca-sedwick-two-girls-arrested>
21. Vandeboosch, H. and Cleemput, K.V. Defining cyberbullying: Qualitative research into the perceptions of youngsters. *Cyberpsychology and Behavior* 11, 4 (2008), 499–503.
22. Varjas, K., Talley, J., Meyers, J., Parris, L., and Cutts, H. High school students' perceptions of motivations for cyberbullying: An exploratory study. *Original Research* 11, 3 (2010), 269–273.

So-Hyun Lee (sohyun1010@yonsei.ac.kr) is a Ph.D. candidate in the Graduate School of Information at Yonsei University, Seoul, Korea.

Hee-Woong Kim (kimhw@yonsei.ac.kr), the corresponding author, is the Underwood Distinguished Professor in the Graduate School of Information at Yonsei University, Seoul, Korea.

© 2015 ACM 00010782/15/11 \$15.00



Watch the authors discuss their work in this exclusive *Communications* video. <http://cacm.acm.org/videos/why-people-post-benevolent-and-malicious-comments-online>

DOI:10.1145/2735625

Sentiment analysis helps detect NBA players' pre-game moods from their tweets and predict their on-court performance.

BY CHENYAN XU, YANG YU, AND CHUN-KEUNG HOI

Hidden In-Game Intelligence in NBA Players' Tweets

THE SPORTS INDUSTRY is big business globally and domestically;¹⁰ for example, the New York Knicks of the National Basketball Association (NBA) generated \$287 million in revenue in 2013.² In order for sports organizations to maximize their financial performance, they must win on the court. The operational staffs, including coaches and general managers, must consistently make the right decisions despite many constraints, including a league-imposed salary cap and team budgets. Sports analytics plays an increasingly important role in such decisions.

Sports analytics traditionally involves statistical techniques for analyzing historical player performance. General managers have used it to build their rosters and coaches have used it in conjunction with their domain knowledge to adjust lineups and improve

players' on-court performance. Though ongoing sports analytics research and practices center mostly on the structured data of player profiles and historical performance,¹ this article explores the extent NBA teams can use "unstructured" social media data to further their sports analytics efforts. This novel focus is motivated by the prevalence of social media analytics in all kinds of business domains over the past five years. Specifically, our objective is to show how NBA players' pre-game emotional state, as captured through their tweets, or the messages they post on Twitter, before a game can help predict on-court performance in the game.

The framework we propose here can be used to inform decisions regarding game-day player assessment, lineup changes, and on-court strategy. Given NBA teams' longstanding positive attitude toward and practice of sports analytics, the league is an appropriate context for illustrating the new role we envision for sports analytics.

We begin by exploring the widespread use of Twitter among NBA players, then how our framework could benefit sports analysts in the NBA and their potential motivation for adopting the approach. This discussion includes justification for why the framework and potential future relevant work could interest analysts and motivate software developers to add processing of unstructured data (such as social media content) to their commercial products. We next detail how social media analytics can be incorporated into analysts' decision-making processes and follow with a case study

» key insights

- **Sentiment analysis, an emerging text-mining technique, can help analyze the tweets of NBA players.**
- **Coaches, general managers, and other staff of NBA teams can use it to discern and address players' moods before games.**
- **Our study shows players' before-game mood, as captured through the sentiment they reveal in their tweets, is positively associated with their on-court performance.**



implementing the core techniques of the framework.

Twitter and NBA Players

The past few years have seen an explosion of tweets by NBA players. With the unique features of Twitter compared to other social media applications and its worldwide user base, NBA players have used this emerging technology to communicate with fans, journalists, peers, friends, and others. They actively voice their opinions, thoughts, and feelings through short, direct messages of 140 characters or less on Twitter. According to data from <http://Tweeting-Athletes.com> and [Reference.com, 353 active players, or approximately 78.4% of the total, had Twitter accounts during the 2012–2013 season, posting a total of 91,659 tweets.](http://www.Basketball-</p>
</div>
<div data-bbox=)

Table 1 includes several tweets by NBA players expressing their views and reactions to major NBA-related events. NBA players' spontaneous use of Twitter renders Twitter a natural environment for spreading information. Players' tweets thus provide insight into their unfiltered thoughts that may otherwise be unavailable. The tweets in Table 2 show players' use of Twitter is not as uniform as their use of conventional media, producing unpredictable consequences for the league, teams,

and the players themselves.

Along with the widespread use of Twitter by NBA players, what effect does the large number of players' brief messages have on their teams? Compared to most non-sports organizations, which frown on employees' using social networking sites for personal purposes, the NBA and its teams are even less friendly toward players' use of Twitter, concerned it might distract from winning games;⁹ for example, in 2009, the league established a policy saying coaches and players are not allowed to use social networking sites beginning 45 minutes before game time until after post-game interviews.¹⁷ The

Table 1. Examples of NBA players using Twitter to react to major NBA-related events.

Date	Exemplar Tweets	Tweet Background
Dec. 2011	<i>After the recent news about the possible end of the #lockout, we're all excited to see the light at the end of the tunnel!! #WeLoveThisGame</i> — Pau Gasol	The second-longest NBA lockout in history came to an end.* A lot of players tweeted their excitement that the new season is finally coming.
May 2013	<i>Congratulations to Jason - society couldn't hope for a more eloquent and positive role model: buff.ly/100vOMd</i> — Steve Nash <i>Jason's s-xuality doesn't change the fact that he is a great friend and was a great teammate</i> — Jason Kidd	On May 6, 2013, NBA center Jason Collins announced he is a gay, making him the first openly gay player in the league. He then received tremendous support from other players on Twitter.
Apr. 2014	<i>I couldn't play for him.</i> — Kobe Bryant <i>All this sterling outrage speaks to how naive and gullible folks have become about race</i> — David West	On April 25, 2014, TMZ released an audio clip of Los Angeles Clippers owner Donald Sterling's racist comments on minorities. He was then severely criticized by NBA players on Twitter expressing their disappointment and anger toward him and calling for an immediate investigation.

Note:

* The 161-day lockout began July 1, 2011 and ended December 8, 2011. It delayed the start of the 2011–2012 regular season from November 1 to December 25 and reduced the regular season from 82 to 66 games.

Table 2. Examples of NBA players' reckless use of Twitter.

Date	Exemplar Tweets	Tweet Background	Aftermath
Mar. 2009	<i>In da locker room, snuck to post my twitt. We're playing the Celtics, tie ball game at da half. Coach wants more toughness. I gotta step up.</i> — Charlie Villanueva	Milwaukee Bucks forward Charlie Villanueva posted a tweet during halftime of a home game against the Boston Celtics.	It not only caused the media to criticize Villanueva for not paying attention during the game but also the league to establish a policy preventing players, coaches, and officials from using Twitter and other social networks during games.
Jun. 2009	<i>Today is a sad day...Kevin McHale will NOT be back as head coach this season</i> — Kevin Love	Minnesota Timberwolves forward Kevin Love disclosed on Twitter header coach Kevin McHale would not return for the new season and he was sad about it.	For the first time, such breaking news was made public before a conventional media source reported it. The tweet also put Love in an awkward position publicly.
Dec. 2012	<i>Somebody tell serg Abaka. He aint bout dis life. Next time he run up on me im goin in his mouth. That's a promise. He doin 2 much.</i> — Stephen Jackson	San Antonio Spurs swingman Stephen Jackson posted a threatening tweet to another player.	He was later fined \$25,000.*

Note:

* Stephen Jackson's penalty was just the tip of the iceberg, where many of NBA players get fined due to inappropriate Twitter messages.

Miami Heat, Toronto Raptors, Milwaukee Bucks, Los Angeles Clippers, and other teams have expanded the ban to “team times,” including practices and meetings.¹⁷ Still, players are avid users of Twitter, and the number of tweets they send shows no sign of decreasing.

Here, we suggest NBA teams consider the value of players' tweets through a sports analytics perspective. When governments can listen to, analyze, and understand their citizens' opinions through such channels as social networking sites and bulletin-board

systems and use it in their administrative decision-making processes,⁶ NBA coaches and general managers can use players' tweets to identify player moods before games and their potential effect on the court. To this end, our proposed framework offers a solution.

The Framework as a Solution

Baskerville and Myers³ wrote that in knowledge-creation networks, motivations and qualifications are the paramount issues to be considered when one domain consumes ideas

produced by another domain. Since social media analytics is a novel creation combining information systems and computer science, sports analysts should welcome our framework, which is grounded in social media analytics, only if they are motivated and the framework itself demonstrates enough practical benefit.

Motivations. Analysts must first understand the demand shaping the markets likely to adopt the framework. First, organizations in non-sports domains have benefited from

social media. The value of social media is presumably transferrable to sports organizations like the NBA and should align with their interest in achieving superior on-court, as well as related business, performance. Anecdotes throughout the business and statistical literature cite how organizations of all types take advantage of social media to gain a competitive edge. Social media can be used as an R&D tool to collect customer thoughts and feedback for product and service innovation and improvement, branding and sales, and customer support.⁵ While an increasing number of organizations have begun to deploy social media as an operational tool for managing employees, facilitating knowledge-management systems, and enhancing cross-functional collaboration,¹³ only a few sources report NBA teams have explored social media for operational purposes. Such dormancy is a surprise, given NBA teams demonstrate the same enthusiasm for social media as every other organization; all NBA teams have created Twitter accounts and actively use them year-round to interact with fans, aiming to fill seats and boost TV ratings. In this sense, the framework, which helps observe, understand, and manage player performance, should attract teams' interest. On the other hand, in 2011, Alamar and Mehrotra¹ wrote "... the most significant structural barrier to the growth of sports analytics is not only the absence of a clear doorway for teams ... but also the lack a clear process for developing the skills needed to open that doorway."¹ Though preliminary, the framework we propose here could serve as a good starting point for NBA teams use of social media as an operational tool within their own organizations.

Given the popularity of sports analytics among NBA teams, social media analytics should also motivate them to adopt these methods. NBA team managements have long been disciples of sports analytics. Before sports analytics, domain experts, including scouts, coaches, and general managers, were the sole members of sports teams qualified (and authorized) to make sports decisions.¹⁶ However, with data from through a variety of conventional and multimedia sources, domain experts



The spontaneous use of Twitter among NBA players raises the issue of whether these players even consider the consequences of their posted messages.



can quickly be overwhelmed processing a growing mountain of data. Moreover, they are less capable than statistical analysts at identifying the hidden patterns and trends behind the data. As a result, an increasing number of sports organizations turn to specialized analysts with advanced statistical know-how to make sense of it, leading to the current rise of sports analytics.

Sports analytics is well regarded; for example, the Player Efficiency Rating, devised by John Hollinger, a former ESPN writer and analyst, is a widely used indicator of NBA players' overall performance. Many NBA teams now have statisticians on their payrolls; for example, the Dallas Mavericks have worked with Wayne Winston, a professor of decision science at Indiana University, assisting coaching staff with on- and off-court decisions. Since 2010, a new breed of management has become more prominent in the NBA, and an increasing number of statisticians, including John Hollinger of the Memphis Grizzlies and Darly Morey of the Houston Rockets have begun to assume key roles in the front office. Some NBA teams contract with third-party companies for specialized analytics services, as between, say, the Portland Trailblazers and Citizen Sports in the league's player draft. Pursuing the potential benefit available through sports analytics, NBA teams are active proponents and participants in the annual MIT Sloan Sports Analytics Conference (<http://www.sloansportsconference.com>), looking to promote analytics tools for better on-court team performance.

While conventional sports analytics focuses primarily on structured data, we propose a framework that uses social media analytics to process unstructured textual data in sports analytics. Given widespread recognition of existing sports analytics, NBA teams should welcome the enhancement.

The data needed by the framework is easily obtained, further encouraging NBA teams to implement the framework. Sports analytics relies on the availability of raw data.¹⁰ Basketball aside, sports analytics is most widely applied in professional golf (<http://www.pga.com/home> and <http://www.lpga.com>) and Major League Baseball (<http://mlb.mlb.com/home>), given the ocean of data available through mul-

tiple sources. The proposed framework needs three sets of data: player/game profile (such as age, position, salary, and home/away game); player performance (such as field goals and percentage, plus/minus (+/-); and player tweets. The first two are available within NBA teams, as well as through outside databases compiled by third parties, including Basketball-Reference.com, ESPN (<http://www.espn.com>), and 82games.com (<http://www.82games.com>). Player tweets are easily viewed on Twitter, and analysts can use Web crawlers to gather data.

The social media analytics techniques needed by the framework are common features of existing commercial and open source software. Social media analytics includes a pool of established modeling and analytics techniques. The framework uses sentiment analysis to discern player mood from player tweets and regression, a type of trend analysis, to identify how that mood is associated with on-court performance. Both techniques are incorporated into existing software, including SAS Sentiment Analysis, SAS Enterprise Miner, and R. These resources free NBA teams from the technical detail of the framework. This low entry barrier to implementation could thus be further motivation for NBA teams to try it.

The framework's concern with identification of players' moods from their tweets has practical implications for three reasons: given the robust association between mood and performance in sports and psychology, NBA teams should want to discern player mood before games; mood adjustment has been an important part of team operation, especially player preparation; and incumbent methods for identifying player mood before games are limited. The framework thus represents a new way to manage player mood.

The consensus in the sports and the psychology literature is that mood affects athletes' competitive performance in which they must closely anticipate and observe their opponents' actions so they instinctively know how to respond. While positive mood helps athletes stay focused and vigilant, stimulating greater effort, motivation, and persistence, negative mood could be a major source of interference that distracts their attention.^{12,18}



Players are heavy users of Twitter in that they update themselves frequently, providing much useful data for social media analytics.

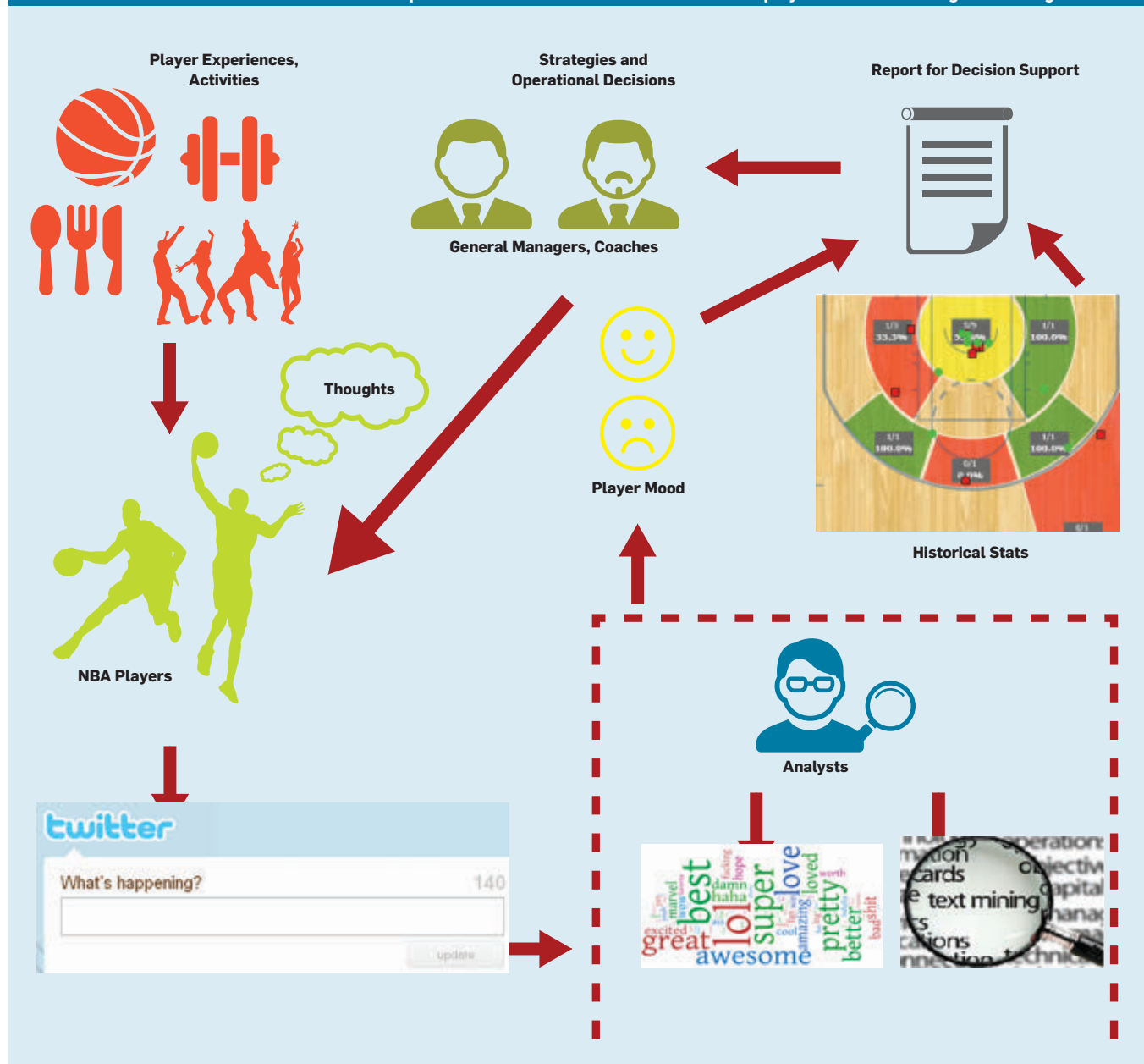


NBA teams recognize the importance of tuning player mood before games, taking various approaches to offset negative mood or generate positive mood; for example, when the Indiana Pacers' center Roy Hibbert struggled during the 2013–2014 NBA playoffs, his teammate Paul George invited him on a fishing trip to help him relax and regain his confidence. This apparently paid off in the form of Hibbert's 28 points and nine rebounds in the next game, as well as in the Pacers' victory over the Washington Wizards. At the same time, following a scandal involving Los Angeles Clippers' owner Donald Sterling, there was a call through mainstream media for Clippers fans and players to boycott game five of the Clippers playoff series against the Golden State Warriors. Feeling his players were being distracted by related media coverage, Clippers coach Doc Rivers canceled an upcoming practice, giving his players a day off to refocus. Other anecdotes reflect how NBA teams work closely with psychologists who provide counseling services to address player mood before critical games.

Effectiveness and efficiency aside, these approaches cannot succeed without first discerning player mood. However, NBA teams cannot follow their players 24 hours a day, especially when off the court. Distributing a questionnaire designed to measure mood is not feasible, as it is time-consuming for professional athletes to complete and cannot be used to test mood before every game. As mentioned earlier, NBA players are active Twitter users, a stream of tweets that represents an enormous, unfiltered opportunity to discern their mood. In this sense, the framework could optimize NBA teams' ongoing process of adjusting player mood by helping them discern mood more effectively.

Qualifications. The framework informs sports analytics in two main ways. First, it is rooted in social media analytics, a mature approach to processing social media data, developing and evaluating informatics tools and frameworks to collect, monitor, analyze, summarize, and visualize social media data to extract useful yet hidden patterns and intelligence.⁷ The rise of social media analytics could be attrib-

Framework of our recommended administrative process that listens to and transforms NBA players' tweets into in-game intelligence.



uted to three forces: growth of social media, with popular applications (such as Facebook, Twitter, and LinkedIn) spawning vast amounts of user-generated content; organizations recognizing the value of user-generated content, as well as other social media data, using it for competitive advantage; and business intelligence and analytics techniques and tools enabling collection, management, analysis, and presentation of social media data.

Social media analytics involves multiple informatics techniques, including sentiment analysis, or opinion mining, topic modeling, social network analysis, trend analysis, and visual analytics.

All play an important role in the various phases of social media analytics and can work together to help decision makers extract intelligence from the social network data stream. Due to the maturity of these tools, social media analytics has been applied to explain, detect, and predict disease outbreaks, election results, macroeconomic processes (such as crime detection), movie box office performance, natural phenomena (such as earthquakes), product sales, and financial markets (such as stock price); see, for example, Feldman,⁸ Kalampokis,¹¹ and Yu et al.²¹ Rui et al.¹⁵ drew on sentiment analysis, a technique that aims to distill mood

underlying textual data, finding overall customer mood from tweets with regard to a movie has a direct effect on ticket sales. Along with the virtues and robustness of social media analytics, our framework can benefit NBA team management and players alike.

NBA players' tweets represent high-quality text data for social media analytics because players use Twitter with spontaneous and genuine intent. Earlier research found athletes use social network sites to show affection and generally express themselves. Unlike conventional media, Twitter's natural communications environment ensures NBA players' unadulterated

perspective on events and topics is expressed immediately. Moreover, the spontaneous use of Twitter among NBA players raises the issue of whether these players even consider the consequences^a of their posted messages.²⁰ This problem is serious enough some ESPN analysts have suggested “maybe next time he’ll think before he Tweets,” motivating NBA teams and the league to institute regulations to curb “lack of thought.”²⁰ Nevertheless, there is sufficient evidence to indicate the tweets communicated by NBA players are truly reflective of their opinions, thoughts, and feelings. As a result, Twitter could indeed be a credible channel for social media analytics to discern player mood.

Applying Sentiment Analysis

Determining and extracting a player’s mood from his tweets is not straightforward, as tweets include unstructured text and information. Tools are needed to analyze a sea of textual data to distill the hidden intelligence. The framework we recommend through which NBA teams can add information in players’ tweets to administrative decision making (see the figure here) involves analyzing players’ public tweets. The teams could thus know players’ mental status (such as negative moods) and introduce necessary interventions to address potentially less-than-optimal on-court performance beforehand. In the framework, NBA players post tweets as reactions to a number of things, including games, coaches’ on- and off-court decisions, or, more frequently, life encounters. At the core of the approach is sentiment analysis that could help sift through players’ tweets and identify their moods before games. The results are then summarized as a report to be presented to team management, which can then address players’ negative moods or look to inspire positive moods.

The distilled mood from players’ tweets is admittedly not free of bias

due to a number of factors. First, bias can arise from self-selection. Certain players might simply post what they want seen on Twitter, hoping to promote a good though insincere image. While we cannot rule out the possibility of self-selection, its confounding effect is minimal due to players’ spontaneous and genuine use of Twitter. Second, the procedures for identifying player mood from mountains of textual data are themselves another potential source of bias. However, the virtues of sentiment analysis can compensate for bias. As noted earlier, this emerging text-mining technique has demonstrated robust validity in gauging customer mood through Twitter, and quantified mood has been used to help predict a number of future events.

The next section explores a case study of the process of applying sentiment analysis to identify NBA players’ moods underlying their tweets, further showing how mood is associated with performance in upcoming games.

Case Study

This case study serves two purposes: detail a process of applying sentiment analysis to identify NBA player mood before games and test whether a player’s identified mood is associated with on-court performance in the ensuing game. For the study, we built two datasets: Athletes Generated Content dataset (AGC), which collected the tweets of NBA players posted during the 2012–2013 season, and the sports performance dataset, which likewise collected player data in all games during the 2012–2013 season.

Following the framework proposed by Matsudatira,¹⁴ we used Tweeting-Athletes.com to assemble a comprehensive list of NBA players (353 total) with Twitter accounts during the 2012–2013 season. The AGC dataset collected the tweets posted by these players during the season. For each tweet (content aside), the dataset also included descriptive attributes (such as timestamp when the tweet was posted). AGC found almost all active players in the league at the time had a Twitter account and posted 91,659 tweets^b that season, with most players, 266 of the 353, or 75.3%,

posting at least 100 tweets for the season. The dataset also identified some noticeable “heavy users” of tweets; for example, Dwight Howard, star center of the Houston Rockets, alone posted 1,214 tweets.

To build the sports-performance dataset, we retrieved box-score data and player data for the 2012–2013 season from Basketball-Reference.com. We combined the information from each game with the information from each player who played the game as a single record in the dataset. Extracted game information included game date, game type, home/away, opponent and win/loss, with score. Player information included age (on the day the game was played), games started, minutes played, field goals (percentage), three-point field goals (percentage), free throws (percentage), and plus/minus (+/-).^c

In order to represent all tweets accurately, we preprocessed their content by filtering out pure re-tweets and information-oriented tweets containing URL links. We also removed non-English tweets. Given the fact that tweets are replete with non-standard English, we applied a data-cleansing mechanism. Misspellings (such as “stealling” and “cuting”) represent a primary source of nonstandard English usage. We corrected misspellings through the minimum Hamming distance, an automatic error-correction algorithm. Non-standard English might have also originated from repeating letters (such as “Awwwwful,” “Awfuuuul,” and “Ruude”). We thus replaced letters occurring three or more times consecutively in the same word with two consecutive occurrences of the same letter. In the examples of repeating letters, the algorithm resulted in tokens “awful,” “awful,” and “Ruude,” respectively. We subsequently corrected the tokens as misspellings, if necessary, through the minimum Hamming distance.

We applied AFINN sentiment lexicons and an extended hand-assembled list of emoticons to detect the polarity of players’ sentiment in the tweets,

a For example, in June 2009, Minnesota Timberwolves forward Kevin Love revealed through Twitter that head coach Kevin McHale would not be returning the following season and was sad about it; the tweet put Love in an awkward position, as it made McHale’s decision public without his consent.

b Of the 51,847 original posts, 47,468 were in English.

c Plus/minus (+/-) is a well-established metric that accesses the overall impact a player has on the game and useful means to determine a players’ on-court performance, as it accounts for everything a player does on the court, even when unquantifiable.

Table 3. Emoticons.

Emoticons mapped to :)	Emoticons mapped to :(
:)	:(
:~)	:-(
:)	:(
:-p or :-q	:-e
^_^ or =^_-=	:-()
(^0^A) or ^0^A	X-<
=)	:{
:-D :D or =D	>:O or >:-O or >:o or >:-o

or players' moods behind the tweets. AFINN is a sentiment lexicon containing English words developed by Finn Årup Nielsen of the Technical University of Denmark. It rates words between -5 (negative) and +5 (positive), with a larger number representing a more positive sentiment. Though based on the Affective Norms for English Words lexicon proposed by Bradley and Lang,⁴ AFINN is a more refined and focused list based on the language used in microblogging platforms, including slang, obscene words, acronyms, and Web jargon prevalent on the Internet.⁴ We used an application based on R to automatically extract features related to the AFINN lexicon from a given tweet. We calculated an AFINN score for the tweet by summing the ratings of the positive and negative words matching the lexicon. The computed AFINN score reflects the mood of a player behind the tweet. Moreover, tweets are characterized by various emoticons that can express mood; for example, :) or :-~) express positive mood, and :(expresses negative mood. We also accounted for the emoticons in our extended list by mapping them to either ☹ or 😊,^d as in Table 3; the examples in Table 4 represent a snapshot of the results of our overall sentiment analysis.

Recall the second goal of our study was to test the association between players' identified moods before games and their on-court performance in subsequent games. A player might post several tweets before a game. Before we could simply aggregate the AFINN scores of the tweets to identify a player's mood before a game, we

^d Each word in the AFINN list has a score from -5 to +5 to describe sentiment polarity and degree; in our study, we extended the AFINN list by assigning -2 to ☹ and +2 to 😊.

Table 4. Examples of players' tweets reflecting mood polarities.

NBA Player	Tweet	Timestamp	AFINN Score*	Mood Polarity
 Position: Forward Team: Atlanta Hawks	Nice win for us on the road vs. Portland. Much better effort tonite. Go Hawks!	13-Nov-12 6:13:18 AM	5	Positive
	Shaq! RT @owen_then: @AL_Horford who's you alltime favorite center??	13-Nov-12 6:53:11 AM	1	Positive
	Happy Birthday! RT @Md813: @AL_Horford Can ur #1 fan get a birthday tweet??? :)	13-Nov-12 8:33:34 AM	2	Positive
	Thanks. I really enjoy having my own segment! RT @KimberMcCart: Love the @AL_Horford show on Tuesdays with @kingcfb and @RealMattlanta	14-Nov-12 4:55:42 PM	4	Positive
 Position: Guard Team: Cleveland Cavaliers	What's good! Here in Sacramento, getting ready for the game tonite. Go Hawks!	16-Nov-12 4:37:43 PM	4	Positive
	How u gonna fire the owner?	28-Nov-12 8:25:12 AM	-1	Negative
 Position: Guard Team: Los Angeles Lakers	@Naimthestar haha u sound really upset over there sir and is this a bad time to bring up that game on thanksgiving	28-Nov-12 8:39:51 AM	-2	Negative
 Position: Center Team: Cleveland Cavaliers	nomoreyears!	06-Nov-12 8:19:45 PM	-1	Negative
	Philly should be better than this. Damn shame.	06-Nov-12 8:43:52 PM	-3	Negative
	No hope. No change. Nobama.	07-Nov-12 6:58:26 PM	-3	Negative

Note:

* The AFINN score reflects the extent to which a player's mood hidden in a tweet is positive. The larger the AFINN score, the more positive the sentiment of a tweet, and vice versa.

also had to deal with timestamps. NBA games typically start at 8:00 P.M. on game day. Due to the league's media policy prohibiting players, coaches, and officials from using Twitter and other social network sites from 45 minutes prior to game time until after post-game interviews, we then selected 7:00 P.M. (of game day) as the cutoff time to define "before game." That is to say, for a particular game played by a particular player, we used his tweets back to 7:00 P.M. on game day until 7:00 P.M. one day before to discern his mood before the game. We then calculated the AFINN scores of the tweets submitted during this timeframe. The total score represents a player's mood before a game. The higher the aggregated score, the more positive a player's mood. Moreover, to enable comparison of mood among different players, we normalized the daily AFINN scores through the mean and standard deviation of all his seasonal available records.

To test the mood-performance relationship, we developed a model relating a player's mood before a game to his performance in the game. We began with a standard baseline model, including mood variable only, then extended the baseline model to a more comprehensive one consisting of variables commonly included in the sports analytics literature.

We thus adopted the following model

$$P_{i,t} = \alpha + \beta M_{i,t} + \varepsilon_{i,t} \text{ (baseline model)}$$

where i, t indexes a player-game-day's combination, $P_{i,t}$ represented the performance of player i for a game at day t , M represented a player's mood, and β was the coefficient of the regression model. Here, $\varepsilon_{i,t}$ was the unobserved player/game-specific effect reflecting the idiosyncratic characteristics associated with each player and with the game itself; Table 5 includes estimates under the column heading "Baseline Model."

The baseline model might be subject to the "endogeneity" influence common in sports analytics. To limit it, we added control variables that are potential predictors of sports performance. As suggested in the sports analytics literature, we devised the following extended model

Coaches, players, fans, and researchers alike share a well-founded belief that athletes' emotional states affect their on-court performance.

$$P_{i,t} = \alpha + \beta_1 M_{i,t} + \beta_2 P_{i,t-1} + \beta_3 \text{Salary}_{i,t} + \beta_4 \text{Age}_{i,t} + \beta_5 \text{Age}_{i,t}^2 + \beta_6 \text{Home}_{i,t} + \beta_7 \text{PosC}_i + \beta_8 \text{PosG}_i + \varepsilon_{i,t} \text{ (extended model)}$$

where $P_{i,t-1}$ is one game-day-lagged performance of player i for a game at day t , $\text{Salary}_{i,t}$ is the natural logarithm of the player's salary in the game season, $\text{Age}_{i,t}^2$ is the player's age on game day, $\text{Home}_{i,t}$ is the game if a home game (1) or away game (0), PosC_i and PosG_i are two position dummy variables (0 or 1) identifying the player's position (center, forward, or guard), and β_1 through β_8 are the coefficients of the extended regression model; the estimates are listed in Table 5 under the heading "Extended Model."

Consistent with the psychology and sports literature, we found mood does affect player performance— $\beta = 0.17, p < 0.01$ in the baseline model, and $\beta = 0.16, p < 0.01$ in the extended model. This result confirms a player's mood, as captured by his tweets posted prior to a game, is positively associated with his performance in the game. Such a finding is essential because it highlights the informational value of players' tweets. Team management could draw on sentiment analysis to identify players' moods by examining their tweets, detecting potential concerns regarding on-court performance. Coaching staff could then address players' negative moods or help inspire positive moods. In this sense, with the proposed framework, coaches and management staff are able to keep pace with players' emotional status by following their tweets. The extended model takes a step further by considering player attributes, including position, age, and salary, as well as an important game attribute, home or away game. This analysis supports the robustness of the results we found in the baseline model using Twitter as the sole intelligence source. The extended model suggests other applications of the framework for future work and users of sports analytics; for example, an analyst would be able to explore how the mood-performance relationship varies with different player factors and game factors.

Conclusion

Business intelligence involves using technologies to transform data into

Table 5. Parameter estimation.

Variables	Description and measure	Baseline Model			Extended Model		
		Coefficients	Standard error	t Stat	Coefficients	Standard error	t Stat
$M_{i,t}$	Player mood on game day (standardized)	0.17***	0.06	2.67	0.16***	0.06	2.55
$P_{i,t-1}$	Performance in previous game				0.06***	0.17	3.64
$Salary_{i,t}$	Nature log of salary in current season				1.14***	0.18	6.22
$Age_{i,t}$	Player's age on game day				-0.18 (NS)	0.60	-0.30
$Age_{i,t}^2$	Square of age on game day				0.003 (NS)	.01	0.37
$PosC_i$	Center flag				-0.82 (NS)	0.6	-1.36
$PosG_i$	Guard flag				-0.03 (NS)	0.41	-0.09
$Home_{i,t}$	Home game or away game				3.43***	0.38	9.12
F		7.14			20.68		
Adjusted R^2		1.04%			5.10%		
Standard error of estimate		11.10			10.92		
Observations		3,443			3,344		

Note: Significance level is represented as *** for $p < 0.01$, ** for $p < 0.05$, * for $p < 0.1$, and NS for non-significant p values.

meaningful information for business analysis and decision making. Against this backdrop, our proposed framework could be regarded as providing a new avenue for sports intelligence for coaches, general managers, and various sports organizations with the potential for expanding the sports analytics horizon. Along with the use of tweets among NBA players, social media analytics has emerged as a practical means for gathering intelligence about players and their performance. Players' tweets provide insight into their thoughts and opinions, especially what cannot be obtained through other sources. Moreover, players are heavy users of Twitter in that they update themselves frequently, providing much useful data for social media analytics. Our framework demonstrates the value of players' tweets as a source of direct game-performance intelligence.

Using sentiment analysis on players' tweets, coaches and general managers can unleash the power of the unstructured data in the tweets, transforming it into valuable information concerning players' pre-game moods and likely performance outcomes. Coaches, players, fans, and researchers alike share a well-founded belief that athletes' emotional states affect their on-court performance. Following this logic, we analyzed 91,659 tweets posted by NBA players during the

2012–2013 season, finding their pre-game emotional states, as captured through sentiment analysis on their tweets, was directly related to on-court performance after controlling for other factors affecting performance.

Acknowledgments

We thank the reviewers, as well as Robert Barbato, John Ettlie, Sean Hansen, Victor Perotti, Hao Zhang, and participants of the Saunders College of Business at the Rochester Institute of Technology research colloquium (2014–2015) for their insightful comments, suggestions, and advice. 

References

- Alamar, B. and Mehrotra, V. Beyond 'Moneyball': The rapidly evolving world of sports analytics. *Analytics Magazine* (Aug. 30, 2011).
- Badenhausen, K. Billion-dollar Knicks and Lakers top list of NBA's most valuable teams. *Forbes* (Jan. 23, 2013).
- Baskerville, R.L. and Myers, M.D. Information systems as a reference discipline. *MIS Quarterly* 26, 1 (Mar. 2002), 1–14.
- Bravo-Marquez, F., Mendoza, M., and Poblete, B. Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems* 69, 15 (Oct. 2014), 86–99.
- Culnan, M.J., McHugh, P.J., and Zubillaga, J.I. How large U.S. companies can use Twitter and other social media to gain business value. *MIS Quarterly Executive* 9, 4 (Dec. 2010), 243–259.
- Evangelopoulos, N. and Visinescu, L. Text-mining the voice of the people. *Commun. ACM* 55, 2 (Feb. 2012), 62–69.
- Fan, W. and Gordon, M. The power of social media analytics. *Commun. ACM* 57, 6 (June 2014), 74–81.
- Feldman, R. Techniques and applications for sentiment analysis. *Commun. ACM* 56, 4 (Apr. 2013), 82–89.
- Friedman, D.J. Social media in sports: Can professional sports league commissioners punish 'twackles dummies'? *Pace Intellectual Property, Sports & Entertainment Law Forum* 2, 1 (Spring 2012), 74–102.
- Fry, M. and Ohlmann, J. Introduction to the special issue on analytics in sports: Part I: General sports applications. *Interfaces* 42, 2 (Apr. 2012), 105–118.

- Kalampokis, E. Understanding the predictive power of social media. *Internet Research* 23, 5 (2013), 544–559.
- Lazarus, R.S. How emotions influence performance in competitive sports. *The Sport Psychologist* 14, 3 (2000), 229–252.
- Majchrzak, A., Cherbakov, L., and Ives, B. Harnessing the power of the crowds with corporate social networking tools: How IBM does it. *MIS Quarterly Executive* 8, 2 (June 2009), 103–118.
- Matsudaira, K. Capturing and structuring data mined from the Web. *Commun. ACM* 57, 3 (Mar. 2014), 10–11.
- Rui, H., Liu, Y., and Whinston, A. Whose and what chatter matters? The effect of tweets on movie sales. *Decision Support Systems* 55, 4 (Nov. 2013), 863–870.
- Schumaker, R., Solieman, O., and Chen, H. Sports knowledge management and data mining. *Annual Review of Information Science and Technology* 44, 1 (2010), 115–157.
- Stein, M. NBA social media guidelines out. ESPN.com, Sept. 30, 2009; <http://sports.espn.go.com/nba/news/story?id=4520907>
- Totterdell, P. Catching moods and hitting runs: Mood linkage and subjective performance in professional sport teams. *Journal of Applied Psychology* 85, 6 (2000), 848–859.
- Totterdell, P. Mood scores: Mood and performance in professional cricketers. *British Journal of Psychology* 90, 3 (Aug. 1999), 317–332.
- Walsh, D. All a Twitter: Social networking, college athletes, and the First Amendment. *William & Mary Bill of Rights Journal* 20, 2 (2011), 619–650.
- Yu, Y., Duan, W., and Cao, Q. The impact of social and conventional media on firm equity value: A sentiment analysis approach. *Decision Support Systems* 55, 4 (Nov. 2013), 919–926.

Chenyan Xu (chenyan.xu@stockton.edu) is an assistant professor in the Computer Science & Information Systems Program at Stockton University, Galloway, NJ.

Yang Yu (yyu@saunders.rit.edu) is an assistant professor of management information systems in the Saunders College of Business at the Rochester Institute of Technology, Rochester, NY.

Chun-Keung Hoi (ckhoi@saunders.rit.edu) is a professor of finance in the Saunders College of Business at the Rochester Institute of Technology, Rochester, NY.

Inductive programming can liberate users from performing tedious and repetitive tasks.

BY SUMIT GULWANI, JOSÉ HERNÁNDEZ-ORALLO, EMANUEL KITZELMANN, STEPHEN H. MUGGLETON, UTE SCHMID, AND BENJAMIN ZORN

Inductive Programming Meets the Real World

MUCH OF THE world's population use computers for everyday tasks, but most fail to benefit from the power of computation due to their inability to program. Most crucially, users often have to perform repetitive actions manually because they are not able to use the macro languages available for many application programs. Recently, a first mass-market product was presented in the form of the Flash Fill feature in Microsoft Excel 2013. Flash Fill allows end users to automatically generate string-processing programs for spreadsheets from one or more user-provided examples. Flash Fill is able to learn a large variety of quite complex programs from only a few examples because of incorporation of inductive programming methods.

Inductive programming (IP) is an interdisciplinary domain of research in computer science, artificial

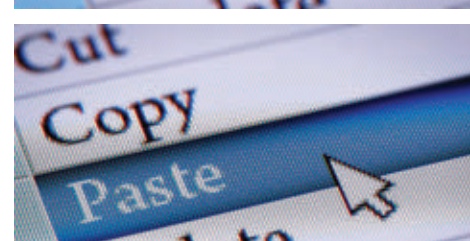
intelligence, and cognitive science that studies the automatic synthesis of computer programs from examples and background knowledge. IP developed from research on inductive program synthesis, now called *inductive functional programming* (IFP), and from inductive inference techniques using logic, nowadays termed *inductive logic programming* (ILP). IFP addresses the synthesis of recursive functional programs generalized from regularities detected in (traces of) input/output examples^{19,41} using generate-and-test approaches based on evolutionary^{27,34,35} or systematic^{16,28} search or data-driven analytical approaches.^{6,11,17,23,38} Its development is complementary to efforts in synthesizing programs from complete specifications using deductive and formal methods.⁸

ILP originated from research on induction in a logical framework^{30,39} with influence from artificial intelligence, machine learning, and relational databases. It is a mature area with its own theory, implementations, and applications and recently celebrated the 20th anniversary³³ of its inception as an annual series of international conferences.

Over the last decade IP has attracted a series of international workshops. Recent surveys^{7,13,18} reflect the wide variety of implementations and applications in this area.

» key insights

- Supporting end users to automate complex and repetitive tasks using computers is a big challenge for which novel technological breakthroughs are demanded.
- The integration of inductive programming techniques in software applications can support users by learning domain-specific programs from observing interactions of the user with the system.
- Inductive programming is being transferred to real-world applications such as spreadsheet tools, scripting, and intelligent program tutors.
- In contrast to standard machine learning, in inductive programming learning from few examples is possible because users and systems share the same background knowledge.



In the domain of end-user programming, programming by demonstration approaches were proposed that support the learning of small routines from observing the input behavior of users.^{5,22,24} Excel's Flash Fill provides an impressive illustration that program synthesis methods developed in IP can be successfully applied to gain more flexibility and power for end-user programming.^{9,11} Further applications are being realized for other desktop applications (for example, PowerShell scripting in the Convert-From-String cmdlet²) as well as for special-purpose devices such as home robots and smartphones.

In this article, several of these current applications are presented. We contrast the specific characteristics of IP with those of typical machine learning approaches and we show how IP is related to cognitive models of human inductive learning. We finally discuss recent techniques—such as the use of domain-specific languages and meta-level learning—

that widen the scope and power of IP and discuss new challenges.

Real-World Applications

Originally, IP was applied to synthesizing functional or logic programs for general-purpose tasks such as manipulating data structures (for example, sorting or reversing a list). These investigations showed that small programs could be synthesized from a few input/output examples. The recent IT revolution has created real-world opportunities for such techniques. Most of today's large number of computer users are non-programmers and are limited to being passive consumers of the software that is made available to them. IP can empower such users to more effectively leverage computers for automating their daily repetitive tasks. We discuss here some such opportunities, especially in the areas of end-user programming and education.

End-user programming. End users of computational devices often need to create *small* (and perhaps one-off)

scripts to automate repetitive tasks. These users can easily specify their intent using *examples*, making IP a great fit. For instance, consider the domain of data manipulation. Documents of various types, such as text/log files, spreadsheets, and Web pages, offer their creators great flexibility in storing and organizing hierarchical data by combining presentation and formatting with the underlying data model. However, this makes it extremely difficult to extract the underlying data for common tasks such as data processing, querying, altering the presentation view, or transforming data to another storage format.

Existing programmatic solutions to manipulating data (such as Excel macro language, regular expression libraries inside Perl/Python, and JQuery library for JavaScript) have three key limitations. First, the solutions are domain-specific and require expertise in different technologies for different document types. Second, they require understanding of the entire underlying document structure including the data fields the end user is not interested in (some of which may not even be visible in the presentation layer of the document). Third, and most significantly, they require knowledge of programming. As a result, users must resort to manually performing repetitive tasks, which is both time consuming and error prone.

Inductive synthesis can help out with a variety of data manipulation tasks. These include: (a) Extracting data from semi-structured documents including text files, Web pages, and spreadsheets²³ (as shown in Figure 1). (b) Transformation of atomic data types such as strings⁹ (as illustrated in Figure 2) or numbers. Transformation of composite data types such as tables¹¹ and XML.³⁵ (c) Formatting data.³⁶ Combining these technologies in a pipeline of extraction, transformation, and formatting can allow end users to perform sophisticated data manipulation tasks.

Computer-aided education. Human learning and communication is often structured around examples—be it a student trying to understand or master a certain concept using examples, or be it a teacher trying to understand a student's misconceptions or provide feedback using example

Figure 1. FlashExtract.²³

A framework for extracting data from documents of various kinds such as text files and Web pages using examples. Once the user highlights one or two examples of each field in the text file in (a), FlashExtract extracts more such instances and arranges them in a structured format in the table in (b). This is enabled by synthesis of a program in the domain-specific language (DSL) in (c) that is consistent with the examples in (a) followed by execution of that program on the text file in (a).

Ana Trujillo
357 21st Place SE
Redmond, WA
(757) 555-1634

Antonio Moreno
515 93th Lane
Renton, WA
(411) 555-2786

Thomas Hardy
742 17th Street NE
Seattle, WA
(412) 555-5719

Christina Berglund
475 22th Lane
Redmond, WA
(443) 555-6774

Hanna Moos
785 45th Street NE
Puyallup, WA
(376) 555-2462

Frederique Citeaux
308 66th Place
Redmond, WA
(689) 555-2770

(a)

Label 1	Label 2	Label 3
Ana Trujillo	Redmond	(757) 555-1634
Antonio Moreno	Renton	(411) 555-2786
Thomas Hardy	Seattle	(412) 555-5719
Christina Berglund	Redmond	(443) 555-6774
Hanna Moos	Puyallup	(376) 555-2462
Frederique Citeaux	Redmond	(689) 555-2770

(b)

```

PairSeq SS ::= LinesMap( $\lambda x$ : Pair(Pos( $x, p_1$ ), Pos( $x, p_2$ )), LS)
| StartSeqMap( $\lambda x$ : Pair( $x$ , Pos( $R_0[x]$ ,  $p$ )), PS)
LineSeq LS ::= FilterInt(init, iter, BLS)
BoolLineSeq BLS ::= FilterBool( $b$ , split( $R_0$ , 'n'))
PositionSeq PS ::= LinesMap( $\lambda x$ : Pos( $x$ ,  $p$ ), LS)
| FilterInt(init, iter, PosSeq( $R_0$ ,  $rr$ ))
Pred b ::=  $\lambda x$ : {Starts, Ends}With( $r, x$ ) |  $\lambda x$ : Contains( $r, k, x$ )

```

(c)

behaviors. Example-based reasoning techniques developed in the inductive synthesis community can help automate several repetitive and structured tasks in education including problem generation, solution generation, and feedback generation.¹⁰ These tasks can be automated for a wide variety of STEM subject domains including logic, automata theory, programming, arithmetic, algebra, and geometry. For instance, Figure 3 shows the output of an inductive synthesis technique for generating algebraic proof problems similar to a given example problem.

Future opportunities. We have described important real-world applications of IP. We believe there are many other domains to which IP can and will be applied in the near future. Any domain in which a set of high-level abstractions already exists is a strong candidate for IP. For example, the If This Then That (IFTTT) service (<http://ifttt.com/>), which allows end users to express small rule-based programs using triggers and actions, is an excellent candidate for application of IP. IFTTT programs connect triggers (such as *I was tagged in a photo*) with actions (such as *send an email message*) over specific channels such as Facebook. In such a domain, IP can be used to learn programs from examples of a user doing the task. For instance, it can learn a program to send a text message every time a smartphone user leaves work for home. Looking further ahead, automatically building robot strategies from user provided examples³⁰ is a promising new direction for IP.

As the frameworks to build IP-based solutions mature, including meta-synthesis frameworks that simplify the process of building synthesizers (as we will discuss later), it will become easier for developers to create new IP-empowered applications.

IP vs. Machine Learning

IP is concerned about making machines learn programs automatically and can hence be considered another machine learning paradigm. So, what is distinctive about inductive programming? Table 1 outlines a series of differences, some of which we discuss here.

We will also use a running example to indicate some of the features about IP. Figure 4 shows an illustrative ap-

Figure 2. Flash Fill.⁹

An Excel 2013 feature that automates repetitive string transformations using examples. Once the user performs one instance of the desired transformation (row 2, col. B) and proceeds to transforming another instance (row 3, col. B), Flash Fill learns a program `Concatenate(ToLower(SubString(v,WordToken,1)), " ", ToLower(SubString(v,WordToken,2)))`, which extracts the first two words in input string v (col. A), converts them to lowercase, and concatenates them separated by a space character, to automate the repetitive task.

	A	B
1	Email	Column 2
2	Nancy.FreeHafer@fourthcoffee.com	nancy freehafer
3	Andrew.Cencici@northwindtraders.com	andrew cencici
4	Jan.Kotas@litwareinc.com	jan kotas
5	Mariya.Sergienko@gradicdesigninstitute.com	mariya sergienko
6	Steven.Thorpe@northwindtraders.com	steven thorpe
7	Michael.Neipper@northwindtraders.com	michael neipper
8	Robert.Zare@northwindtraders.com	robert zare
9	Laura.Giussani@adventure-works.com	laura giussani
10	Anne.HL@northwindtraders.com	anne hl
11	Alexander.David@contoso.com	alexander david
12	Kim.Shane@northwindtraders.com	kim shane
13	Manish.Chopra@northwindtraders.com	manish chopra
14	Gerwald.Oberleitner@northwindtraders.com	gerwald oberleitner
15	Amr.Zaki@northwindtraders.com	amr zaki
16	Yvonne.McKay@northwindtraders.com	yvonne mckay
17	Amanda.Pinto@northwindtraders.com	amanda pinto

Figure 3. Problem generation for algebraic proof problems involving identities over analytic functions.

A given problem is generalized into a template and valid instantiations are found by testing on random values for free variables.

Example Problem ↓	$\frac{\sin A}{1+\cos A} + \frac{1+\cos A}{\sin A} = 2 \csc A$
Generalized Problem Template ↓	$\frac{T_1 A}{1+T_2 A} + \frac{1+T_3 A}{T_4 A} = 2 T_5 A$ where $T_i \in \{\cos, \sin, \tan, \cot, \sec, \csc\}$
New Similar Problems	$\frac{\cos A}{1-\sin A} + \frac{1-\sin A}{\cos A} = 2 \tan A$ $\frac{\cos A}{1+\sin A} + \frac{1+\sin A}{\cos A} = 2 \sec A$ $\frac{\cot A}{1+\csc A} + \frac{1+\csc A}{\cot A} = 2 \sec A$ $\frac{\tan A}{1+\sec A} + \frac{1+\sec A}{\tan A} = 2 \csc A$ $\frac{\sin A}{1-\cos A} + \frac{1-\cos A}{\sin A} = 2 \cot A$

Table 1. A simplified comparison between inductive programming and other machine learning paradigms.

	Inductive Programming	Other Machine Learning Paradigms
Number of examples	Small	Large, for example, big data
Kind of data	Relational, constructor-based datatypes	Flat tables, sequential data,
Data Source	Human experts, software applications, HCI, and others	Transactional databases, Internet, sensors (IoT), and others.
Hypothesis Language	Declarative: general programming languages or domain-specific languages	Linear, non-linear, distance-based, kernel-based, rule-based, probabilistic, etc.
Search strategy	Refinement, abstraction operators, brute-force.	Gradient-descent, data partition, covering, instance-based, etc.
Representation learning	Higher-order and predicate/function invention	Deep learning and feature learning.
Pattern comprehensibility	Common.	Uncommon.
Pattern expressiveness	Usually recursive, even Turing-complete.	Feature-value, not Turing complete.
Learning bias	Using background knowledge and constraints	Using prior distributions, parameters and features.
Evaluation	Diverse criteria, including simplicity, comprehensibility	Oriented to error (or loss) minimisation.
Validation	Code inspection, divide-and-conquer debugging, background knowledge consistency	Statistical reasoning (only a few techniques are locally inspectable).

plication where the goal is to identify repetitive patterns and rules about a user's personal contacts. The IP system learns an easy rule stating the user's boss must be added to her circles and a more complex one that states any person who is married to a family member should also be added to her circles. There we can see some of the distinctive features of IP systems, such as the small number of examples, the kind and source of data, the role of background knowledge, the interaction and feedback from the user, the use of a common declarative language, the use of recursion, and the comprehensibility and expressiveness of the learned patterns. Though machine learning and IP are quite complementary, they can also work well together. For instance, if IP generates multiple programs that are consistent with the provided examples, then machine learning can be used to rank such programs.¹¹

Small data. As collecting and storing data is becoming cheaper, it is easy to gain the impression the only interesting datasets today involve *big data*. However, datasets from a single user's interaction with whatever kind of device are usually quite small, such as the amount of data gathered about a person's agenda, as shown at the top of Figure 4.

It is well known that learning from small numbers of examples is more difficult and unreliable than learning from lots of data. The fewer examples we have, the more prone we are to overfitting, especially with ex-

pressive languages. IP is particularly useful when the number of examples is small but the hypothesis space is large (Turing-complete).

Declarative representation. Most (statistical) machine learning techniques are based on probabilities, distances, weights, kernels, matrices, and so on. None of these approaches, except for techniques based on (propositional) decision trees and rules, are *declarative*, that is, expressed as potentially comprehensible rules. Hence, another distinctive feature of IP is it uses a rich symbolic representation, as hypotheses are usually *declarative* programs.

The declarative approach permits the use of a single language to represent background knowledge, examples and hypotheses, as shown in Figure 4. Apart from the accessibility of one single language for the (end-)user, knowledge can be inspected, revised and integrated with other sources of knowledge much more easily. As a result, incremental, cumulative or life-long learning becomes easier.¹³ For instance, NELL (Never-Ending Language Learner)³ uses an ILP algorithm that learns probabilistic Horn clauses.

Today, many languages in IP are *hybrid* such as functional logic programming languages, logic programming with types and higher-order constructs, constraints, probabilities, and so on. The logic (ILP) vs. functional (IFP) debate has also been surpassed recently by the breakthrough of domain-specific languages (DSL), which are usually better suited for the application at hand, as we discuss later.

Refinement and abstraction. Another particular issue about IP is the way the hypothesis space is arranged by properly combining several inference mechanisms such as deduction, abduction, and induction. Many early IP operators were inversions of deduction operators, leading to bottom-up and top-down approaches, where generalization and specialization operators, respectively, are used.³³ More generally, *refinement* and *abstraction* operators, including the use of higher-order functions, predicate, and function invention, can be defined according to the operational semantics of the language.

This configures many levels between merely extensional facts and more intensional knowledge, leading to a hierarchical structure. Actually, the use of the same representation language for facts, background knowledge and hypotheses, as illustrated in Figure 4, facilitates this hierarchy.

Deep knowledge. Because of the abstraction mechanisms and the use of background knowledge, IP considers learning as a knowledge acquisition process. In Figure 4, for instance, inductive programming has access to some information about contact groups as well as relationships between the contacts (such as family bonds or work hierarchies). Such knowledge is known as *background knowledge* and works as a powerful explicit bias to reduce the search space and to find the right level of generalization.

Knowledge can be considered *deep* if it references lower-level definitions,

including recursively referencing itself. Representation of such structured and deep knowledge is achieved by programming languages that feature variables, rich operational semantics and, most especially, *recursion*. Recursion is a key issue in inductive programming.^{19,30,38,41} Note that both the background knowledge and the new hypothesis in the Figure 4 example are recursive.

This is in contrast to other machine learning approaches where background knowledge has only the form of prior distributions, probabilities, or features. The difference is also significant with other non-symbolic approaches to deep learning,¹ a new approach in machine learning where more complex models and features are also built in a hierarchical way, but data, knowledge, and bias are represented differently.

Purpose and evaluation. In other machine learning approaches, hypotheses are measured by different metrics accounting for a degree of error. The purpose of IP is not just to maximize some particular error metric, but to find meaningful programs that are operational, according to the purpose of the IP application. This usually implies they have to be consistent with most of (if not all) the data but also with the background knowledge and other possible constraints. Also, as hypotheses are declarative (and possibly recursive), the evaluation is more diverse, including criteria such as simplicity, comprehensibility, coherence, and time/space complexity.

Recent Techniques

IP is essentially a search problem, and can benefit from techniques developed in various communities. We present here certain classes of techniques used in recent IP work.

DSL synthesizers. DSLs have been introduced in the IP scenario under the following methodology:

1. *Problem definition:* Identify a vertical domain of tasks and collect common scenarios by studying help forums and conducting user studies.

2. *Domain-specific language:* Design a DSL that is expressive enough to capture several real-world tasks in the domain, but also restricted enough to enable efficient learning from examples. Figure 1(c) de-

scribes one such DSL for extracting data from text files. (The full version of this DSL along with its semantics is described in Le and Gulwani.²³) This DSL allows for extracting a sequence of substrings using composition of filter and map operations.

3. *Synthesis algorithm:* Most of these algorithms work by applying divide-and-conquer deductive techniques that

systematically reduce the problem to the synthesis of subexpressions of the original expression (by translating the examples for the expression to the examples for the sub-expressions). These algorithms typically end up computing a set of DSL programs.

4. *Ranking:* Rank the various programs returned by the synthesizer perhaps using machine learning techniques.

Figure 4. An example of the interaction with an IP system and the key role of the background knowledge.

A user interacts with several devices about his/her agenda and contact groups. The system gathers a collection of facts, rules and operators into its background knowledge. From this background knowledge and a few examples, the system is able to infer new rules that—once validated—can be used to enlarge the background knowledge and also to make recommendations or suggest choices for the user.

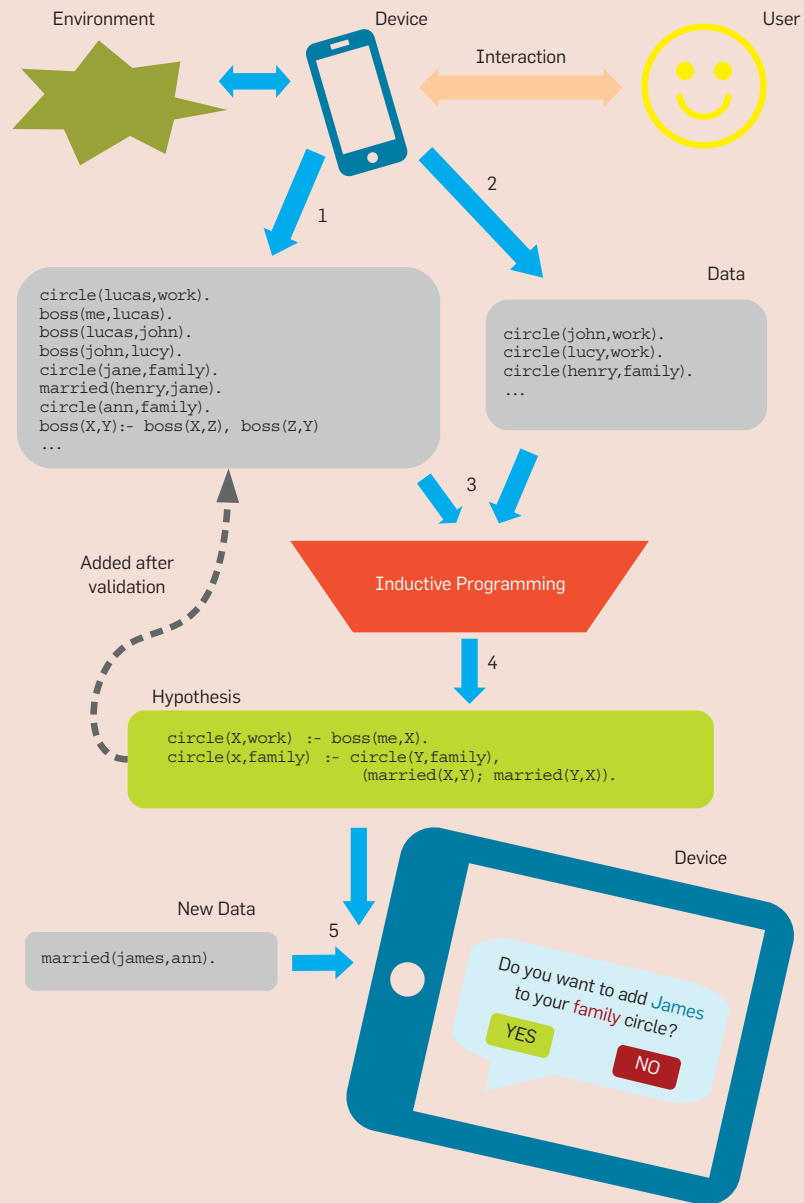


Figure 5. Given examples to reverse a list for lengths zero to three, Igor2 synthesizes a program using the higher-order function fold.

The auxiliary function f , which appends a single element to the end of a list and parameterizes the fold, is not given as background knowledge but is invented.

```
reverse x = fold f [] x

f x0 [] = [x0]

f x0 (x1 : xs) = x1 : f x0 xs
```

Table 2. Use of the chain hypothesis on the boss predicate from Figure 4 to prove a new example of boss(lucas, lacy) using the meta-rule and the background knowledge.

Name	Meta-Rule
Chain	$P(x, y) \leftarrow Q(x, z), R(z, y)$
Example	Background Knowledge
boss(lucas, lacy)	boss(lucas, john). boss(john, lacy).
Meta-Substitution	Chain Hypothesis
$P=Q=R=$ boss	boss(X,Y) :- boss(X,Z), boss(Z,Y)

This methodology has been applied to various domains including the transformation of syntactic strings,^{9,29} semantic strings, numbers, and tables.¹¹

Meta-synthesis frameworks. Domain-specific synthesizers (as opposed to general-purpose synthesizers) offer several advantages related to efficiency (such as the ability to synthesize programs quickly) and ranking (such as the ability to synthesize intended programs from fewer examples). However, the design and development of a domain-specific synthesizer is a nontrivial process requiring critical domain insights and implementation effort. Furthermore, any changes to the DSL require making nontrivial changes to the synthesizer.

A meta-synthesis framework allows easy development of synthesizers for a related family of DSLs that are built using the same core set of combinators. Building such a framework involves the following steps:

1. Identify a family of vertical task domains that allow a common user interaction model.
2. Design an algebra for DSLs. A DSL is an ordered set of grammar rules (to model ranking).
3. Design a search algorithm for each algebra operator such that it is compositional and inductive.

Meta-synthesis frameworks can al-

low synthesizer writers to easily develop domain-specific synthesizers, similar to how declarative parsing frameworks like lex and yacc allow a compiler writer to easily write a parser. The FlashExtract framework²³ and the Test Driven Synthesis framework³⁵ allow easy development of synthesizers for extracting and transforming data from documents of various types such as text files, Web pages, XML documents, tables, and spreadsheets. Figure 1(c) describes a DSL composed of Filter and Map operators, which are supported by the FlashExtract framework.

The FlashExtract framework is thus able to automatically construct an efficient synthesis algorithm for this DSL.

Higher-order functions are a possibility to provide a bias when searching for hypotheses. In contrast to DSLs, higher-order functions do not tailor IP to a predefined domain, but instead provide common patterns for processing recursive (linked) data as background knowledge to the IP system. For instance, the `fold` higher-order function (also known as *reduce*) iterates over a list of elements and combines the elements by applying another function that is also given as a parameter to the `fold`. For example, `fold (+) 1, 2, 3, 4]` would combine the numbers in the list with the plus function and re-

turn 10. Rather than learning a recursive function, the IP system then only needs to pick the suitable higher-order function and instantiate it appropriately. One of the first systems that made use of higher-order functions in IP was MAGICHASKELLER,¹⁶ which generates Haskell functions from a small set of positive inputs. The generated programs are instantiations of a predefined set of higher-order functions such as `fold`. An extension of the analytical IP system Igor2, also implemented in Haskell, takes a similar approach. In contrast to Katayama,¹⁶ which finds programs by enumeration, Igor2 analyzes the given data to decide which higher-order function fits. The argument function to instantiate the higher-order function is either picked from background knowledge or, if not existing, is invented as an auxiliary function. The use of this technique not only results in a speed-up of synthesis but also enlarges the scope of synthesizable programs.¹⁴ An example for induction with higher-order functions is given in Figure 5. Finally, Henderson¹² proposed to use higher order to constrain and guide the search of programs for cumulative learning where functions induced from examples are abstracted and can then be used to induce more complex programs. Unlike a DSL, higher-orderness does not restrict IP to a predefined domain, instead it guides search.

Meta-interpretive Learning (MIL) is a recent ILP technique^{31,32} aimed at supporting learning of recursive definitions. A powerful and novel aspect of MIL is that when learning a predicate definition it automatically introduces sub-definitions, allowing decomposition into a hierarchy of reusable parts. MIL is based on an adapted version of a Prolog meta-interpreter. Normally such a meta-interpreter derives a proof by repeatedly fetching first-order Prolog clauses whose heads unify with a given goal. By contrast, a meta-interpretive learner additionally fetches higher-order meta-rules whose heads unify with the goal, and saves the resulting meta-substitutions to form a program. To illustrate this idea, consider the meta-rule in Table 2 relating to P , Q , and R . In this example, the Chain Hypothesis on the boss predicate from Figure 4 is learned from the Meta-substitutions

into the X , Y , and Z meta-variables by proving the Example using the Meta-rule and the Background knowledge.

Given the higher-order substitutions, instantiated program clauses can be reconstructed and reused in later proofs, allowing a form of IP that supports the automatic construction of a hierarchically defined program.

In Lin et al.²⁵ the authors applied MIL to a task involving string transformations tasks previously studied by Gulwani.⁹ Figure 6 shows the outcome of applying MIL to learning a set of such tasks, using two approaches, *dependent* and *independent* learning, where in the former new definitions are allowed to call already learned definitions at lower levels. Dependent learning produced more compact programs owing to the reuse of existing subdefinitions. This in turn led to reduced search times since the smaller task definitions required less search to find them.

Constraint solving. The general idea here is to reduce the synthesis problem to an equivalent satisfiability problem that is expressed as a standard logical formula. Then, this formula can be solved by a general off-the-shelf tool using the recent advances made in the technology of Satisfiability (SAT) and Satisfiability Modulo Theory (SMT) solvers. This approach has been applied to synthesis from complete formal specifications, but its applicability has been limited to synthesizing restricted forms of programs. On the other hand, if the specification is in the form of examples, then the reduction of the synthesis problem to solving of SAT/SMT constraints can be performed for a larger variety of programs. These examples may be generated inside a counter-example guided inductive synthesis loop⁴⁰ (which involves using a validation technology to find new test inputs on which the current version of the synthesized program does not meet the given specification), or using a distinguishing input-based methodology¹⁵ (which involves finding new test inputs that distinguish two semantically distinct synthesized programs, both of which are consistent with the given set of examples).

Challenges

There is an ongoing research effort in IP to address increasingly challenging problems in terms of size, effectiveness, and robustness.

Compositionality. The ability of IP to perform adequately for more complex tasks will require breakthroughs in several areas. First, the underlying complexity of the search space for correct solutions limits the overall usability of IP, especially in interactive settings where instant feedback is required. There will undoubtedly be improvements in the performance of such algorithms, including approaches such as version space algebras that provide compact representations of the search space. Ultimately, there will be limits to complexity that no algorithm improvements can address. In such cases, new approaches are needed that allow users to decompose more complex tasks into sufficiently small subtasks and then incrementally compose the solutions provided by IP for each subtask.

New kinds of brute-force search.

The general idea here is to systematically explore the entire state space of artifacts and check the correctness of each candidate against the given examples. This approach works relatively well when the specification consists of examples (as opposed to a formal relational specification) since checking

the correctness of a candidate solution against examples can be done much faster than validating the correctness against a formal relational specification. However, this is easier said than done because of the huge underlying state space of potential artifacts and often requires innovative nontrivial optimizations, such as goal-directed search, branch and bound, complexity-guided evolutionary approaches, clues based on textual features of examples,²⁸ and offline indexing.¹⁶

Domain change. Applying IP to new domains efficiently will also require new approaches, including the creation of meta-synthesizers as mentioned earlier. Because the application of IP techniques in real-world applications is relatively new, there is insufficient experience in exploring the space of applications to clearly identify common patterns that might arise across domains. It is likely that in the short term, domain-specific IP systems will be developed in an ad hoc way, and which over time, as experience with such systems grows, new approaches will systematize and formalize the ad hoc practices, so systems become more general and reusable across different domains.

Validation. It is important that the artifacts produced by IP give the end user confidence that what they have created is correct and makes sense. For

Figure 6. A comparison of the programs generated by Dependent and Independent Learning using MIL.

Nodes marked n correspond to programs that solve task n , and nodes are arranged vertically according to their sizes. For Dependent Learning (left), the arrows correspond to the calling relationships of the induced programs. Dependent learning produces reuse of existing subdefinitions that in turn leads to reduced search times.

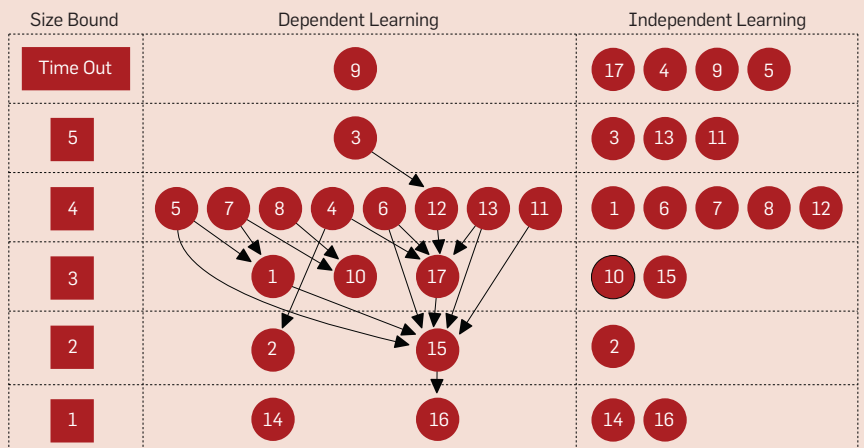
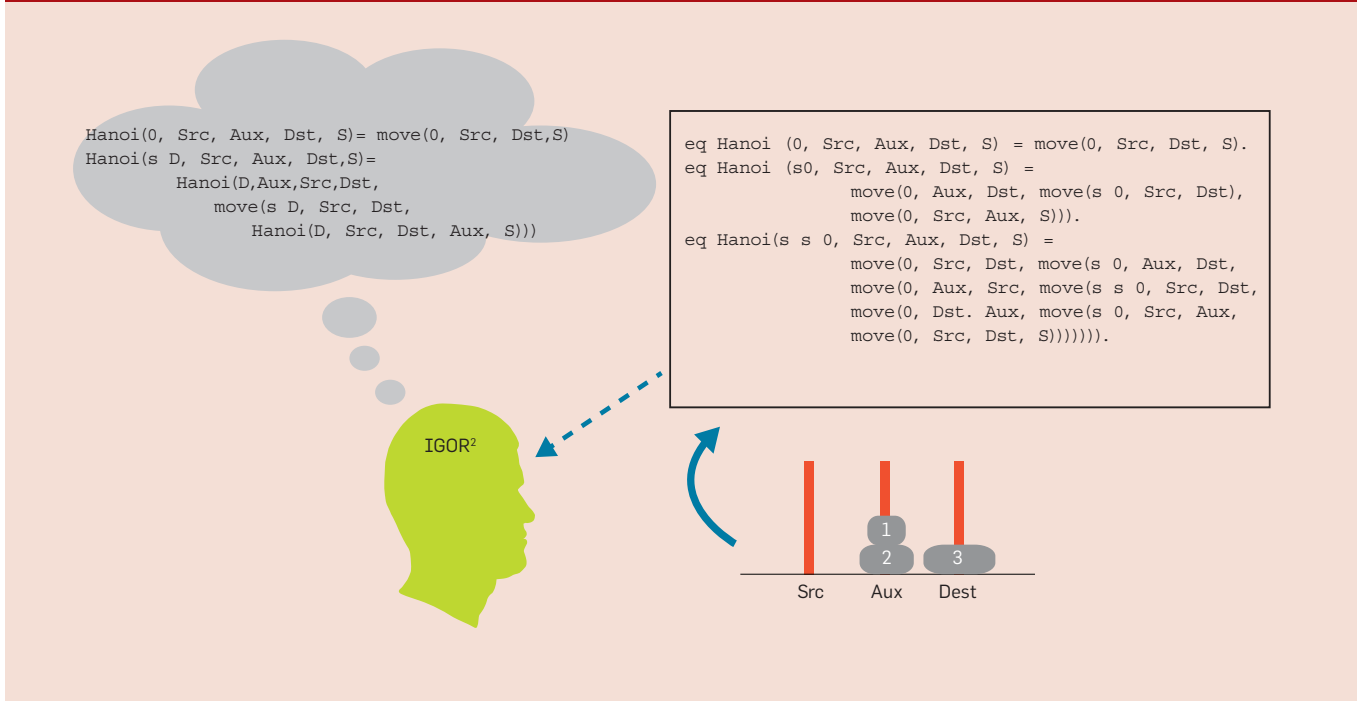


Figure 7. Input of the *Tower of Hanoi* problem for Igor2 and the induced recursive rule set.

instance, the plethora of automatically named hierarchy of invented sub-tasks generated by approaches such as Meta-Interpretive Learning (as discussed previously) can lead to confusion if the new names do not bear a clear correspondence to the semantics of the sub-tasks being defined. To address such challenges, we must find new approaches to explain the behavior of the resulting program to the user in intuitive terms and find ways for them to guide the solution if it is incorrect. There is great room for creativity on this problem, such as the use of abstractions that connect the user to the IP result, the ability to highlight those inputs where the tool is less confident and the user should consider inspecting the results, explicitly showing the inferences the synthesized program is applying in domain-specific intuitive ways (for example, using pictures), and paraphrasing synthesized programs in natural language and letting the user make stylized edits.

Noise tolerance. Real data is often unclean—some values might be missing and/or incorrect, while some values might occur in different formats (as in representations for dates and numbers). Sometimes even the background knowledge can be incorrect if the user accidentally makes mistakes in providing it.

Addressing the issue of robustness to such noise may be best done in a domain-specific manner. For example, if a table contains mostly correct data with a few outliers, existing techniques to detect and report outliers (or even just missing values) will help the IP process. Fortunately, there is a body of work in the existing ML literature that can be applied to this problem (see, for example, Chandola et al.⁴).

Making IP more cognitive. Cognitive science and psychology have shown that humans learn from a small number of—usually positive—examples and are relatively intolerant to exceptions.²⁶ Coherence, simplicity and explanatory power are guiding rules in human inductive inference. The role of background knowledge and the necessary constructs that need to be developed in order to acquire more abstract concepts have also been studied in cognitive science.⁴² The progressive acquisition of deep knowledge in humans is especially prominent in language learning but also in learning from problem-solving experience. Recently, IP has been used in the context of cognitive modeling, demonstrating that generalized rules can be learned from only a few positive examples.³⁷ For instance, Figure 7 shows the result of learning the *Tower of Hanoi* problem induced by the IP system Igor2. This

result is equivalent to the generalization from three disc to n disc problems (some) humans would infer from the same examples.²⁰ However, up until now there are no empirical studies that allow for a detailed comparison between high-level human learning of complex routines and the training input and the induced programs of IP systems, to see whether they lead to similar solutions and, when they diverge, to see whether the IP solution is still comprehensible to humans.

Presupposing the declarative nature of learning in IP systems is sufficiently similar to knowledge-level learning in humans, IP systems could be augmented with a Cognitive User Interface⁴³ with the ultimate goal that machines interact like humans, and evaluate whether in this way they can become more intuitive, trustable, familiar, and predictable—including predicting when the system is going to fail. In order to achieve this through IP we need to settle the *interaction model*. For instance, the supervision from the user can be limited to some rewards (“OK” buttons) or penalties (“Cancel” buttons) about what the system is doing, as illustrated in Figure 4. Alternatively, the user can give a few examples, the IP system makes guesses for other examples and the user corrects them.^{5,11} In this interac-

tive (or query) learning process the user can choose among a set of candidate hypotheses by showing where they differ, using a *distinguishing input* generated by the user—or more effectively—by the IP system itself.¹⁵

Conclusion

Since the 1970s, basic research in IFP and ILP resulted in the development of fundamental algorithms tackling the problem of inducing programs from input/output examples. However, these approaches remained within the context of artificial intelligence research and did not trigger a successful transfer into technologies applicable in a wider context. In 2009, Tessa Lau presented a critical discussion of programming by demonstration systems noting that adoption of such systems is not yet widespread, and proposing this is mainly due to lack of usability of such systems.²¹ In this article, we have presented recent work in IP where we identified several new approaches and techniques that have the potential to overcome some restrictions of previous systems: learning from very few positive examples becomes possible when users and systems share background knowledge that can be represented in a declarative way, which, combined with name inference, is likely to be more easily understandable. Using algorithmic techniques developed in either or both ILP and IFP as well as the use of higher-order functions and meta-interpretative learning resulted in more powerful IP algorithms; the adoption of techniques based on domain-specific languages has allowed the realization of technologies ready to use in mass-market products as demonstrated by Flash Fill. Hopefully, the recent achievements will attract more researchers from the different areas in which IP originated—AI, machine learning, functional programming, ILP, software engineering, and cognitive science—to tackle the challenge of bringing IP from the lab into the real world. 

References

1. Bengio, Y., Courville, A. and Vincent, P. Representation learning: A review and new perspectives. *Pattern Anal. Machine Intell.* 35, 8 (2013), 1798–1828.
2. Bielawski, B. Using the convertfrom-string cmdlet to parse structured text. *PowerShell Magazine*, (Sept. 9, 2004); <http://www.powershellmagazine.com/2014/09/09/using-the-convertfrom-string-cmdlet-to-parse-structured-text/>
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka-Jr, E.R. and T.M. Mitchell, T.M. Toward an architecture for never-ending language learning. In *AAAI*, 2010.
4. Chandola, V., Banerjee, A. and V. Kumar, V. Anomaly detection: A survey. *ACM Computing Surveys* 41, 3 (2009), 15.
5. Cypher, A. (Ed). *Watch What I Do: Programming by Demonstration*. MIT Press, Cambridge, MA, 1993.
6. Ferri-Ramírez, C., Hernández-Orallo, J. and Ramírez-Quintana, M.J. Incremental learning of functional logic programs. In *Proceedings of FLOPS*, 2001, 233–247.
7. Flener, P. and Schmid, U. An introduction to inductive programming. *AI Review* 29, 1 (2009), 45–62.
8. Gulwani, S. Dimensions in program synthesis. In *Proceedings of PPDP*, 2010.
9. Gulwani, S. Automating string processing in spreadsheets using input-output examples. In *Proceedings of POPL*, 2011; <http://research.microsoft.com/users/sumitg/flashfill.html>.
10. Gulwani, S. Example-based learning in computer-aided STEM education. *Commun. ACM* 57, 8 (Aug 2014), 70–80.
11. Gulwani, S., Harris, W. and Singh, R. Spreadsheet data manipulation using examples. *Commun. ACM* 55, 8 (Aug. 2012), 97–105.
12. Henderson, R.J. and Muggleton, S.H. Automatic invention of functional abstractions. *Latest Advances in Inductive Logic Programming*, 2012.
13. Hernández-Orallo, J. Deep knowledge: Inductive programming as an answer, Dagstuhl TR 13502, 2013.
14. Hofmann, M. and Kitzelmann, E. I/O guided detection of list catamorphisms—towards problem specific use of program templates in IP. In *ACM SIGPLAN PEPM*, 2010.
15. Jha, J., Gulwani, S., Seshia, S. and Tiwari, A. Oracle-guided component-based program synthesis. In *Proceedings of the ICSE*, 2010.
16. Katayama, S. Efficient exhaustive generation of functional programs using Monte-Carlo search with iterative deepening. In *Proceedings of PRICAI*, 2008.
17. Kitzelmann, E. Analytical inductive functional programming. *LOPSTR 2008, LNCS 5438*. Springer, 2009, 87–102.
18. Kitzelmann, E. Inductive programming: A survey of program synthesis techniques. In *AAIP*, Springer, 2010, 50–73.
19. Kitzelmann, E. and Schmid, U. Inductive synthesis of functional programs: An explanation based generalization approach. *J. Machine Learning Research* 7, (Feb. 2006), 429–454.
20. Kotovsky, K., Hayes, J.R. and Simon, H.A.. Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology* 17, 2 (1985), 248–294.
21. Lau, T.A. Why programming-by-demonstration systems fail: Lessons learned for usable AI. *AI Mag.* 30, 4, (2009), 65–67.
22. Lau, T.A., Wolfman, S.A., Domingos, P. and Weld, D.S. Programming by demonstration using version space algebra. *Machine Learning* 53, 1-2 (2003), 111–156.
23. Le, V. and Gulwani, S. FlashExtract: A framework for data extraction by examples. In *Proceedings of PLDI*, 2014.
24. Lieberman, H. (Ed). *Your Wish is My Command: Programming by Example*. Morgan Kaufmann, 2001.
25. Lin, D., Dechter, E., Ellis, K., Tenenbaum, J.B. and Muggleton, S.H. Bias reformulation for one-shot function induction. In *Proceedings of ECAI*, 2014.
26. Marcus, G.F. *The Algebraic Mind. Integrating Connectionism and Cognitive Science*. Bradford, Cambridge, MA, 2001.
27. Martínez-Plumed, C. Ferri, Hernández-Orallo, J. and M.J. Ramírez-Quintana. On the definition of a general learning system with user-defined operators. *arXiv preprint arXiv:1311.4235*, 2013.
28. Menon, A., Tamuz, O., Gulwani, S., Lampson, B. and Kalai, A. A machine learning framework for programming by example. In *Proceedings of the ICML*, 2013.
29. Miller, R.C. and Myers, B.A. Multiple selections in smart text editing. In *Proceedings of UII*, 2002, 103–110.
30. Muggleton, S.H. Inductive Logic Programming. *New Generation Computing* 8, 4 (1991), 295–318.
31. Muggleton, S.H. and Lin, D. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. *IJCAI 2013*, 1551–1557.
32. Muggleton, S.H., Lin, D., Pahlavi, N. and Tamaddon-Nezhad, A. Meta-interpretive learning: application to grammatical inference. *Machine Learning* 94 (2014), 25–49.
33. Muggleton, S.H., De Raedt, L., Poole, D., Bratko, I., Flach, P. and Inoue, P. ILP turns 20: Biography and future challenges. *Machine Learning* 86, 1 (2011), 3–23.
34. Olsson, R. Inductive functional programming using incremental program transformation. *Artificial Intelligence* 74, 1 (1995), 55–83.
35. Perelman, D., Gulwani, S., Grossman, D. and Provost, P. Test-driven synthesis. *PLDI*, 2014.
36. Raza, M., Gulwani, S. and Milic-Frayling, N. Programming by example using least general generalizations. *AAAI*, 2014.
37. Schmid, U. and Kitzelmann, E. Inductive rule learning on the knowledge level. *Cognitive Systems Research* 12, 3 (2011), 237–248.
38. Schmid, U. and Wyszotki, F. Induction of recursive program schemes. *ECML 1398 LNAI* (1998), 214–225.
39. Shapiro, E.Y. An algorithm that infers theories from facts. *IJCAI* (1981), 446–451.
40. Solar-Lezama, A. *Program Synthesis by Sketching*. Ph.D thesis, UC Berkeley, 2008.
41. Summers, P.D. A methodology for LISP program construction from examples. *J. ACM* 24, 1 (1977), 162–175.
42. Tenenbaum, J.B., Griffiths, T.L. and Kemp, C. Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences* 10, 7 (2006), 309–318.
43. Young, S. Cognitive user interfaces. *IEEE Signal Processing* 27, 3 (2010), 128–140.

Sumit Gulwani (sumitg@microsoft.com) is principal researcher and research manager of a programming-by-example research and engineering group at Microsoft Corp., Redmond, WA.

José Hernández-Orallo (jorallo@dsic.upv.es) is a reader at the Universitat Politècnica de València. He is supported by EU (FEDER) and Spanish projects PCIN-2013-037, TIN 2013-45732-C4-1-P and GV PROMETEOII2015/013.

Emanuel Kitzelmann (ekitzelmann@gmail.com) is a teacher at Adam-Josef-Cüppers Commercial College, Ratingen, Germany.

Stephen H. Muggleton (s.muggleton@imperial.ac.uk) is a professor in the Department of Computing at Imperial College London, U.K.

Ute Schmid (ute.schmid@uni-bamberg.de) is a professor at University of Bamberg, Germany.

Benjamin Zorn (Ben.Zorn@microsoft.com) is principal researcher and research co-manager of the Research in Software Engineering (RISE) group at Microsoft Research in Redmond, WA.

MobileHCI 2016

September 6-9, Florence, Italy



DEADLINES

FEBRUARY 12, 2016

for Full and Short Papers

FEBRUARY 26, 2016

for Workshop proposals

MAY 13, 2016

for Posters, Demos, and Tutorials

GENERAL CHAIRS

Fabio Paternò (CNR-ISTI, Italy)

Kaisa Väänänen (TUT, Finland)

PAPER CHAIRS

Karen Church (Yahoo, USA)

Jonna Hakkila (U. Lapland, Finland)

Antonio Kruger (DFKI, Germany)

Marcos Serrano (U. Toulouse, France)

MobileHCI is sponsored by



Association for
Computing Machinery



SIGCHI

MobileHCI seeks contributions in the form of innovations, insights, or analyses related to human interaction and experiences with mobile devices and services. We seek richness and diversity in topic as well as approach, method, and viewpoint. In no particular order, this includes contributions in: *Systems & Infrastructures*. The design, architecture, deployment, and evaluation of systems and infrastructures that support development of or interaction with mobile devices and services.

Devices & interaction techniques. The design, construction, usage, and evaluation of devices and techniques that create valuable new capabilities for mobile human-computer interaction.

Applications & experiences. Descriptions of the design, empirical study of interactive applications, or analysis of usage trends that leverage mobile devices and systems.

Methodologies & tools. New methods and tools designed for or applied to studying or building mobile user interfaces, applications, and mobile users.

Theories & models. Critical analysis or organizing theory with clearly motivated relevance to the design or study of mobile human-computer interaction; taxonomies of design or devices.

Visions. Well-argued and well-supported visions of the future of mobile computing or non-traditional topics that bear on mobility.

We solicit original research and technical papers not published elsewhere focusing on the following topics (but not limited to):

- Novel user interfaces and interaction techniques
- Mobile social networks and systems
- Design of location-based or context-aware services for mobile devices
- Context-aware systems
- Multimodal interaction (including audio and speech)
- User-centred design tools and methods for mobile systems
- Ethnographical and field studies with mobile technology
- Group interaction and mobility communities
- Services for mobile devices
- Design, evaluation and case studies of mobile applications
- Wearable computing, smart clothes, new devices and sensors
- Design and tools for cross-device user interfaces
- Mobile entertainment, storytelling and location based gaming
- Aesthetic interaction and experience design
- Affective computing in the mobile context
- Personal assistance with mobile devices
- Mobility and work environments
- Evaluation and usability of mobile devices and services
- Mobile accessibility



research highlights

P. 102

**Technical
Perspective
The Path to
Light Transport**

By Frédo Durand

P. 103

**Geometric Tools for
Exploring Manifolds
of Light Transport Paths**

By Wenzel Jakob and Steve Marschner

P. 112

**Technical
Perspective
Treating Networks
Like Programs**

By George Varghese

P. 113

**Software Dataplane
Verification**

By Mihai Dobrescu and Katerina Argyraki

Technical Perspective

The Path to Light Transport

By Frédo Durand

THE INTERACTION OF light with materials results in a rich visual experience that can bring wonder and puzzlement to children and adults alike. Light can follow a complex path before reaching our eyes, reflecting off objects or bending to yield shiny reflections, rainbows, or the bewildering patches of light known as caustics. Physically based rendering seeks to faithfully simulate this complex process through numerical integration and the modeling of light-matter interaction. But it offers more than a formidable technical challenge, it also provides us with a fascinating behind-the-scenes perspective on the genesis of our visual experience.

The difficulty of light simulation stems from the very complexity of the physical process. While the simulation of light reaching objects directly from a light source is simple enough, it only accounts for a fraction of the light illuminating our world. Much light bounces off objects in a recursive way, and can have a significant impact even after many bounces, especially if the objects it reflects off are very shiny or translucent. Initially, computer graphics rendering avoided this complexity and only considered direct illumination, relying on various hacks such as fake additional lights to make the results look acceptable. Most early CGI in movies such as *Toy Story* and *Jurassic Park* did not take into account indirect illumination and relied instead on the artists' ability to fake believable lighting.

Over the last two decades, the field has made tremendous progress in simulating light interreflections for relatively easy cases, in particular when materials are not too shiny. In this case light gets more and more diffuse as it bounces around, which requires limited precision to resolve. Simple forms of physically based rendering are now used in movie production. And for the majority of its catalog, Ikea has now switched to fully photorealistic 3D computer graphics

rendered with global illumination.

However, with the easy aspects of indirect lighting now under control, the hard cases become salient: light reflections involving multiple shiny or transparent objects. If you have ever wondered at a dinner table where the bright patterns focused by a wine glass come from, consider that it is even harder for algorithms to find the proper light path combinations.

To better understand the difficulty of the problem, consider its formal definition. The value of a pixel is the integral of light reaching a given point of the scene after it has been reflected an arbitrary number of times. The domain of integration is the space of light paths, where each path links a series of vertices in the scene. Direct illumination corresponds to a single vertex, the visible point, between a light source and the camera. Indirect illumination can involve two, three, or more vertices. Modern solutions typically involve Monte Carlo approaches and random sampling to approximate the integral at each pixel. The first challenge comes from the high dimensionality of the

The paper builds on the Metropolis Light Transport, which works by performing a random walk through path space carefully engineered to linger near paths in exact proportion to their contribution to the image.

domain, proportional to the number of bounces. The second challenge is caused by shiny materials, refraction, and complex scene geometry, which can make the integrand extremely complex. Current sampling solutions miss these important but hard-to-find paths. For example, caustics are formed when light reflects in just the right direction off a very shiny object, or when it is bent by a refractive material with a given curvature and index of refraction.

The following paper presents a technique to address these challenging problems of light transport by characterizing mathematically the narrow manifolds of high-contribution light paths within the high-dimensional path space. It builds on a Markov Chain Monte Carlo method known as Metropolis Light Transport, which works by performing a random walk through path space that is carefully engineered to linger near paths in exact proportion to their contribution to the image. The Markov chain approach replaces the problem of *finding* an important path starting from nothing with the problem of *modifying* a path to explore high-contribution areas.

The key idea of this paper is to characterize the path-space geometry of these high-contribution areas and to walk in the direction of the ridge (or manifold), not across it. For this, the authors have had to characterize the complex geometry of multi-bounce specular paths, where changes in one vertex location affect the incoming direction of light from the previous vertex as well as outgoing direction to the next one, which means the problem is correlated across a full set of bounces. They have derived equations that enable efficient walks, and the results speak for themselves. 

Frédo Durand (fredo@mit.edu) is a professor of electrical engineering and computer science at the Massachusetts Institute of Technology, Cambridge, MA.

Copyright held by author.

Geometric Tools for Exploring Manifolds of Light Transport Paths

By Wenzel Jakob and Steve Marschner

Abstract

Photorealistic images created using physical simulations of light have become a ubiquitous element of our everyday lives. The most successful techniques for producing such images replicate the key physical phenomena in a detailed software simulation, including the emission of light by sources, transport through space, and scattering in the atmosphere and at the surfaces of objects. Mathematically, this computation involves the approximation of many high-dimensional integrals, one for each pixel of the image, usually using Monte Carlo methods. Although a great deal of progress has been made on rendering algorithms, so that physically based rendering is now routinely used in many applications, commonly occurring situations can still cause these algorithms to become impractically slow, forcing users to make unrealistic scene modifications to obtain satisfactory results.

Light transport is complex because light can flow along a great variety of different paths through a scene, though only a subset of these makes relevant contributions to the final image. The simulation becomes ineffective when it is difficult to find the important paths. Commonly occurring materials like smooth metal or glass surfaces can easily lead to such situations, where only very few lighting paths participate, leading to spiky integrands and poor convergence. How to efficiently handle such cases in general has been a long-standing problem.

In this paper, we provide a geometric solution to this problem by representing light paths as points in an abstract high-dimensional configuration space that is defined by a system of constraint equations. This configuration space is a differentiable manifold, which can be locally parameterized in the neighborhood of an existing path. Building on this framework, we propose Manifold Exploration, a rendering technique that efficiently explores the integration domain by taking geometrically informed steps on the manifold of light paths.

1. INTRODUCTION

The central goal of light transport algorithms in computer graphics is the generation of *renderings*, two-dimensional images that depict a simulated environment as if photographed by a virtual camera. Their applications are diverse and range from predictive visualization in design and architecture to visual effects and fully rendered movie productions. Fueled by the pervasive use of renderings in our society, the demands on their realism have rapidly grown over the years: generally, users of rendering software want their images to look, in various ways, “real,” which has driven the field toward

rendering techniques that are based on the physics of light transport. Coupled with the decreasing cost of computation, these methods have seen substantial adoption over the last two decades.

Physics-based rendering techniques require a detailed model of the scene including the shape and optical properties of all objects including light sources; the final rendering is then generated from a detailed simulation of *transport* and *scattering*, that is, the propagation of light and its interaction with the materials that comprise the objects. Mathematically, this simulation is equivalent to computing a high dimensional integral for every pixel. Due to the challenging nature of these integrals (no analytic solutions, discontinuities and singularities, unbounded number of dimensions), *Monte Carlo* (MC) techniques are in most cases the only available tool for evaluating them. The details of this computation will be discussed shortly in Section 1.1.

The function to be integrated for each pixel is defined in terms of the input scene description provided by the user. Certain input, particularly scenes containing metal, glass, or other shiny surfaces, can lead to integrals that are difficult to evaluate. Depending on the roughness of the surfaces, the integrand can take on large values over small regions of the integration domain. Surfaces of lower roughness lead to smaller and higher-valued regions, which eventually collapse to lower-dimensional sets with singular integrands as the surface roughness tends to zero.

MC techniques evaluate the integrand at many randomly chosen points throughout the integration domain, and the result of the integral is then approximated by the average of these samples. Convergence problems arise whenever high-valued regions receive too few samples. Depending on the method used, this manifests as objectionable noise or other visual artifacts in the output image that gradually disappear as the sample count N tends to infinity. However, due to the slow convergence rate of MC integration (typical error is $O(N^{-0.5})$), it may not be an option to wait for the error to average out. Such situations can force users of rendering software to make unrealistic scene modifications (e.g., disabling certain light interactions), thereby compromising realism in exchange for obtaining converged-looking results within a reasonable time.

The original version of this paper is entitled “Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport” and was published in the *Proceedings of SIGGRAPH 2012*.

In this paper, we propose to address these convergence problems, using a geometric approach to efficiently place samples in these problematic regions. Focusing at first on the ideally smooth case, we interpret the high-valued regions of the integrand as implicitly defined manifolds that can be locally parameterized with a simple equation-solving iteration. We then build on this parameterization as part of a new *Markov Chain Monte Carlo* (MCMC) technique to enable a more systematic exploration of the important parts of the integration domain.

This article covers background material and explains the underlying theory but can only provide a high-level overview of the actual algorithm. We refer the reader to the original paper for an in-depth discussion with technical details.

Before discussing the specifics of our approach, we review relevant background material, starting with the standard light transport model used in computer graphics and leading up to the path space framework proposed by Veach.¹¹

1.1. Geometric optics and light paths

Light transport simulations in computer graphics are generally conducted in the setting of geometric optics, that is, without considering the wave nature of light. This is an excellent approximation due to the large size of normal scenes compared to the wavelength of visible light (380–700 nm). It is also common to ignore the effects of polarization. Without loss of generality, we furthermore assume monochromatic (non-colored) illumination to simplify the definitions.

To create the output image, the rendering algorithm must determine how much of the illumination emitted at the light sources arrives at the camera. In the geometric optics framework discussed above, light moves along straight lines until an interaction (i.e., a scattering event) changes the direction of propagation. Two types of interactions are possible: scattering at a *surface*, and scattering in a *volume* that fills the region between surfaces. We only discuss the surface case here—extensions of the proposed technique to volumes are possible⁶ but beyond the scope of this article.

The combination of light propagation along straight lines and changes of direction at surfaces leads to the concept of a *light path*, which can be thought of as the trajectory of a particle carrying an infinitesimal portion of the illumination. A light path is a piecewise linear curve $\bar{x} = \mathbf{x}_1 \dots \mathbf{x}_n$ with endpoints $\mathbf{x}_1$ and $\mathbf{x}_n$ and intermediate scattering vertices $\mathbf{x}_2, \dots, \mathbf{x}_{n-1}$. The space of all possible light paths is a union consisting of paths with just the endpoints, paths that have one intermediate scattering event, and so on. More formally,

$$\mathcal{P} = \bigcup_{n=2}^{\infty} \mathcal{P}_n, \quad (1)$$

$$\mathcal{P}_n = \{\mathbf{x}_1 \dots \mathbf{x}_n \mid \mathbf{x}_1, \dots, \mathbf{x}_n \in \mathcal{M}\}, \quad (2)$$

where the set $\mathcal{M}$ is the union of all surfaces that make up the scene. In practice, we are only interested in a small subset of $\mathcal{P}$, namely those paths which start on a light source and end on the aperture of the camera. However, to compute the total amount of illumination arriving at the sensor, this binary classification is not enough: because some paths carry more

illumination from the light source to the camera than others, we furthermore require a weighting function $\varphi : \mathcal{P} \rightarrow \mathbb{R}$ which quantifies this “light-carrying capacity.” The total illumination I arriving at the camera is then given by a weighted integral over light paths, that is,

$$I = \int_{\mathcal{P}} \varphi(\bar{x}) dA(\bar{x}) = \int_{\mathcal{P}_2} \varphi(\mathbf{x}_1 \mathbf{x}_2) dA(\mathbf{x}_1, \mathbf{x}_2) + \int_{\mathcal{P}_3} \varphi(\mathbf{x}_1 \mathbf{x}_2 \mathbf{x}_3) dA(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) + \dots, \quad (3)$$

where the first integral over $\mathcal{P}$ is simply compact notation for the sum of many separate integrals over the subsets $\mathcal{P}_k$ of $\mathcal{P}$, and dA is the surface area product measure.

The definition of the weighting function φ depends on the number of vertices of the input light path and always consists of a product of terms—one for each vertex and edge of the path:

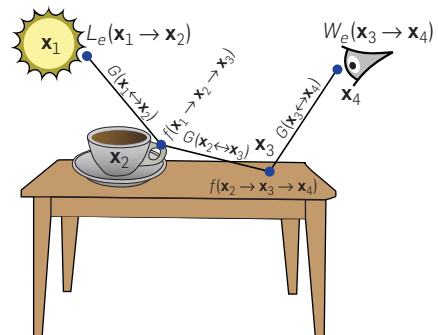
$$\begin{aligned} \varphi(\mathbf{x}_1 \dots \mathbf{x}_n) &= L_e(\mathbf{x}_1 \rightarrow \mathbf{x}_2) \\ &\left[\prod_{k=2}^{n-1} G(\mathbf{x}_{k-1} \leftrightarrow \mathbf{x}_k) f(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k \rightarrow \mathbf{x}_{k+1}) \right] \\ &G(\mathbf{x}_{n-1} \leftrightarrow \mathbf{x}_n) W_e(\mathbf{x}_{n-1} \rightarrow \mathbf{x}_n). \end{aligned} \quad (4)$$

Figure 1 shows an example light path and the different weighting terms. These play the following roles:

- $L_e(\mathbf{x}_1 \rightarrow \mathbf{x}_2)$ is the emission profile of the light source, where the arrow indicates the direction of propagation. This term expresses the amount of light emitted from position $\mathbf{x}_1$ traveling towards $\mathbf{x}_2$. It is equal to zero when $\mathbf{x}_1$ is not located on a light source.
- $W_e(\mathbf{x}_{n-1} \rightarrow \mathbf{x}_n)$ is the sensitivity profile of the camera; it is equal to zero when $\mathbf{x}_n$ is not located on the aperture. To separate the illumination received by each pixel, we can think of every pixel j as an independent sensor with sensitivity profile $W_e^{(j)}$.
- $G(\mathbf{x} \leftrightarrow \mathbf{y})$ is the *geometric term*, which specifies the differential amount of illumination carried along segments of the light path. The two-sided arrows denote the symmetric nature of this function. Among other things, it accounts for visibility: when there is no unobstructed

Figure 1. Visualization of a simple light path with four vertices and its corresponding weighting function.

$$\begin{aligned} \varphi(\bar{x}) &= L_e(\mathbf{x}_1 \rightarrow \mathbf{x}_2) G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_2) f(\mathbf{x}_1 \rightarrow \mathbf{x}_2 \rightarrow \mathbf{x}_3) G(\mathbf{x}_2 \leftrightarrow \mathbf{x}_3) \\ &f(\mathbf{x}_2 \rightarrow \mathbf{x}_3 \rightarrow \mathbf{x}_4) G(\mathbf{x}_3 \leftrightarrow \mathbf{x}_4) W_e(\mathbf{x}_3 \rightarrow \mathbf{x}_4) \end{aligned}$$



line of sight between $\mathbf{x}$ and $\mathbf{y}$, G evaluates to zero.

- $f(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k \rightarrow \mathbf{x}_{k+1})$ specifies how much of the light that travels from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$ is then scattered towards position $\mathbf{x}_{k+1}$. This function essentially characterizes the material appearance of an object (e.g., whether it is made of wood, plastic, concrete, etc.).

Considerable research over the last 40 years has investigated realistic expressions for the L_e , W_e , and f terms. In this article, we do not discuss their definition and prefer to think of them as black box functions that can be queried by the rendering algorithm. This is similar to how rendering software is implemented in practice: a scene description might reference a particular material (e.g., car paint) whose corresponding function f is implemented in an external shared library. This library is then loaded and made available to the rendering algorithm, which accesses it through a high-level interface shared by all materials (but without specific knowledge about its internal structure).

1.2. Light path regular expressions

Different materials can interact with light in fundamentally different ways, which has important implications on the design of rendering algorithms. It is helpful to distinguish between interactions using a 1-letter classification for each type of path vertex:

- S (*ideal specular*): specular surfaces indicate boundaries between materials with different indices of refraction (e.g., air and water). Ideal specular boundaries have no roughness and cause and incident ray of light to be scattered into a *discrete* set of outgoing directions (Figure 2). Examples of specular materials include polished glass and metal surfaces and smooth coatings.
- G (*glossy*): glossy surfaces also mark an index of refraction transition, but in this case the surface is affected by small-scale roughness. This causes the same ray to scatter into a continuous distribution of directions which concentrates around the same directions as the ideally smooth case.
- D (*diffuse*): diffuse surfaces reflect light into a directional distribution that is either uniform or close to uniform; examples include clay and plaster.

We additionally assign the labels L and E to light source and camera (“eye”) vertices, respectively, allowing classification of entire light paths using a sequence of symbols (e.g.,

“LSSDDE”). Larger classes of paths can be described using Heckbert’s *path regular expressions*,³ which add convenient regular expression rules such as the Kleene star “*” and plus “+” operators. For instance, LD+E refers to light that has been scattered only by diffuse surfaces before reaching the camera. We will use this formalism shortly.

1.3. Light path sampling techniques

The objective of a rendering algorithm is the efficient approximation of Equation (3) given a model of the scene that defines the set of surfaces $\mathcal{M}$ and a path weighting function $\varphi(\bar{\mathbf{x}})$. As discussed earlier, this is generally done using MC integration, which entails randomly generating many different paths $\bar{\mathbf{x}}_i$ and computing averages of their corresponding weights $\varphi(\bar{\mathbf{x}}_i)$, yielding an *unbiased* estimator, whose expected value equals the desired answer. *Biased* estimators can achieve lower errors in some situations—however, these methods are beyond the scope of this article.

A naïve application of the MC principle might look as follows: to estimate the integral for paths of length n in Equation (3), randomly sample n positions distributed with respect to surface area on $\mathcal{M}$. These can be concatenated to form a light path and passed as an argument to the function φ . A running average of these evaluations should then converge to the right answer. However, this approach does not work well for many reasons: most paths sampled in this way would not carry appreciable amounts of energy from the light source to the camera, for example, because some of the path edges pass through occluders that block the illumination.

To increase the probability of sampling relevant light paths, it is therefore common to use *importance sampling* techniques that focus more computation on regions of the integration domain where “good” samples are likely to be found.

The most popular importance sampling techniques in rendering construct light paths one vertex at a time (we will refer to this as *sequential* sampling). In each iteration, they randomly sample the position of an additional light path vertex $\mathbf{x}_i$ with a probability density that is proportional to the (partial) weighting function $\varphi(\mathbf{x}_0 \dots \mathbf{x}_{i-1} \mathbf{x}_i \dots)$ containing only factors depending on $\mathbf{x}_0 \dots \mathbf{x}_i$. The resulting algorithm behaves in a physically intuitive way: each iteration computes the trajectory of an idealized light “particle” that moves through the scene.

Figure 3 illustrates the behavior of several variants of this approach when rendering *caustics*, which we define as light paths matching the regular expression LS+DS+E. They form interesting light patterns at the bottom of the swimming pool due to focusing effect of ripples in the water surface.

In Figure 3a, particles are emitted proportional to the light source emission profile L_e . The highlighted path is the trajectory of a particle that encounters the water surface and refracts into the pool. The refraction is an ideal specular interaction described by Snell’s law and the Fresnel equations. The diffuse concrete surface at the pool bottom then reflects the particle upwards into a direction drawn from a uniform distribution, where it is refracted once more by the water surface. Ultimately, the particle never hits the camera aperture and thus cannot contribute to the output image.

Figure 2. Ideal specular materials scatter an incident ray of light (red) into a discrete set of outgoing directions (blue, green). Rough specular materials scatter the same ray into a continuous distribution of directions.

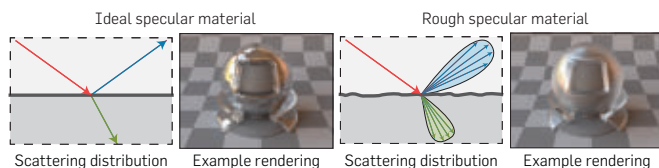
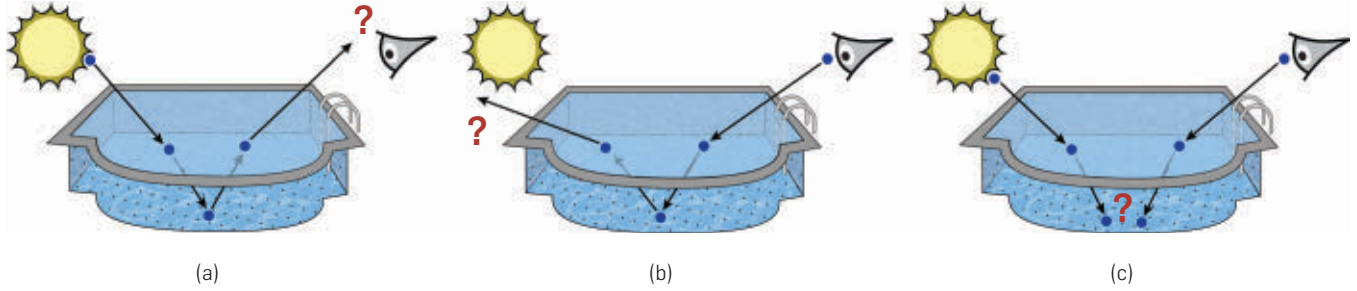


Figure 3. Visual illustration of the difficulties of sequential path sampling methods when rendering caustic patterns at the bottom of a swimming pool. (a, b) Unidirectional techniques sample light paths by executing a random walk consisting of alternating transport and scattering steps. The only way to successfully complete a path in this manner is to randomly “hit” the light source or camera, which happens with exceedingly low probability. (c) Bidirectional techniques trace paths from both sides, but in this case they cannot create a common vertex at the bottom of the pool to join the partial light paths.



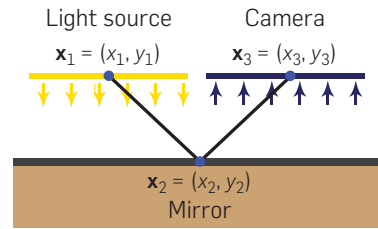
Paths can also be generated in the opposite way, starting from the camera and working towards a light source (Figure 3b). This is the basic approach of the *path tracing* algorithm proposed by Kajiya.⁷ While vastly superior to the uniform area sampling approach discussed earlier, both path tracing variants remain extremely inefficient in this scene: in order to construct a complete light path $\bar{x}$ with $\varphi(\bar{x}) > 0$, the path must reach the “other end” (camera or light source) by chance, which happens with exceedingly low probability: assuming, for example, that rays leave the pool with a uniform distribution in Figure 3b, the probability of hitting the sun with an angular diameter of $\sim 0.5^\circ$ is about 10^{-5} .

*Bidirectional path tracing*¹¹ combines the previous two techniques by tracing partial light paths from both sides and creating connections between them (Figure 3c), which works in many scenes where the two uni-directional methods fail. Yet, even this approach is impractical here: vertices on the water surface cannot be used to join two partial paths, since the resulting pair of incident and outgoing directions would not satisfy Snell’s law. It is possible to generate two vertices at the bottom of the pool as shown in the figure, but these cannot be connected: the resulting path edge would be fully contained in a surface rather than representing transport *between* surfaces.

The main difficulty in scenes like this is that caustic paths are tightly constrained: they must start on the light source, end on the aperture, and satisfy Snell’s law in two places. Sequential sampling approaches are able to satisfy all but one constraint and run into issues when there is no way to complete the majority of partially generated paths.

Paths like the one examined in Figure 3 lead to poor convergence in other settings as well; they are collectively referred to as *specular-diffuse-specular* (SDS) paths due to the occurrence of this sequence of interactions in their path classification. SDS paths occur in common situations such as a tabletop seen through a drinking glass standing on it, a bottle containing shampoo or other translucent liquid, a shop window viewed and illuminated from outside, as well as scattering inside the eye of a virtual character. Even in scenes where these paths do not cause dramatic effects, their presence can lead to excessively slow convergence in rendering algorithms that attempt to account for

Figure 4. A motivating example in two dimensions: specular reflection in a mirror.



all transport paths. It is important to note that while the SDS class of paths is a well-studied example case, other classes (e.g., involving glossy interactions) can lead to many similar issues. We are interested in algorithms that generalize to these situations.

To approach these challenges, this work proposes a new theoretical framework and algorithms that enable a more systematic exploration of the integration domain, driven by geometric insights about the structure of these problematic paths. The main difference with respect to sequential sampling approaches is that our method samples complete chains of specular interactions at once, permitting a more comprehensive view of the underlying constraints.

2. SPECULAR MANIFOLDS

When a scene contains ideal specular materials, these act like constraints that collapse some dimensions of the space of light paths, leaving behind a lower-dimensional manifold embedded in path space.

We illustrate this using a simple demonstration in two dimensions, in which a camera observes a planar light source through an opposing mirror (Figure 4). We will refer to a light path joining two endpoints through a sequence of k ideal specular scattering events as a *specular chain* of length k . A specular chain of length 1 from the light source to the camera is shown in the figure. The space of all such chains can be parameterized using the horizontal coordinates x_1 , x_2 , and x_3 (we assume that the vertical coordinates are fixed).

Reflections in the mirror must satisfy the law of specular reflection; in this simple scene, it simply states that

$$x_2 = \frac{x_1 + x_3}{2}, \quad (5)$$

that is, the x coordinate of the second vertex must be exactly half-way between the endpoints. Note that this equation can also be understood as the implicit definition of a plane in $\mathbb{R}^3$ ($x_1 - 2x_2 + x_3 = 0$).

When interpreting the set of all candidate light paths as a three-dimensional space $\mathcal{P}_3$ of coordinate tuples (x_1, x_2, x_3) , this constraint then says that the subset of relevant paths has one fewer dimension and is given by the intersection of $\mathcal{P}_3$ with the plane equation (5). With this extra knowledge, it is now easy to sample valid specular chains, for example, by generating x_1 and x_3 and solving for x_2 .

Given general non-planar shapes, the problem unfortunately becomes considerably more complicated, since the equations that have to be satisfied are nonlinear and may admit many solutions. Prior work has led to algorithms that can find solutions even in such cases: Mitchell and Hanrahan⁹ devised a method to render caustics due to implicitly defined reflectors using interval Newton's method. Walter et al.¹³ proposed a related method that computes the singly scattered radiance within a refractive object with triangle mesh boundaries.

Both of these methods perform a global search over a two-dimensional space to locate all length-1 chains connecting two given endpoints, which becomes infeasible for longer specular chains ($k > 1$) that require a 2^k dimensional search. Both methods are also closely tied to the representation of the underlying geometry (implicitly defined surfaces in one case and triangle meshes in the other).

Like these works, our method finds valid specular chains. But because it does so within the *neighborhood* of a given path, it avoids the complexities of a full global search, and it can work with a wide range of geometric representations. Another difference is that we use our manifold formalism to build a fully general rendering system that is not limited to rendering specular paths in isolation.

Two related prior works have also investigated the local structure of light paths: the widely used method of ray differentials⁴ tracks the footprint of a bundle of rays emanating from a pixel through specular interactions to perform texture antialiasing.

The analysis of reflection geometry presented by Chen and Arvo¹ is closest to the mathematics underlying our approach. Using Fermat's principle, the authors derive a second-order expansion of the neighborhood of a path to accelerate the interactive display of reflections on curved surfaces. However, the application of this expansion and the goals of the research are entirely different: in their case, estimating changes to viewing paths, and in our case, tracking the evolution of specular chains in a very general context, as part of an unbiased MC rendering algorithm.

2.1. Integrals over specular manifolds

Let us return to our previous example of the swimming pool involving the family of light paths LSDSE. These paths belong to the $\mathcal{P}_5$ component of $\mathcal{P}$, which is a 10-dimensional space (two dimensions for each surface position). As we will see shortly, the paths that contribute have to

satisfy two constraint equations involving unit directions in $\mathbb{R}^3$ (which each have 2 degrees of freedom). This constrains a total of four dimensions of the path, meaning that all contributing paths lie on a manifold of dimension 6 embedded in $\mathcal{P}_5$.

The integral in Equation (3) is more naturally expressed as an integral over this *specular manifold* S , rather than as an integral over the entire path space:

$$\int_S \varphi(\mathbf{x}_1 \cdots \mathbf{x}_5) dA(\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5).$$

Note the absence of the specular vertices $\mathbf{x}_2$ and $\mathbf{x}_4$ in the integral's area product measure. The contribution function φ still has the same form, a product of terms corresponding to vertices and edges of the path, but singular reflection functions at specular vertices are replaced with (unitless) specular reflectance values, and the geometric terms are replaced by generalized geometric terms over specular chains that we will denote $G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_2 \leftrightarrow \mathbf{x}_3)$ and $G(\mathbf{x}_3 \leftrightarrow \mathbf{x}_4 \leftrightarrow \mathbf{x}_5)$.

The standard geometric term $G(\mathbf{x} \leftrightarrow \mathbf{y})$ for a non-specular edge computes the area ratio of an (infinitesimally) small surface patch at one vertex and its projection onto projected solid angles as seen from the other vertex (Figure 5). The generalized geometry factor is defined analogously: the ratio of solid angle at one end of the specular chain with respect to area at the other end of the chain, considering the path as a function of the positions of the endpoints. Details on its computation can be found in the original article.

2.2. Constraints for reflection and refraction

Equation (5) introduced a simple specular reflection constraint for axis-aligned geometry in two dimensions. We now generalize this constraint to arbitrary geometry in three dimensions and to both specular reflection and refraction.

Recall the law of specular reflection, which states that incident and outgoing directions make the same angle with the surface normal. Furthermore, all three vectors must be coplanar (Figure 6). We use an equivalent reformulation of this law, which states that the *half direction vector* of the incident and outgoing direction ω_i and ω_o , defined as

Figure 5. The geometric term (left) and the generalized geometric term (right) are both derivatives of projected solid angle at one end with respect to area at the far end.

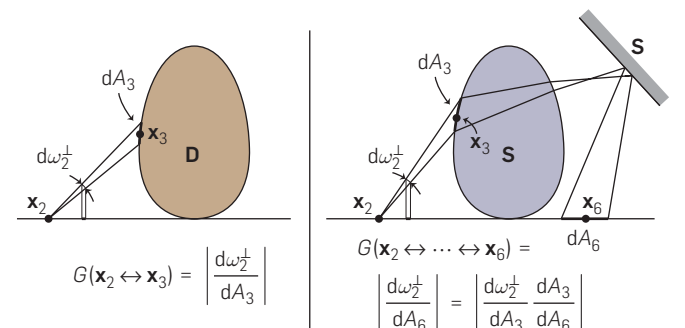
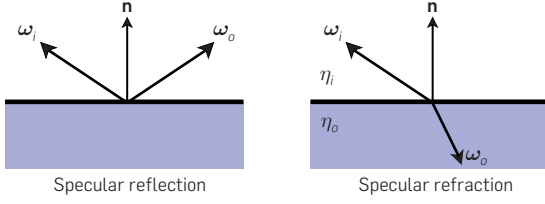


Figure 6. In-plane view of the surface normal ($\mathbf{n}$) and incident (ω_i) and outgoing (ω_o) directions of a surface marking a transition between indices of refraction η_i and η_o .



$$h(\omega_i, \omega_o) = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|} \quad (6)$$

is equal to the surface normal, that is, $h(\omega_i, \omega_o) = \mathbf{n}$. In the case of refraction, the relationship of these directions is explained by Snell's law. Using a generalized definition of the half direction vector which includes weighting by the incident and outgoing indices of refraction¹²; that is,

$$h(\omega_i, \omega_o) = \frac{\eta_i \omega_i + \eta_o \omega_o}{\|\eta_i \omega_i + \eta_o \omega_o\|}, \quad (7)$$

we are able to use a single constraint $h(\omega_i, \omega_o) = \pm \mathbf{n}$ which subsumes both Snell's law and the law of specular reflection (in which case η_i equals η_o). Each specular vertex $\mathbf{x}_i$ of a path $\bar{\mathbf{x}}$ must satisfy this generalized constraint involving its own position and the positions of the preceding and following vertices. Note that this constraint involves unit vectors with only *two* degrees of freedom. We project (7) onto a two-dimensional subspace to reflect its dimensionality:

$$\mathbf{c}_i(\bar{\mathbf{x}}) = T(\mathbf{x}_i)^T h(\overrightarrow{\mathbf{x}_i \mathbf{x}_{i-1}}, \overrightarrow{\mathbf{x}_i \mathbf{x}_{i+1}}), \quad (8)$$

The functions $\mathbf{c}_i : \mathcal{P} \rightarrow \mathbb{R}^2$ compute the generalized half-vector at vertex $\mathbf{x}_i$ and project it onto the tangent space of the underlying scene geometry at this position, which is spanned by the columns of the matrix $T(\mathbf{x}_i) \in \mathbb{R}^{3 \times 2}$. The resulting 2-vector is zero when $h(\omega_i, \omega_o)$ is parallel to the normal. The specular manifold is then simply the set

$$\mathcal{S} = \{\bar{\mathbf{x}} \in \mathcal{P} \mid \mathbf{c}_i(\bar{\mathbf{x}}) = \mathbf{0} \text{ if } \mathbf{x}_i \text{ is specular}\}. \quad (9)$$

2.3. Local manifold geometry

The complex nonlinear behavior of $\mathcal{S}$ severely limits our ability to reason about its geometric structure globally. In this section, we therefore focus on *local* properties, leading to an explicit expression for the tangent space at any point on the manifold. Later, this will constitute the key geometric information needed to construct a numerical procedure that is able to move between points on the manifold.

For simplicity, we restrict ourselves to the case of a single specular chain $\bar{\mathbf{x}} = \mathbf{x}_1 \dots \mathbf{x}_k$ with $k - 2$ specular vertices and non-specular endpoints $\mathbf{x}_1$ and $\mathbf{x}_k$, which suffices to cover most cases by separate application to each specular chain along a path; additional cases are discussed in the original article. To analyze the geometry locally, we require a point in $\mathcal{S}$, that is, a light path $\bar{\mathbf{x}}$ satisfying all specular constraints, to be given.

We assume that local parameterizations of the surfaces in the scene on small neighborhoods around every vertex are provided via functions $\hat{\mathbf{x}}_i(u_i, v_i) : \mathbb{R}^2 \rightarrow \mathcal{M}$, where $\hat{\mathbf{x}}_i(0, 0) = \mathbf{x}_i$. We can then express the constraints $\mathbf{c}_i$ in terms of these local coordinates and stack them on top of each other to create a new function $\hat{\mathbf{c}}$ with signature $\hat{\mathbf{c}} : \mathbb{R}^{2k} \rightarrow \mathbb{R}^{2k-4}$, which maps $2k$ local coordinate values to $2k - 4 = 2(k - 2)$ projected half direction vector coordinates—two for each of the specular vertices of the chain. The set

$$\mathcal{S}_{\text{loc}} = \{(u_1, v_1, \dots, u_k, v_k) \in \mathbb{R}^{2k} \mid \hat{\mathbf{c}}(\dots) = \mathbf{0}\}, \quad (10)$$

then describes the (four-dimensional) specular manifold in terms of local coordinates around the path $\bar{\mathbf{x}}$, which is identified with the origin. Under the assumption that the Jacobian of $\hat{\mathbf{c}}$ has full rank (more on this shortly), the Implicit Function Theorem¹⁰ states that the implicitly defined manifold (10) can be converted into the (explicit) graph of a function $\mathbf{q} : \mathbb{R}^4 \rightarrow \mathbb{R}^{2k-4}$ on an epsilon ball $\mathcal{B}_4(\varepsilon)$ around the origin. Different functions $\mathbf{q}$ are possible—in our case, the most useful variant determines the positions of all the specular vertices from the positions of the non-specular endpoints, that is,

$$\mathcal{S}'_{\text{loc}} = \{(u_1, v_1, \mathbf{q}(u_1, v_1, u_k, v_k), u_k, v_k) \mid (u_1, v_1, u_k, v_k) \in \mathcal{B}_4(\varepsilon)\}. \quad (11)$$

Unfortunately, the theorem does not specify *how* to compute $\mathbf{q}$ —it only guarantees the *existence* of such a function. It does, however, provide an explicit expression for the derivative of $\mathbf{q}$, which contains all information we need to compute a basis for the tangent space at the path $\bar{\mathbf{x}}$, which corresponds to the origin in local coordinates. This involves the Jacobian of the constraint function $\nabla \hat{\mathbf{c}}(\mathbf{0})$, which is a matrix of $k - 2$ by $k - 2$ blocks with a block tridiagonal structure (Figure 7).

If we block the derivative $\nabla \hat{\mathbf{c}}$, as shown in the figure, into 2-column matrices $\mathbf{B}_1$ and $\mathbf{B}_k$ for the first and last vertices and a square matrix $\mathbf{A}$ for the specular chain, then the tangent space to the manifold in local coordinates is

$$T_{\mathcal{S}}(\bar{\mathbf{x}}) = -\mathbf{A}^{-1} [\mathbf{B}_1 \quad \mathbf{B}_k]. \quad (12)$$

This matrix is $k - 2$ by 2 blocks in size, and each block gives the derivative of one vertex with respect to one endpoint.

This construction computes tangents with respect to a graph parameterization of the manifold, which is guaranteed to exist for a suitable choice of independent variables. Because we always use the endpoint vertices for this purpose, difficulties arise when one of the endpoints is located exactly at the fold of a caustic wavefront, in which case $\nabla \hat{\mathbf{c}}$ becomes rank-deficient and $\mathbf{A}$ fails to be invertible. This happens rarely in practice and is not a problem for our method, which allows for occasional parameterization failures. In other contexts where this is not acceptable, the chain could be parameterized by a different pair of vertices when a non-invertible matrix is detected.

These theoretical results about the structure of the specular manifold can be used in an algorithm to solve for specular paths, which we discuss next.

Figure 7. The linear system used to compute the tangent space and its interpretation as a derivative of a specular chain. (a) An example path, (b) Associated constraints, (c) Constraint Jacobian, and (d) Tangent space.

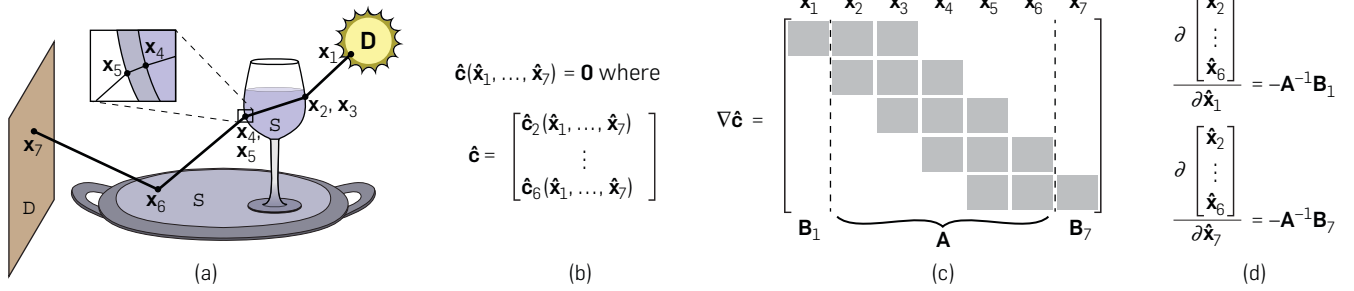
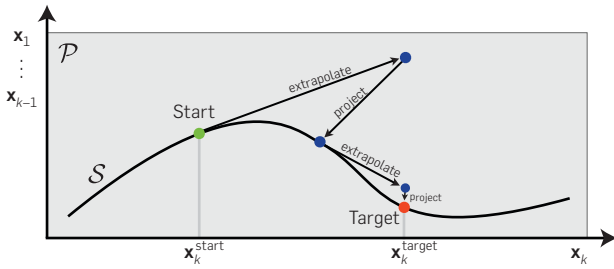


Figure 8. We use a Newton-like iteration to locally parameterize the specular manifold.



2.4. Walking on the specular manifold

In practice, we always keep one endpoint fixed (e.g., x_1), while parameterizing the remaining two-dimensional set. Figure 8 shows a conceptual “graph” of the manifold of a specular chain that is parameterized by the last vertex x_k . This vertex is initially located at x_k^{start} , and we search for a valid configuration where it is at position x_k^{target} . The derivation in Section 2.3 provides a way of extrapolating the necessary change of $x_2, \dots, x_{k-1}$ to first order, but this is not enough: an expansion, no matter to what order, will generally not be able to find a valid path that is located on the manifold S .

To address this issue, we combine the extrapolation with a simple projection operation, which maps approximate paths back onto S by intersecting the extrapolated ray $x_1 \rightarrow x_2$ with the scene geometry and using the appropriate laws of reflection and refraction to compute the remaining vertex locations. The combination of extrapolation and projection behaves like Newton’s method, exhibiting quadratic convergence near the solution; details on this iteration can be found in the original paper.

2.5. Applications in the context of MCMC

We previously discussed MC techniques, which sample large numbers of independent light paths. MCMC methods introduced to graphics with the seminal *Metropolis Light Transport* (MLT) algorithm¹¹ provide an interesting departure from this approach: MLT maintains a valid path as an invariant and generates *correlated* samples by applying random modifications to it.

Figure 9. Example of a novel MCMC path perturbation strategy enabled by our specular manifold walk.

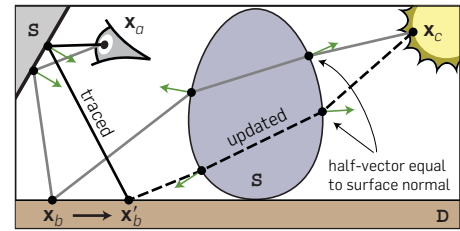


Figure 9 shows a sketch of how manifold walks can be used in this setting: a MCMC algorithm modifies a path by perturbing the outgoing direction at vertex x_a . Propagating this direction through a specular reflection leads to a modified position x_b' on a diffuse surface. To complete the partial path, it is necessary to find a specular chain connecting x_b' to the light source. Here, we can simply apply a manifold walk to the existing specular chain $x_b \dots x_c$ to solve for an updated configuration $x_b' \dots x_c$. The key observation is that MCMC explores the space of light paths using localized steps, which is a perfect match for our local parameterization of the specular manifold.

2.6. Glossy materials

All derivations so far focused on ideal specular materials, but it is straightforward to extend our approach to glossy materials as well. Figure 10 shows a high level sketch of this generalization. In the ideal specular case, there is a single specular chain (or discrete set) connecting x_b and x_c (top left), and all energy is concentrated on a lower-dimensional specular manifold defined by $c(\mathbf{x}) = \mathbf{0}$ (top right).

In the glossy case, there is a continuous family of chains connecting x_b and x_c (bottom left), and the space of light paths has its energy concentrated in a thin “band” *near* the specular manifold. The key idea of how we handle glossy materials is to take steps along a family of parallel offset manifolds $c(\mathbf{x}) = \mathbf{k}$ (bottom right) so that path space near the specular manifold can be explored without stepping out of this thin band of near-specular light transport.

3. RESULTS

We implemented prior work and a new manifold-walk-based MCMC technique as extension modules to the Mitsuba renderer⁵ and conducted a series of equal-time renderings of challenging scenes. We compare against Primary sample space MLT⁸ (PSSMLT), Path space MLT¹¹ (MLT), and Energy redistribution path tracing² (ERPT). Detailed statistics and additional comparisons are available in the supplementary material of the original paper.

Figures 11 and 12 show the resulting renderings of an interior scene containing approximately 2 million triangles with shading normals and a mixture of glossy, diffuse, and specular surfaces and some scattering volumes. We rendered on single Amazon EC2 cc1.4xlarge instances (eight-core Intel Xeon X5570 machines) and allotted 1 hour of rendering time to each technique; the results are intentionally unconverged to permit a visual analysis of the convergence behavior.

Figure 10. Sketch of our generalization to glossy materials.

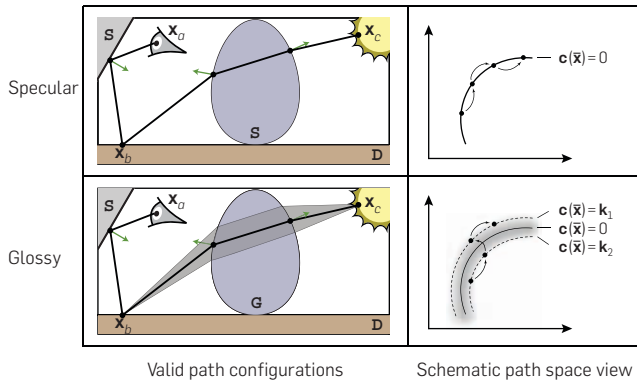


Figure 11. TABLE: This interior scene shows chinaware, a teapot containing an absorbing medium, and a butter dish on a glossy silver tray. Illumination comes from a complex chandelier with glass-enclosed bulbs. Prior methods have difficulty finding and exploring relevant light paths, which causes noise and other convergence artifacts. Equal-time renderings (1 h). (a) MLT, (b) ERPT, (c) PSSMLT, and (d) Our method.

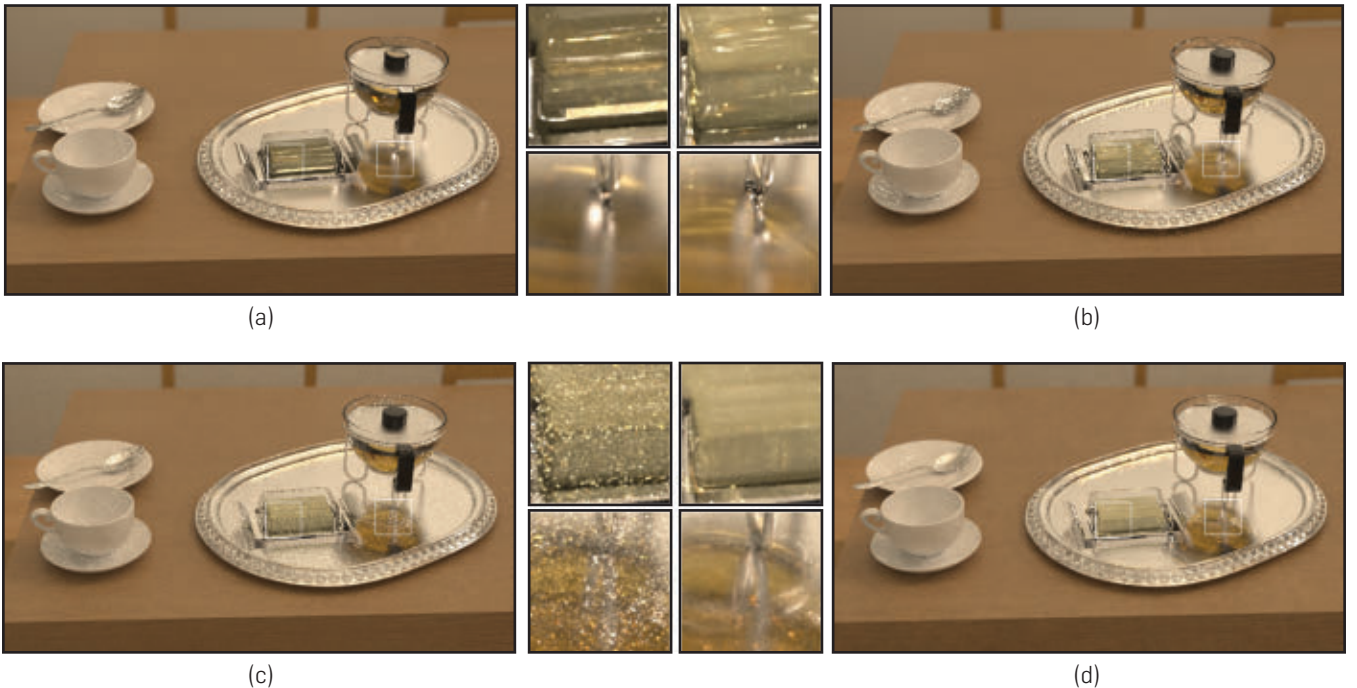


Figure 12. GLASSEGG: This view of a different part of the room, now lit through windows using a spherical environment map surrounding the scene, contains an anisotropic scattering medium inside the glass egg. Equal-time renderings (1h). (a) MLT, (b) ERPT, (c) PSSMLT, and (d) Our method.



showed how to implicitly define and locally parameterize these manifolds using a simple root-finding iteration, providing a powerful set of tools for a more efficient exploration of path space in the context of MCMC integration.

This new algorithm shares certain limitations with its predecessors. Most importantly, it needs well distributed seed paths because it can only explore connected components of the manifold for which seed paths are provided. As the number of connected components exceeds the number of samples that can be generated, local exploration of path space becomes ineffective; how to efficiently explore path space in such cases is still an unsolved problem.

Acknowledgments

This research was conducted in conjunction with the Intel Science and Technology Center for Visual Computing. Additional funding was provided by the National Science Foundation under grant IIS-1011919. The authors are indebted to Olesya Isaenko, who meticulously crafted the example scenes used in the evaluation. The authors also thank John Hughes for long-ago conversations that sparked the idea of manifolds of specular paths. 

References

- Chen, M., Arvo J. Theory and application of specular path perturbation. *ACM Trans. Graph.* 19, 4 (Oct. 2000), 246–278.
- Cline, D., Talbot, J., Egbert P. Energy redistribution path tracing. *ACM Trans. Graph.* 24, 3 (Aug. 2005), 1186–1195.
- Heckbert, P.S. Adaptive radiosity textures for bidirectional ray tracing. *SIGGRAPH Comput. Graph.* 24, 4 (Sept. 1990), 145–154. DOI: 10.1145/97880.97895 <http://doi.acm.org/10.1145/97880.97895>.
- Igehy, H. Tracing ray differentials. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99)* (1999), ACM Press/Addison-Wesley Publishing Co., New York, NY, 179–186. DOI: 10.1145/311535.311555 <http://dx.doi.org/10.1145/311535.311555>.
- Jakob, W. Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>.
- Jakob, W. Light transport on path-space manifolds. PhD thesis, Cornell University (Aug. 2013).
- Kajiya, J.T. The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150. DOI: 10.1145/15886.15902 <http://doi.acm.org/10.1145/15886.15902>.
- Kelemen, C., Szirmay-Kalos, L., Antal, G., Csonka, F. A simple and robust mutation strategy for the metropolis light transport algorithm. *Computer Graphics Forum* 21, 3 (2002), 531–540.
- Mitchell, D., Hanrahan, P. Illumination from curved reflectors. *SIGGRAPH Comput. Graph.* 26, 2 (Jul. 1992), 283–291. DOI: 10.1145/142920.134082 <http://doi.acm.org/10.1145/142920.134082>.
- Spivak, M. *Calculus on Manifolds*. Addison-Wesley, Boston, MA, 1965.
- Veach, E. Robust Monte Carlo methods for light transport simulation. PhD thesis, Stanford University, 1997.
- Veach, E. Robust Monte Carlo methods for light transport simulation. PhD dissertation, Stanford University, Stanford, CA, USA (1998). Advisor(s) Leonidas J. Guibas. AAI9837162.
- Walter, B., Zhao, S., Holzschuch, N., Bala, K. Single scattering in refractive media with triangle mesh boundaries. *ACM Trans. Graph.* 28, 3 (Jul. 2009), 92:1–92:8.

Wenzel Jakob (wenzel@inf.ethz.ch),
Interactive Geometry Lab, ETH Zürich,
Zürich, Switzerland.

Steve Marschner (srm@cs.cornell.edu),
Department of Computer Science, Cornell
University, Ithaca, NY.

Copyright held by authors.

Technical Perspective

Treating Networks Like Programs

By George Varghese

WHY ARE MY packets being dropped? Why does my Internet connection keep resetting? Programmers developing software have access to a panoply of tools including compilers, debuggers, static and dynamic checkers. Unfortunately, network operators have access to only crude tools such as TraceRoute and Ping. These tools have remained unchanged for decades while networks have grown increasingly complicated: load balancers spread traffic using unknown hash functions, and firewalls interdict messages based on cryptic rules that human operators enter, sometimes incorrectly. In a world of services accessed from tablets and mobile devices, network failures are a debilitating source of operational expenditures.

On the surface, networks with their collections of routers, switches, network interface cards, and cabling do not resemble code. But one can view the entire network as a “program” that takes packets from the input edges of the network and outputs packets to the output edge after possibly rewriting the packet. The forwarding table at routers can, for instance, be viewed as a large Switch statement across destination addresses.

This lens has led several researchers to frame a new agenda for networks, sometimes called network verification. They ask: what are the equivalents of compilers, debuggers, and static checkers for networks? Early work has already produced static reachability checkers of impressive speed and scale, algorithms to synthesize network firewalls from policy specifications, automatic network testers, and debuggers.

Some of the research has gone beyond opportunistic application of ideas from programming languages to a new merging of two streams of thought. For example, network verification often seeks all headers that reach a destination (more analogous to allSAT than SAT), extends notions in

software testing of covering every line or function to covering every link and router queue, and finesses the problem of global time in debugging using notions of causality.

The following paper is exciting because it takes existing static checking of networks to a new level by checking the real code in the forwarding path of a Click router using classical software verification tools, and not just a model of router software as was used in previous work. It also proves absolute properties and can discover all possible bugs.

The reader may wonder why software routers are interesting when the forwarding path of the vast majority of routers used today is implemented in inscrutable hardware chips. One answer is there has been a resurgence in software networking on virtual machines in public clouds such as Azure and EC2. Second, the simplifications exploited in the paper are based on code structured as pipelines, which is the way almost all real routers are built. Thus, in principle, their ideas could be applied to hardware routers as well.

The following paper is exciting because it takes existing static checking of networks to a new level by checking the real code in the forwarding path of a Click router using classical software verification tools.

Third, there is a new movement toward designing flexible routers using a pipeline-based language called P4 that has similarities to Click. Could the techniques in this paper be useful to verify P4 programs?

Verification experts may cry foul because control flow is restricted and the data structures used are merely arrays with no dynamic pointer allocation. Have the authors merely defined the complexities of verification away? The authors rightly argue their contribution is in pointing out that small limitations in the way router forwarding software (and possibly hardware) is written can make it possible to efficiently verify router behavior, and they argue the cost of such restrictions—such as using slightly less efficient lookup tables—is small. The techniques in this paper can be combined with other techniques to verify entire networks at a more realistic level of abstraction than was considered previously.

Others may argue the results do not produce the copious amount of bugs found in prior work. However, I find it heartening the bugs they found in Click were in the interaction of obscure cases in seemingly orthogonal features such as fragmentation and IP options. Such bugs often lurk in the rarely used pathways of the execution, are difficult to flush out using conventional testing techniques, and can sometimes surface later in expensive live-site incidents.

Impressive strides have been made in applying programming languages techniques to find bugs in large complex code including bug finding for Linux and formal verification for seL4. While this work can be regarded as an experimental rocket launch, it gives us hope that someday we will reach the moon. 

George Varghese (varghese@microsoft.com) is a principal researcher at Microsoft Research, Mountain View, CA.

Copyright held by author.

Software Dataplane Verification

By Mihai Dobrescu and Katerina Argyraki

Abstract

The industry is in the mood for programmable networks, where an operator can dynamically deploy network functions on network devices, akin to how one deploys virtual machines on physical machines in a cloud environment. Such flexibility brings along the threat of unpredictable behavior and performance. What are the minimum restrictions that we need to impose on network functionality such that we are able to verify that a network device behaves and performs as expected, for example, does not crash or enter an infinite loop? We present the result of working iteratively on two tasks: designing a domain-specific verification tool for packet-processing software, while trying to identify a minimal set of restrictions that packet-processing software must satisfy in order to be verification-friendly. Our main insight is that packet-processing software is a good candidate for domain-specific verification, for example, because it typically consists of distinct pieces of code that share limited mutable state; we can leverage this and other properties to sidestep fundamental verification challenges. We apply our ideas on Click packet-processing software; we perform complete and sound verification of an IP router and two simple middleboxes within tens of minutes, whereas a state-of-the-art general-purpose tool fails to complete the same task within several hours.

1. INTRODUCTION

What kind of programming interface should a network device expose to its operator? Answering this question involves a fundamental trade-off between flexibility and verifiability: the more flexible the interface we choose, the wider the range of functionality that we can deploy on a network device, but the harder it is to verify that the network device will behave and perform as expected, for example, it will not crash, no matter how we configure it or what traffic it receives.

Traditional network devices (switches and routers) do not expose any interface for frequent reprogramming. They implement a well-defined set of rarely changing algorithms, like Ethernet switching or IP forwarding. Part of this functionality is typically hardcoded in application-specific integrated circuits (ASICs) and cannot be reprogrammed. The rest is implemented in software running on programmable processors and can be reprogrammed by installing a new image on the device, but this is a potentially disruptive task that is not meant to be part of regular operation.

At the other end of the spectrum, certain Active Network devices—introduced by Tennenhouse and Wetherall in the late 1990s¹¹—can execute code specified in the packets they receive. In that universe, the network does not implement a predefined set of algorithms; it is a distributed execution platform, closer in spirit to today's clouds than to a traditional network.

However, such flexibility brings along the threat of unpredictable behavior and performance. Any system of non-trivial size that is subject to frequent updates typically suffers from behavior and performance bugs. Consider a piece of packet-processing code that enters an infinite loop upon receiving a particular type of packet; deploying this code on a network device would enable an attacker to disrupt network operation by causing network devices to stop processing new packets.

Until recently, network operators did not see an immediate economic benefit from subjecting the network to this threat; but now the landscape is changing. Today the network implements an evolving set of algorithms that go beyond switching and forwarding, for example, load-balancing, intrusion prevention and detection, or redundancy elimination. And since traditional network devices do not expose any interface for frequent reprogramming, when a network operator wants to extend the functionality of her network, say, with a new intrusion-prevention algorithm, she typically has to deploy a new specialized middlebox. It was recently reported that specialized middleboxes can make a significant fraction of network infrastructure.²⁵ It was also reported that one can cut network operation costs by half by getting rid of all these middleboxes in favor of flexible, programmable network devices.²⁴

Enter the modern, industry-driven reincarnation of Active Networking: Network Function Virtualization.¹⁰ The idea is to enable a network operator to dynamically deploy pieces of functionality (“network functions”) on network devices, akin to how one deploys virtual machines on physical machines in a cloud environment.

Given the industry's mood for network programming, this is a great time to revisit the question of what the network-programming interface should look like. Should it allow to deploy arbitrary network functions on network devices? Or should it restrict network functionality to enable verification? Is the flexibility/verifiability trade-off the same as it was in the days of Active Networking?

One thing that has changed since the late 1990s is software verification tools. Fully automatic, complete and sound verification of arbitrary code is still out of reach, but there is significant progress in verifying software from specific domains, for example, operating-system kernels,¹⁷ device drivers,^{1, 8} aviation control systems.³ Most importantly, there are verification tools like Klee⁴ and S2E⁷ that are designed to be used by researchers or developers who are not verification experts. These tools provide basic

A previous version of this paper was published in *Proceedings of the 11th USENIX Conference on Networked Systems and Design and Implementation*, 2014, pp. 101–114.

verification functions that can be used as building blocks for domain-specific verification.

Given these new verification tools, what are the minimum restrictions that we need to impose on network functionality such that we are able to verify that a network device satisfies useful properties? There are different ways to approach this question: one could start from a restricted, verification-friendly programming model and broaden it as much as possible; or, one could start from a popular, but not verification-friendly programming model and restrict it as little as necessary. We chose the latter in an effort to be practical.

In this paper, we present the result of working iteratively on two tasks: designing a domain-specific verification tool for packet-processing software, while trying to identify a minimal set of restrictions that packet-processing software must satisfy in order to be verification-friendly. As a first step, we apply our ideas on Click¹⁸ packet-processing software; we verify that an IP router, a Network Address Translation (NAT) box, and a simple traffic monitor do not crash and do not execute more than a given number of instructions per packet, independently from their configuration and the traffic they receive. Our domain-specific tool completes its proofs within tens of minutes, whereas a state-of-the-art general-purpose tool could not complete the same tasks within hours.

Our main insight is that packet-processing software is a good candidate for domain-specific verification: First, it is typically written in the form of a pipeline of distinct packet-processing elements (e.g., a firewall element, an encryption element, an IP lookup element) that do not share mutable state between them, other than the packet passed from one element to the next and associated metadata. Second, it does not typically involve recursion or loops with an unpredictable number of iterations. Intuitively, these characteristics make it easier to reason about the behavior and performance of packet-processing software than that of arbitrary software. Our verification tool has semantic information about certain common aspects of packet-processing software (e.g., a “packet” is not an arbitrary data structure, a “packet-processing loop” is not an arbitrary control structure), and this information enables it to sidestep fundamental verification challenges.

In the rest of the paper, we provide basic background (Section 2), describe the design of our verification tool (Section 3), summarize our evaluation results (Section 4), and close with a discussion of open issues (Section 5).

2. SETUP

In this section, we provide background on network devices (Section 2.1), specify the particular network devices that our work addresses (Section 2.2), provide background on symbolic execution, the main verification technique that we build on (Section 2.3), outline the main insight that motivated our work (Section 2.4), and summarize prior work (Section 2.5).

2.1. Network devices and click

A traditional network device consists of two components: the

“data plane,” which implements packet-processing algorithms (e.g., IP forwarding, filtering, encryption, and encapsulation) and the “control plane,” which implements the algorithms that populate the data plane’s data structures (e.g., the routing algorithms that populate the IP forwarding table). The data plane can be implemented in ASICs or in software (or a combination of the two), while the control plane is typically implemented in software.

Our work concerns software data planes that are subject to certain requirements: (a) Flexibility, for example, it should be easy to extend packet processing with a new filtering algorithm, or to replace the existing IP lookup algorithm with a new, faster one. (b) Predictability, for example, a packet should not be processed for more than a few microseconds; given that a software data plane is typically CPU-bound when processing small packets or performing CPU-intensive tasks (e.g., encryption or intrusion detection), a latency constraint translates into a constraint in the number (and nature) of instructions executed per packet.

To combine flexibility with predictability, it makes sense to follow certain principles when designing and implementing a software data plane: (a) Avoid inefficient programming patterns or patterns that may result in unpredictable performance, for example, recursion or loops with an unpredictable number of iterations. (b) Structure the data plane as a pipeline of distinct packet-processing elements (e.g., a filtering element, an encryption element, an IP forwarding element) that do not share mutable state, other than passing the packet itself from one element to the next, together with associated metadata.

These principles are the result of discussing with developers and our own experience, but we do not have quantitative evidence that most software data planes follow them. Avoiding inefficient or unpredictable patterns is common sense—one can hardly envision a network device that executes unbounded loops upon receiving a packet. Structuring the data plane as a pipeline of distinct, loosely coupled elements simplifies code development and maintenance. To the best of our knowledge, the first packet-processing framework that explicitly advocated this principle is Click: “A Click router is assembled from packet processing modules called *elements*. Individual elements implement simple router functions like packet classification, queuing, ... A router configuration is a directed graph with elements at the vertices; packets flow along the edges of the graphs.”¹⁸ Click is arguably the most popular packet-processing framework within the academic community. However, commercial middleboxes are typically not open-source, and we acknowledge that they may not follow a Click-like structure. One of the points we make with this work is that they should, because this structure is verification-friendly without compromising performance.

2.2. Pipeline structure

We focus on software data planes that consist of distinct packet-processing elements, where each element may access three types of state (Table 1):

Packet state is owned by exactly one element at any point in time. It can be read or written only by its owner; the current

Table 1. Types of packet-processing state

	Written by	Read by	Transferable ownership
Packet state	Owner	Owner	Yes
Private state	Owner	Owner	No
Static state	—	Any	—

owner (and nobody else) may atomically transfer ownership to another element. Packet state is used for communicating packet content and metadata between elements. For each newly arrived packet, there is typically an element that reads it from the network, creates a packet object, and transfers object ownership to the next element in the pipeline. Once an element has transferred ownership of a packet, it cannot read or write it any more.

Private state is owned by one element and never changes ownership. It can be read or written only by its owner, and it persists across the processing of multiple packets. A typical example is a map in a NAT element, or a flow table in a traffic-monitoring element.

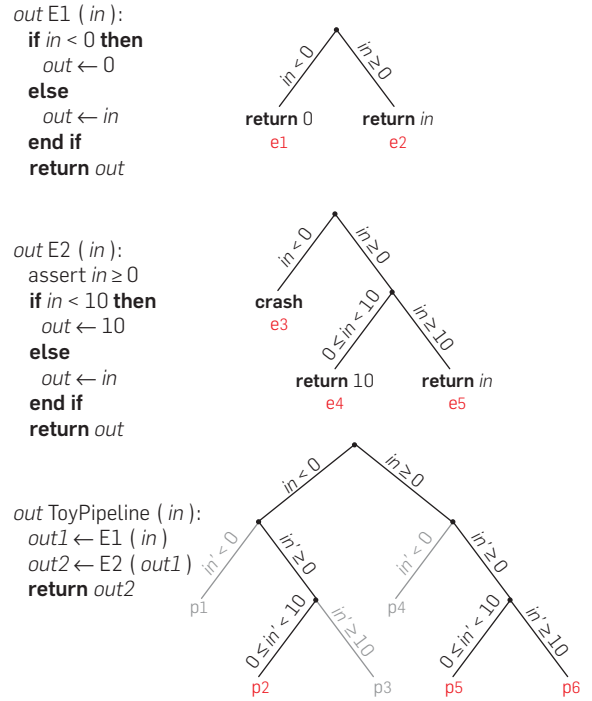
Static state can be read by any element but not written by any element. This state is immutable as far as the data plane is concerned. A typical example is an IP forwarding table.

2.3. Symbolic execution

A program can be viewed as an “execution tree,” where each node corresponds to a program state, and each edge is associated with a basic block. Running the program for a given input leads to the execution of a sequence of instructions that corresponds to a path through the execution tree, from the root to a leaf. For example, the program E_1 in Figure 1 may execute two instruction sequences: one for input $in < 0$ and the other for input $in \geq 0$; hence, its execution tree (shown to the right of the program) consists of two edges, one for each “input class” and instruction sequence.

Symbolic execution^{4, 14} is a practical way of generating execution trees. During normal execution of a program, each variable is assigned a concrete value, and only a single path of the tree is executed. During symbolic execution, a variable may be symbolic, that is, assigned a set of values that is specified by an associated constraint. For example, a symbolic integer x with associated constraint $x > 2 \wedge x < 5$ is the set of concrete values $x = \{3, 4\}$. A symbolic-execution engine can take a program, make the program’s input symbolic, and execute all the paths that are feasible given this input.

Consider the program E_2 in Figure 1 and assume that the input in can take any integer value. To symbolically execute this program, we start at the root of the tree and execute all the feasible paths. As we go down each path, we collect two pieces of information: the “path constraint” specifies which values of in lead to this path, and the “symbolic state” maps each variable to its current value on this path. For example, at the end of path e_4 , the path constraint is $C = (in \geq 0 \wedge in < 10)$, and the symbolic state is $S = \{out = 10\}$; at the end of path e_3 , the path constraint is $C = (in \geq 10)$, and the symbolic state is $S = \{out = in\}$.

Figure 1. A toy pipeline that consists of two elements.


In principle, symbolic execution provides an automatic, conceptually straightforward way to verify a program, that is, (dis)prove that the program satisfies a target property. For example, suppose we want to prove that program E_1 in Figure 1 never crashes, no matter what input it receives; if we symbolically execute E_1 specifying no constraint for in , we will execute all two paths of the program without crashing, which constitutes proof that the program does not crash for any input. Similarly, if we symbolically execute E_2 in Figure 1 specifying no constraint for in , we will execute all three paths of the program, and path e_3 will end in a crash, which constitutes proof that the program can crash when $in < 0$.

In practice, symbolic execution is rarely used to generate proofs, because of path explosion: The number of paths in a program generally grows exponentially with the number of branching points. As a result, a real program (even one that consists of a few hundred lines of code) can have so many paths that it is impossible to execute all of them in useful time.

On the other hand, symbolic execution is successfully used to test programs. The difference between testing a program and proving something about it lies in “path coverage,” which is the fraction of the program’s paths that are executed. Proving something about a program through symbolic execution means executing 100% of its paths, which is not a requirement for testing. A popular metric for evaluating a testing tool is “line coverage,” which is the fraction of the program’s lines of code that are exercised by the tool. It is worth noting that line coverage and path coverage are very different metrics. For example, when Klee⁴ tests UNIX coreutils like *nice* or *cat*, it achieves line coverage above 70%, but path coverage below 1%.²⁰

There are two main reasons why we chose to build on symbolic execution, despite its limitations: First, there exist publicly available, well-maintained symbolic execution tools^{4, 7} that do not require verification expertise. This is not accidental: symbolic execution can be thought of as a “brute force” approach to verification; it does not require any sophisticated reasoning about program logic, which makes it easy to use (and also hard to scale). Second, symbolic execution does not require access to the source code of the program, only the executable binary. This is consistent with the vision of a network operator verifying packet-processing software written by third parties, which may not have an incentive to provide the source code.

2.4. Starting point

We observe that symbolic execution is a good fit for software data planes, because their pipeline structure can help sidestep path explosion. In a pipeline, two elements (stages) never concurrently hold read or write permissions to the same mutable state, regardless of whether that state is a packet being processed or some other data structure. This level of isolation can help significantly with path explosion.

Our approach is to first analyze each pipeline element in isolation, then compose the results to prove properties about the entire pipeline. This can reduce the amount of work that needs to be done to prove something about the pipeline by as much as an exponential factor: If each element has n branches and roughly 2^n paths, a pipeline of m such elements has roughly $2^{m \cdot n}$ paths. Analyzing each element in isolation (as opposed to the entire pipeline in one piece) can reduce the number of paths that need to be explored roughly from $2^{m \cdot n}$ to $m \cdot 2^n$. In the worst case, the per-element analyses yield that every single pipeline path warrants further analysis (so we end up having to consider all the paths anyway). In practice, we expect that most pipeline paths are irrelevant to the target property, and we only need to consider a small fraction.

2.5. Prior work

Our work is feasible because of advances in program analysis tools for C/C++ code, from Verisoft¹² to modern model checkers^{19, 23} and tools based on symbolic execution.^{4, 5, 7, 14} Of particular relevance to our work is the idea of using compositionality to address path explosion.¹³ Verification techniques have been used before to debug or verify networked systems (but not data planes): Musuvathi and Engler²² adapted the CMC model checker to test the Linux TCP implementation for interoperability with the TCP specification. Bishop et al.² contributed a formal specification of TCP/IP and the sockets API, and they tested existing implementations for conformance to their specification. Killian et al.¹⁶ contributed new algorithms for finding liveness bugs in systems like Pastry and Chord. NICE finds bugs in Open-Flow applications.⁶ SOFT tests OpenFlow switches for inter-operability with reference implementations.²¹ Guha et al.¹⁵ contributed “the first machine-verified [Software Defined Networking] controller.”

3. SYSTEM

In this section, we describe our system: first how it leverages

the pipeline structure to sidestep inter-element path explosion (Section 3.1); second, how it leverages other aspects of packet processing to sidestep intra-element path explosion resulting from loops (Section 3.2), large data structures (Section 3.3), and mutable state (Section 3.4). We close with a quick comment on the kinds of overhead that our design introduces (Section 3.5).

As we describe each technique used by our system, we also state any extra conditions on top of pipeline structure (Section 2.2) that this technique requires from the target software in order to work well. If a software data plane does not meet these conditions, our tool may not be able to complete a proof for this data plane.

We will illustrate our system through Figure 1, which shows a pipeline consisting of two elements. We will use the term *segment* to refer to an instruction sequence through a single element, and the term *path* to refer to an instruction sequence through the entire pipeline. The input *in* corresponds to a newly received packet, and we assume that this may contain anything, that is, we make *in* symbolic and unconstrained.

We developed our system on top of S2E, which is a state-of-the-art, publicly available verification framework for general software.⁷ For illustration purposes, our examples simplify two aspects of our system: first, our examples input *in* is an integer, whereas in reality the input packet object is an array of bytes; second, our example code snippets consist of pseudo-code, whereas in reality S2E takes as input X86 code.

3.1. Pipeline decomposition

Verification consists of two main steps: step 1 searches inside each element, in isolation, for code that may violate the target property, while step 2 determines which of these potential violations are feasible once we assemble the elements into a pipeline. More specifically, we cut each pipeline path into element-level segments (Figure 1). In step 1, we obtain, for each segment, a logical expression that specifies how this segment transforms state; this allows us to identify all the “suspect segments” that may cause the target property to be violated. In step 2, we determine which of the suspect segments are feasible and indeed cause the target property to be violated, once we assemble segments into paths.

In step 1, we analyze each element in isolation: First, we symbolically execute the element assuming unconstrained symbolic input. Next, we conservatively tag as “suspect” all the segments that may cause the target property to be violated. For example, in Figure 1, if the target property is crash-freedom, segment e_3 is tagged as suspect, because, if executed, it leads to a crash.

If we stopped at step 1, our verification would catch all property violations, but could yield false positives: If this step does not yield any suspect segments for any element, then we have proved that the pipeline satisfies the target property. For instance, if none of the elements ever crashes for any input, we have proved that the pipeline never crashes. However, a suspect segment does not necessarily mean that the pipeline violates the target property, because a segment that is feasible in the context of an individual element may become infeasible in the context of the full pipeline. For example, in Figure 1, if we consider element E_2 alone, segment e_3 leads to

a crash; however, in a pipeline where E_2 always follows E_1 , segment e_3 becomes infeasible, and the pipeline never crashes. In program-analysis terminology, in step 1, we over-approximate, that is, we execute some segments that would never be executed within the pipeline that we are aiming to verify.

Step 2 discards suspect segments that are infeasible in the context of the pipeline: First, we construct each potential path p_i that includes at least one suspect segment; each p_i is a sequence of segments e_j . Next, we compose the path constraint and symbolic state for p_i based on the constraints and symbolic state of its constituent segments (that we have already obtained in step 1). Finally, for every p_i , we determine whether it is feasible (based on its constraints) and whether it violates the target property (based on its symbolic state). Note that the last step does not require actually executing p_i , only composing the logical expressions of its constituent segments.

For example, here is how we prove that the pipeline in Figure 1 does not crash:

Step 1:

1. We symbolically execute E_1 assuming input in can take any integer value. We collect the following constraints and symbolic state for its segments e_1 and e_2 .
 - $C_1(in) = (in < 0)$, $S_1(in) = \{out = 0\}$.
 - $C_2(in) = (in \geq 0)$, $S_2(in) = \{out = in\}$.
2. We symbolically execute E_2 assuming input in can take any integer value. We collect the following constraints and symbolic state for its segments e_3 , e_4 , and e_5 :
 - $C_3(in) = (in < 0)$, $S_3(in) = \{crash\}$.
 - $C_4(in) = (in \geq 0 \wedge in < 10)$, $S_4(in) = \{out = 10\}$.
 - $C_5(in) = (in \geq 10)$, $S_5(in) = \{out = in\}$.
3. We tag segment e_3 as suspect.

Step 2:

1. The paths that include the suspect segment are p_1 (i.e., sequence $\langle e_1, e_3 \rangle$) and p_4 (i.e., sequence $\langle e_2, e_3 \rangle$).
2. We compute p_1 's path constraint as $C_1'(in) = C_1(in) \wedge C_3(S_1(in)[out]) = C_1(in) \wedge C_3(0) = (in < 0) \wedge (0 < 0) = False$.
3. We compute p_4 's path constraint as $C_4'(in) = C_2(in) \wedge C_3(S_2(in)[out]) = C_2(in) \wedge C_3(in) = (in \geq 0) \wedge (in < 0) = False$.
4. Both path p_1 's and path p_4 's constraints always evaluate to false, hence p_1 and p_4 are infeasible, that is, there are no feasible paths that include suspect segments, hence the platform never crashes.

Pipeline decomposition enables us to prove properties about the pipeline without having to consider every single pipeline path; but it still requires us to consider every single element segment. This is not straightforward for elements that involve loops or large data structures. We will next discuss how we address each of these scenarios.

3.2. Loops

Loops can create an impractical number of segments within an element. Consider an element that implements the processing of IP options: for each received packet, it loops over the options stored in the packet's IP header and performs the

processing required by each specified option type. If the processing of one option yields up to 2^n segments, then the processing of t options yields up to $2^{t \cdot n}$ segments. For example, in the IP-options element that comes with the Click distribution, the processing of 3 options yields millions of segments that—we estimated—would take months to symbolically execute.

To address this, we reuse the idea of decomposition, this time applying it not to the entire pipeline, but to each loop: If a loop has t iterations, we view it as a “mini-pipeline” that consists of t “mini-elements,” each one corresponding to one iteration of the loop. We have described how, if we have a pipeline of m elements, we symbolically execute each element in isolation, then compose the results to reason about the entire pipeline. Similarly, if we have a loop of t mini-elements (iterations), we symbolically execute each mini-element in isolation, then compose the results to reason about the entire loop. Unlike a pipeline that consists of different element types, a loop of t iterations consists of the same mini-element type, repeated t times; hence, for each loop, we only need to symbolically execute one mini-element.

This brings us to our first extra condition on software data planes: To use pipeline decomposition as we do, the only mutable state shared across components must be the packet object itself. For instance, to decompose a pipeline into individual elements, we rely on the fact that the only mutable state shared across elements is the packet object (Section 3.1). Similarly, to decompose a loop into individual iterations, the only mutable state shared across iterations must be part of the packet object.

For example, consider again an IP-options element: Such an element typically includes a `next` variable, which points to the IP-header location that stores the next option to be processed; each iteration of the main loop starts by reading this variable and ends by incrementing it. In a conventional element, `next` would be a local variable. In our verification-optimized element, `next` is part of the packet metadata, hence part of `packet`. And since, in step 1, we make `packet` symbolic and unconstrained, `next` is also symbolic and unconstrained, allowing us to reason about the behavior of one iteration of the main loop, assuming that iteration may start reading from *anywhere* in the IP header.

CONDITION 1. Any mutable state shared across loop iterations is part of the packet metadata.

3.3. Data structures

Symbolic-execution engines lack the semantics to reason about data structures in a scalable manner. For instance, symbolically executing an element that uses a packet's destination IP address to index an array with a thousand entries will cause a symbolic-execution engine to essentially branch into a thousand different segments—independently from the array content or the logic of the code that uses the returned value. So, if we naïvely feed an element with a forwarding or filtering table of more than a few hundred entries to a symbolic-execution engine, step 1 of our verification process will not complete in useful time.

To address this, when we reason about an element, we abstract away any data-structure access; this allows us to

symbolically execute the element and identify suspect segments, without requiring the symbolic-execution engine to handle any data structures. To reason about the data structures themselves, we rely on other means, for example, manual or static analysis; this restricts us to using only data structures that are manually or statically verifiable, but we have evidence that these are typically sufficient for packet-processing functionality.

This brings us to our second extra condition on software data planes: To reason about different components of the same executable separately, there must exist a well-defined interface between them. For instance, to reason about each pipeline element separately and compose the results, we rely on the existence of a well-defined interface between each pair of elements, which specifies all the state that can be exchanged between them—the packet object (Section 3.1). Similarly, to reason about a data structure separately from the element that uses it and compose the results, the data structure must expose a well-defined interface to the element.

We need an interface that abstracts each data structure as a key/value store that supports at least read, write, membership test, and expiration. The first three operations are straightforward; the last one—expiration—allows an element to indicate that a {key, value} pair will not be accessed by the element any more, hence is ready to be removed and processed by the higher layers. For example, suppose an element maintains a data structure with per-flow packet counters; when a flow completes (e.g., because a FIN packet from that flow is observed), the element can use the expiration operation to signal this completion to the control-plane process that manages traffic statistics.

CONDITION 2. *Elements use data structures that expose a common key/value-store interface.*

Moreover, we need data structures that expose the above interface *and* can be verified in useful time. When we say that a data structure is “verified,” we mean that the implementation of the interface exposed by the data structure is proved to satisfy crash-freedom, bounded-execution, and correctness. The latter depends on the particular semantics of the data structure, for example, a hash-table should satisfy the following property: a “write (key, value)” followed by a “read (key)” should return “value.” If an element uses only data structures for which these properties hold, then, when we reason about the element, we can abstract away all the data-structure implementations and consider only the rest of the element code plus the data-structure interfaces.

CONDITION 3. *Elements use data structures that are implemented on top of verifiable building blocks, for example, pre-allocated arrays.*

As evidence that such data structures exist, we implemented a hash table and a longest-prefix-match table that satisfy crash-freedom and bounded-execution. They both consist of chains of pre-allocated arrays. For instance, our hash table is a sequence of K such arrays; when adding the

k -th key/value pair that hashes to the same index, if $k \leq K$, the new pair is stored in the k -th array, otherwise it cannot be added (the write operation returns False).

We chose arrays as the main building block, because they combine two desirable properties: (a) They enable linear access to packet-processing state, because of their $O(1)$ lookup time. (b) They are easy to verify, because of the simplicity of their semantics. For example, a write to an array (that is within the array bounds) is guaranteed not to cause a crash and not to cause the execution of more than a known number of instructions that depends on the particular CPU architecture. In contrast, a write to a dynamically-growing data structure, for example, a linked list or a radix trie, may result in a variable number of memory allocations, deallocations and accesses, which can fail in unpredictable ways.

3.4. Mutable private state

Mutable private state is hard to reason about because it may depend on a sequence of observed packets (as opposed to the currently observed packet alone). For instance, if an element maintains connection state or traffic statistics, then its private state is a function of all traffic observed since the element was initialized. Hence, it is not enough to reason about the segments of the element that can result from all possible contents of the current packet; we need to reason about the segments that can result from all possible contents of all possible packet sequences that can be observed by the element. The challenge is that symbolic-execution engines (and verification tools in general) are not yet at the point where they can handle symbolic inputs of arbitrary length in a scalable manner.

We have only scratched the surface of verifying elements with mutable state. We can currently verify two specific kinds of such elements: a NAT that maintains per-connection state and rewrites packet headers accordingly, and a traffic monitor that maintains per-flow packet counters. We believe that our approach can be generalized to other elements, but we do not expect to be able to perform complete and sound verification of an element that performs arbitrary state manipulation—claiming that would be close to claiming that we could verify arbitrary software.

Our approach is to break verification step 1 (Section 3.1) into two substeps: the first one searches for “suspect” values of the private state that would cause the target property to be violated, while the second one determines which of these potential violations are feasible given the logic of the element. In the first sub-step, we assume that the private state can take *any* value allowed by its type (i.e., we over-approximate). In the second sub-step, we take into account the fact that private state cannot, in reality, take any value, but is restricted by the particular type of state manipulation performed by the given element. So far, we have not exercised the second sub-step in practice: in the two stateful elements that we have experimented with, the first sub-step did not reveal any suspect states, hence the second one was not exercised.

3.5. Overhead

Our extra conditions introduce two kinds of overhead: First, existing elements may need to be rewritten to satisfy them; this involves rewriting existing loops, replacing existing data

structures, and changing any line of code that accesses a data structure. To reduce this overhead, part of our work is to create a library of data structures that satisfy our conditions. Second, implementing sophisticated data structures on top of pre-allocated arrays typically requires more memory than conventional implementations. We put specific numbers on these kinds of overhead in the next section, where we summarize our evaluation results.

4. EVALUATION HIGHLIGHTS

We tested our ideas on pipelines created with Click. In each tested pipeline, packets are generated by a “generator” element and dropped by a “sink” element; what we verify is the packet-processing code between generator and sink. We answer the following questions: Can we perform complete and sound verification (Section 4.1)? How does verification time increase with pipeline length (Section 4.2)? Can we use our tool to uncover bugs in existing code and useful performance characteristics (Section 4.3)?

4.1. Feasibility and overhead

We considered pipelines with three types of elements: (1) Elements from Click release 2.0.1 that we did not change at all: Classifier, CheckIPHeader, EtherEncap, DecIPTTL, DropBroadcasts, and Strip (Ethernet decapsulation). (2) Elements from the same release that we modified modestly: IPGWOptions and DirectIPLookup. (3) Elements that we wrote from scratch: a NAT and a traffic monitor that maintains per-flow packet counters.

For each pipeline, we proved crash-freedom (the pipeline will not crash) and bounded-execution (the pipeline will not execute more than a given number of instructions per packet). More generally, for each pipeline, we were able to answer questions of the following kind: can line X in element Y be executed with arguments Z in the context of this pipeline? If yes, what is a packet that would cause this line to be executed with these arguments?

We have not yet applied our approach widely enough to have statistically meaningful results on the amount of rewriting effort. To make the IPGWOptions and DirectIPLookup elements satisfy our conditions, we had to change, respectively, about 26 lines (12% of the element’s code) and 130 lines (20% of the element’s code), which took a few hours. We also tried to make the Click IPRewriter element from release 2.0.1 (which generalizes NAT functionality) satisfy our conditions, but we ended up changing most of the element, because most of its code accesses data structures; in the end, we wrote it from scratch (about 870 lines of code), which took a couple of days. Our traffic monitor is about 650 lines of code.

In terms of memory overhead: The Click IPRewriter element stores per-connection state in a hash table, implemented as an array of dynamically growing linked lists (when adding the k -th key/value pair that hashes to the same index, the new pair is stored as the k -th item of a linked list). In contrast, our NAT element uses a hash table implemented as a chain of $K=3$ pre-allocated arrays (this value makes the probability of dropping a connection negligible). Hence, our NAT element may use up to three times more memory to store the same amount

of state. In our opinion, sacrificing memory for verifiability is worth considering given the relative costs of memory and the human support for dealing with network problems.

4.2. Scalability

We now examine how verification time increases with pipeline length. To show the benefit of our domain-specific techniques, we use as a baseline vanilla S2E. We refer to our approach as “dataplane-specific verification” and to the baseline as “generic verification.” We feed the same code to the two systems.

We consider three meaningful pipelines: (a) *edge router* implements a standard IP router with a small forwarding table (10 entries); (b) *core router* is similar but has a large forwarding table (100,000 entries); (c) *network gateway* implements NAT and per-flow statistics collection. Each of them presents an extra verification challenge: the first one includes a loop (processing IP options), the second one a large data structure (the IP lookup table), and the third one mutable private state (in our NAT and traffic monitor elements).

Figure 2a shows how verification time increases as we add more elements to the IP-router pipelines: Dataplane-specific verification completes in less than 20 min. Most of this time is spent on IP options, because this element has significantly more branching points than the rest. Generic verification of the edge router exceeds 12 h (at which point we abort it) the moment we allow packets to carry 2 IP options (so we do not show any data point for it beyond “+IPoption2”). Generic verification of the core router exceeds 12 h the moment we add the IP lookup element to the pipeline (so we do not show any data point for it beyond “+IPlookup”). The difference between the two tools comes from our pipeline decomposition and special treatment of loops and large data structures.

Figure 2b shows the same information for the network-gateway pipeline: Dataplane-specific verification completes in less than 6 min, whereas generic verification exceeds 12 h the moment we add either the traffic monitor or the NAT element. The difference comes from our pipeline decomposition and the fact that we abstract away data-structure implementations.

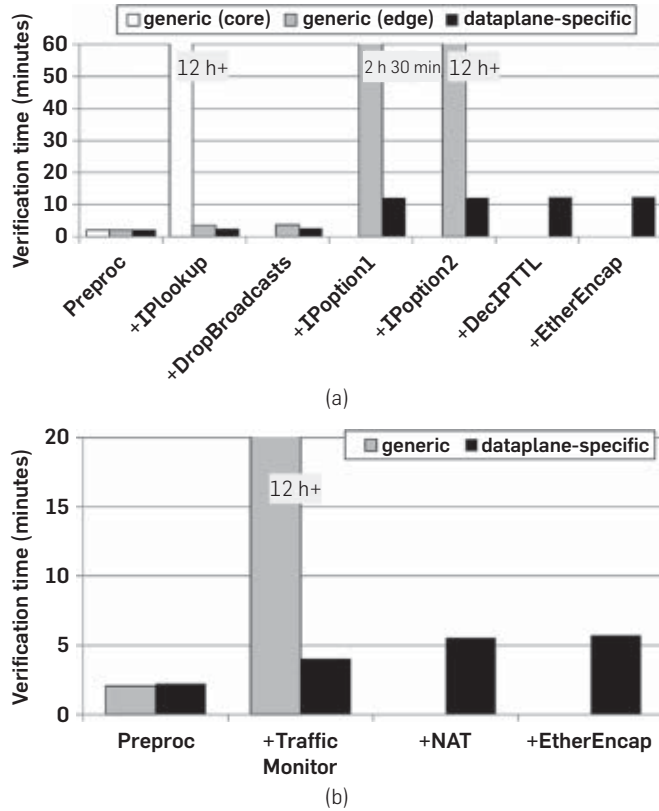
4.3. Usefulness

We said that our tool can help developers debug their code, and network operators to better understand the performance and behavior of their data planes; we now look at a few specific examples.

Bugs in existing code. While trying to prove crash-freedom and bounded-execution for various pipelines, we encountered the following instances where a malicious end-host could disable the pipeline by sending a specially crafted packet. All elements that we refer to belong to Click release 2.0.1 (without any subsequent patches).

Any pipeline that includes the IPFragmenter element, not preceded by the IPGWOptions element, will enter an infinite loop if it tries to fragment a packet that carries a zero-length IP option. This is because the current option length determines where the next iteration of the loop will start reading, hence, a zero-length option causes the loop to

Figure 2. Verification time as a function of pipeline length. “Preproc” consists of the Click Classifier, CheckIPHeader, and Strip elements. (a) IP router. For the dataplane-specific approach, the results are the same for the edge and core pipelines. (b) Network gateway.



get stuck.^a The IPGWOPTIONS element discards any packet with a zero-length option, so including it in the pipeline prevents the bug from being exercised.

Any pipeline that includes the IPRewriter element^b will hit a failed assertion^c if it receives a packet with source IP address/port tuple $T_s = T$ and destination tuple $T_d = T$, where T is the public IP address/port of the IPRewriter. We should note that this particular bug was fixed (independently from us) in a patch subsequent to release 2.0.1.

Longest paths in IP router. We used our tool to construct adversarial—from a performance point of view—workloads for a pipeline implementing a standard IP router. Recent research showed that such a router is capable of multi-Gbps lines rates,⁹ but this result was obtained using work loads of well-formed packets, not meant to exercise the pipeline’s exception paths. Instead, we obtained the pipeline’s 10 (it could have been any number) longest paths, as well as the packets that cause them to be executed.

It is not surprising that the longest paths are executed in response to problematic packets that trigger further packet examination and logging; what may be surprising is that these paths execute 2.5 times as many instructions

as the most common path. Moreover, these extra instructions are CPU-heavy, that is, they include memory accesses and system calls for logging; an attacker may cause significant performance degradation by sending a sequence of packets that are specially crafted to exercise these particular paths. This is useful information to a developer, because it reveals to her paths that may require her attention. It is also useful to a network operator, because it reveals to her the performance limits of a pipeline and the workloads that trigger them—allowing her to decide whether it is suitable for her network.

5. GOING FORWARD

The key enabler and at the same time limitation of our work is that we focus on software data planes that are subject to certain restrictions: (1) They obey a pipeline structure. We think that this is a natural fit for packet-processing software. Most research prototypes are already written this way, and we know of at least one industry prototype as well. Favoring an already popular programming model is, in our opinion, a modest price to pay for verifiability. (2) They satisfy our extra conditions, which may require rewriting existing code (but the resulting code is, in our opinion, easier to read and maintain); and may result in implementations with larger memory footprints than existing code (but trading off memory for verifiability is, in our opinion, worth considering).

Our approach is applicable to packet-processing platforms where each packet is handled by a single processing core and different cores never need to synchronize. We focused on such platforms, because there is compelling evidence that they lead to better performance (by minimizing the number of compulsory cache misses).⁹ We would need new results in order to verify platforms where different cores contend for access to the same data structures.

As we continue our work, we are coming to the following conclusion: symbolic execution is a powerful tool, but it alone cannot take us to complete and sound verification of sophisticated, stateful packet-processing elements. We were able to use it thus far, because we considered only stateless elements (IP router) or elements that maintain relatively simple forms of state (NAT and traffic monitor). To verify stateful elements, we first assume that the element’s state may take any value and check whether the target property holds; if it does not hold for a particular suspect value, we then try to reason about whether the element’s state can ever take that suspect value (Section 3.4). The challenge lies in the second step. In the case of our NAT and traffic monitor, there were no suspect values to begin with—and even if there were, reasoning about the feasible values of the state of these elements is relatively simple. However, an intrusion detection system (IDS) may implement sophisticated state machines that we cannot reason about, even with our domain-specific optimizations.

Hence, we are heading toward an approach where we analyze each element in isolation and compose the results on demand (as our current system does), but we use a wider range of techniques for our per-element analysis. Techniques like model checking or abstract interpretation may be necessary if we want to reason about sophisticated,

^a elements/ip/ipfragmenter.cc, line 69.

^b elements/tcpudp/iprewriter.cc.

^c include/click/heap.hh, line 149.

stateful packet-processing elements. The challenge lies in simplifying these techniques enough to make them accessible to non-verification experts; continuing in the same vein with our current work, we will try to do this by leveraging the idiosyncrasies of packet-processing code. For example, we are exploring the idea of identifying patterns of code that are common in packet processing, and developing specialized techniques for verifying these particular patterns.

There will always be software that we cannot verify, and we by no means advocate that network operators reject it. Rather, we advocate classifying software into “verified” and “non-verified,” to enable network operators to make informed decisions about deploying it on their network devices. For instance, it makes sense to deploy non-verified software in an isolated fashion (e.g., within a container, where only experimental traffic can reach it), to contain its effect on the rest of the network. Perhaps this combination of verification and dual-mode deployment is the way to reconcile the need for predictable network behavior and performance with the vision of a continuously evolving Active Network.

Acknowledgments

Thanks to Stefan Bucur, George Candea, Vitaly Chipounov, Johannes Kinder, Vova Kuznetsov, Christian Maciocco, Dimitris Melissos, David Ott, Iris Safaka, Simon Schubert, and Cristian Zamfir. This work was supported by an Intel gift and a Swiss National Science Foundation grant. 

References

- Ball, T., Rajamani, S.K. The SLAM project: Debugging system software via static analysis. In *Proceedings of the ACM Symposium on the Principles of Programming Languages (POPL)* (2002).
- Bishop, S., Fairbairn, M., Norrish, M., Sewell, P., Smith, M., Wansbrough K. Rigorous specification and conformance testing techniques for network protocols, as applied to TCP, UDP, and Sockets. In *Proceedings of the ACM SIGCOMM Conference* (2005).
- Blanchet, B., Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Monniaux, D., Rival, X. A static analyzer for large safety-critical software. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)* (2003).
- Cadar, C., Dunbar, D., Engler, D.R. KLEE: Unassisted and automatic generation of high-coverage tests for complex systems programs. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)* (2008).
- Cadar, C., Engler, D.R. EXE: Automatically generating inputs of death. In *Proceedings of the ACM Conference on Computer Communication Security (CCS)* (2006).
- Canini, M., Venzano, D., Peresini, P., Kostic, D., Rexford J. A NICE way to test OpenFlow applications. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2012).
- Chipounov, V., Kuznetsov, V., Candea, G. The S2E platform: Design, implementation, and applications. *ACM Transactions on Computer Systems* 30, 1 (February 2012), 2:1–2:49.
- Cook, B., Podelski, A., Rybalchenko, A. Termination proofs for systems code. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)* (2006).
- Dobrescu, M., Egi, N., Argyraki, K., Chun, B.-G., Fall, K., Iannaccone, G., Knies, A., Manesh, M., Ratnasamy, S. RouteBricks: Exploiting parallelism to scale software routers. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)* (2009).
- I.S.G. European Telecommunications Standards Institute (ETSI). *Network Functions Virtualisation (NFV): Use Cases*. Technical Report ETSI GS NFV 001 V1.1.1, ETSI, 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, FRANCE, 2013. Reference DGS/NFV-009.
- Feamster, N., Rexford, J., Zegura, E. The road to SDN: An intellectual history of programmable networks. *ACM Comput. Commun. Rev.* 44, 2 (April 2014), 87–98.
- Godefroid, P. Model checking for programming languages using Verisort. In *Proceedings of the ACM Symposium on the Principles of Programming Languages (POPL)* (1997).
- Godefroid, P. Compositional dynamic test generation. In *Proceedings of the ACM Symposium on the Principles of Programming Languages (POPL)* (2007).
- Godefroid, P., Klarlund, N., Sen, K. DART: Directed automated random testing. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)* (2005).
- Guha, A., Reitblatt, M., Foster, N. Machine-verified network controllers. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)* (2013).
- Killian, C., Anderson, J.W., Jhala, R., Vahdat, A. Life, death, and the critical transition: Finding liveness bugs in systems code. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2007).
- Klein, G., Elphinstone, K., Heiser, G., Andronick, J., Cock, D., Derrin, P., Elkaduwe, D., Engelhardt, K., Norrish, M., Kolanski, R., Sewell, T., Tuch, H., Winwood, S. seL4: Formal verification of an OS kernel. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)* (2009).
- Kohler, E., Morris, R., Chen, B., Jannotti, J., Kaashoek, M.F. The click modular router. *ACM Trans. Comput. Syst.* 18, 3 (August 2000), 263–297.
- Kroening, D., Clarke, E., Yorav, K. Behavioral consistency of C and Verilog programs using bounded model checking. In *Proceedings of the Design Automation Conference (DAC)* (2003).
- Kuznetsov, V., Kinder, J., Bucur, S., Candea, G. Efficient state merging in symbolic execution. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)* (2012).
- Kuzniar, M., Peresini, P., Canini, M., Venzano, D., Kostic, D. A SOFT way for OpenFlow switch interoperability testing. In *Proceedings of the ACM Conference on emerging Networking Experiments and Technologies (CoNEXT)* (2012).
- Musuvathi, M., Engler, D.R. Model checking large network protocol implementations. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2004).
- Musuvathi, M., Park, D., Chou, A., Engler, D.R., Dill, D.L. CMC: A pragmatic approach to model checking. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)* (2002).
- Sekar, V., Egi, N., Ratnasamy, S., Reiter, M.K., Shi, G. Design and implementation of a consolidated middlebox architecture. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2012).
- Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., Sekar, V. Making middleboxes someone else's problem: network processing as a cloud service. In *Proceedings of the ACM SIGCOMM Conference* (2012).

Mihai Dobrescu and Katerina Argyraki, EPFL, Lausanne, Switzerland.

© 2015 ACM 0001-0782/15/11 \$15.00



ECSEE
European Conference
Software Engineering
Education 2016

30 June and 1 July
2016

Seon Monastery
Germany

Full Paper
Submission Deadline:
15 March 2016

Conference Website
www.ecsee.eu

**Previous
A.M. Turing Award
Recipients**

1966 A.J. Perlis
1967 Maurice Wilkes
1968 R.W. Hamming
1969 Marvin Minsky
1970 J.H. Wilkinson
1971 John McCarthy
1972 E.W. Dijkstra
1973 Charles Bachman
1974 Donald Knuth
1975 Allen Newell
1975 Herbert Simon
1976 Michael Rabin
1976 Dana Scott
1977 John Backus
1978 Robert Floyd
1979 Kenneth Iverson
1980 C.A.R. Hoare
1981 Edgar Codd
1982 Stephen Cook
1983 Ken Thompson
1983 Dennis Ritchie
1984 Niklaus Wirth
1985 Richard Karp
1986 John Hopcroft
1986 Robert Tarjan
1987 John Cocke
1988 Ivan Sutherland
1989 William Kahan
1990 Fernando Corbató
1991 Robin Milner
1992 Butler Lampson
1993 Juris Hartmanis
1993 Richard Stearns
1994 Edward Feigenbaum
1994 Raj Reddy
1995 Manuel Blum
1996 Amir Pnueli
1997 Douglas Engelbart
1998 James Gray
1999 Frederick Brooks
2000 Andrew Yao
2001 Ole-Johan Dahl
2001 Kristen Nygaard
2002 Leonard Adleman
2002 Ronald Rivest
2002 Adi Shamir
2003 Alan Kay
2004 Vinton Cerf
2004 Robert Kahn
2005 Peter Naur
2006 Frances E. Allen
2007 Edmund Clarke
2007 E. Allen Emerson
2007 Joseph Sifakis
2008 Barbara Liskov
2009 Charles P. Thacker
2010 Leslie G. Valiant
2011 Judea Pearl
2012 Shafi Goldwasser
2012 Silvio Micali
2013 Leslie Lamport
2014 Michael Stonebraker

ACM A.M. TURING AWARD NOMINATIONS SOLICITED

Nominations are invited for the 2015 ACM A.M. Turing Award. This is ACM's oldest and most prestigious award and is presented annually for major contributions of lasting importance to computing. Although the long-term influences of the nominee's work are taken into consideration, there should be a particular outstanding and trendsetting technical achievement that constitutes the principal claim to the award. The recipient presents an address at an ACM event that will be published in an ACM journal. The award is accompanied by a prize of \$1,000,000. Financial support for the award is provided by Google Inc.

**Nomination information and the online submission form
are available on:**
http://amturing.acm.org/call_for_nominations.cfm

**Additional information on the Turing Laureates
is available on:**
<http://amturing.acm.org/byyear.cfm>

**The deadline for nominations/endorsements is
November 30, 2015.**

**For additional information on ACM's award program
please visit: www.acm.org/awards/**



Association for
Computing Machinery

California State University, San Bernardino School of Computer Science and Engineering Assistant Professor/tenure track

The School of Computer Science and Engineering seeks a visionary and collaborative individual for the position of **Assistant Professor/tenure track**. The position is contingent on funding. We are particularly interested in candidates with strengths in embedded systems and signal processing. Others areas of computer engineering will also be considered. The position is primarily to support the B.S. in Computer Engineering degree program. The School also offers the degrees B.S. in

Computer Science, B.S. in Bioinformatics, B.A. in Computer Systems, and M.S. in Computer Science.

The successful candidate will teach core and elective courses in computer engineering and actively contribute to the school's curriculum development. The successful candidate will maintain a robust, ongoing research agenda, supervise student research, and perform academic advising and mentoring. The successful candidate will also provide service to the School, the College, the University, as well as the community and the profession.

Candidates must have earned a Ph.D. or equivalent doctoral degree in Computer Engineering or a closely related discipline. ABD candidates may be considered; the doctoral degree is required at the time of appointment.

A documented commitment to excellence in teaching, research and service is expected. We strongly encourage women and members of other underrepresented groups to apply.

Upon submission of application, applicants are asked to submit a letter of interest that describes 1) teaching experience and philosophy, 2) research and/or scholarship record and future interests, and 3) a diversity statement. In addition, applicants are asked to submit a vita, copy of transcripts, and names and contact information of three references. To learn more about our school, visit our website at <http://cse.csusb.edu>.

Formal review of applications will begin in October 2015 and continue until the position is filled. If you are interested in this opportunity, we invite you to apply at <https://www.governmentjobs.com/jobs/1200769/instructional-faculty-computer-science-engineering/agency/csusb/apply>

Carleton College Department of Computer Science Tenure-Track Position, Assistant Professor level

Carleton College invites applications for a tenure track position in computer science, in any area of specialization, beginning September 1, 2016. A Ph.D. or its imminent completion is required. We expect to make an appointment at the assistant professor level, but highly qualified candidates at higher ranks will be considered.

Carleton is a highly selective liberal arts college with outstanding, enthusiastic students. We seek an equally enthusiastic computer scientist committed to excellence in teaching, curriculum design, ongoing research, and undergraduate research advising. We are particularly interested in candidates committed to teaching a diverse student body. The appointee would be expected to teach courses at a range of levels, from the introductory level to advanced undergraduate electives.

Carleton's Department of Computer Science (cs.carleton.edu) includes eight faculty and an experienced full-time system administrator, and typically graduates between 35 and 60 majors annually.

To apply, please visit the Carleton College website at jobs.carleton.edu and complete the online application, including a cover letter, CV, brief statements describing teaching philosophy and research agenda, a graduate transcript, and contact information for writers of three letters of reference (at least one of which should address your teaching). Questions about the position or the application process may be directed to csjobs@carleton.edu.

Applications completed by December 15, 2015, including reference letters, will receive full consideration.

Carleton is committed to developing its faculty to better reflect the diversity of our student body and American society. Women and members of minority groups are strongly encouraged to apply. Carleton College does not discriminate on the basis of race, color, creed, ethnicity, religion, sex, national origin, marital status, veteran status, actual or perceived sexual orientation, gender identity and expression, status with regard to public assistance, disability, or age in providing employment or access to its educational facilities and activities.

Clarke University Computer Science Faculty

Clarke University, Dubuque, IA, invites applications for a tenure-track faculty in the Computer Science Dept beginning August 2016. Ph.D. in Computer Science or a related field preferred. Master's with experience may be considered. View full posting and apply at <http://clarke.applicantpool.com>.

Concordia University Faculty of Engineering and Computer Science

Concordia University's Faculty of Engineering and Computer Science offers a multi-disciplinary and research-engaged environment dedicated to incubating innovation, excellence and success. Our teaching and research is daring and transformative and contributes significantly to a sustainable intellectual and economic development of our community. We connect ideas with people and we are redefining the university experience.

The Faculty seeks outstanding candidates for:

Department of Computer Science and Software Engineering: A Natural Sciences and Engineering Research Council (NSERC) Canada Research Chair Tier I in the field of **Software Engineering**.

Department of Computer Science and Software Engineering: A Natural Sciences and Engineering Research Council (NSERC) Canada Research Chair Tier II in **Big Data Analytics** – Data Driven Approaches in Different Disciplines.

Department of Computer Science and Software Engineering: One position in the area of **Human Computer Interaction (HCI)** with emphasis in interaction design and development for interactive systems, immersive environments, web or mobile applications.

Department of Computer Science and Software Engineering: One position in the area of Software Engineering with focus on problem analysis, testing and validation.

Concordia Institute for Information Systems Engineering: One position in **Security of Physical Cyber Systems, Systems Engineering, Infrastructure Protection**, and related areas.

Department of Electrical and Computer Engineering: One position in the area of **Avionics**.

For detailed information about working at Concordia, these positions and deadlines, visit concordia.ca/facultypositions

All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents of Canada will be given priority. Concordia University is strongly committed to employment equity within its community, and to recruiting a diverse faculty and staff. The University encourages applications from all qualified candidates, including women, members of visible minorities, Aboriginal persons, members of asexual minorities, persons with disabilities, and others who may contribute to diversification. Detailed applications should include: a CV, teaching and research statements, evidence of teaching effectiveness, and letters from three referees, and shall be accepted in electronic form (PDF) until December 15, 2015 or until positions are filled.

Cornell University Faculty Positions – Computer Science, Cornell Tech

Multiple faculty positions in computer science are available at Cornell's new **Cornell NYC Tech** campus in New York City. Applications are welcome in all areas of computer science and related fields and at all levels, including tenured and tenure-track. Faculty hired in these positions will be within the Department of Computer Science, which will span the Ithaca and New York City campuses, but their teaching and research will be based in New York City. Candidates whose work fits into one of the three initial hubs at Cornell Tech, Connective Media, Healthier Life, and Built Environment, are particularly encouraged.

Candidates must hold a Ph.D., have demonstrated an ability to conduct outstanding research

at the level of tenure-track or tenured faculty in the Department of Computer Science, and also have a strong interest in the technology commercialization and entrepreneurship mission of the campus. In addition, interest in international programs and/or pre-college (K-12) education is advantageous.

This search may include Cornell faculty positions that are part of the Jacobs Technion-Cornell Institute (Jacobs Institute). To ensure full consideration, applications should be received by December 1, 2015, but will be accepted until all positions are filled. Applicants should submit a curriculum vita, brief statements of research and teaching interests, identify one or two top publications to which they have made significant contributions, and the names and contact information of at least three references on-line at: <https://academicjobsonline.org/ajo/jobs/6037>.

Inquiries about your application may be directed to frecruit@cs.cornell.edu

Diversity and Inclusion are a part of Cornell University's heritage. We are a recognized employer and educator valuing AA/EEO, Protected Veterans, and Individuals with Disabilities.

Cornell University **Department of Computer Science** **Faculty Positions – Computer Science**

The Department of Computer Science of Cornell University, based in Ithaca, New York, has multiple faculty positions open. Applications are welcome in all areas of computer science and related fields and at all levels, including tenured, tenure-

track, and lecturer. We particularly encourage applications from women and under-represented minorities. Tenured and tenure track faculty must hold the equivalent of a Ph.D. Lecturers must hold the equivalent of a Masters degree, with a Ph.D. preferred. Applicants must have demonstrated an ability to conduct outstanding research. Successful candidates are expected to pursue an active research program, to teach graduate and undergraduate courses, and to supervise graduate students. The department is ranked among the top computer science departments in the country, with 35 tenured and tenure-track faculty members across all areas of computer science.

Cornell University is located in Ithaca, New York, a city of about 30,000 people in the heart of the Finger Lakes region. Both Cornell and Ithaca offer a wide range of cultural activities, sports, and outdoor activities with the pleasures of both city and country close at hand. Ithaca is Gorges!

To ensure full consideration, applications should be received by December 1, 2015, but will be accepted until all positions are filled. Applicants should submit a curriculum vita and brief statements of research and teaching interests, identify one or two top publications to which they have made significant contributions, and arrange to have at least three reference letters submitted at: <https://academicjobsonline.org/ajo/jobs/6042>.

Inquiries about your application may be directed to frecruit@cs.cornell.edu.

Diversity and Inclusion are a part of Cornell University's heritage. We are a recognized employer and educator valuing AA/EEO, Protected Veterans, and Individuals with Disabilities.

Florida Institute of Technology **CS Faculty Positions**

Florida Tech invites applications for open-rank faculty positions in computer science that begin in Fall 2016. All areas of CS are considered. The CS department offers BS, MS, and PhD degree programs. Florida Tech is in the Space Coast, where NASA Kennedy Space Center is located. According to Brookings Institute's report on "America's Advanced Industries," Space Coast ranks 7th in Advanced Industries' Share of Total 2013 Employment. **Submission of applications and more information are at: cs.fit.edu/careers/**

Equal Opportunity Employer Minorities/
Women/Veterans/Disabled

We are an E-verify employer

EEO is the Law

http://www.fit.edu/hr/documents/eeoc_law.pdf

Georgia Institute of Technology **School of Computational Science and Engineering** **Tenure-Track Faculty**

Computational Science and Engineering solves real-world problems in science, engineering, health, and social domains, by using high-performance computing, modeling and simulation, and large-scale "big data" analytics. The School of Computational Science and Engineering of the College of Computing at the Georgia Insti-



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Faculty Position in Data Science and Machine Learning

Ecole polytechnique fédérale de Lausanne

The School of Computer and Communication Sciences at EPFL invites applications for faculty positions in computer and communication sciences. We are seeking candidates for tenure-track assistant professor as well as for senior positions.

Successful candidates will develop an independent and creative research program, participate in both undergraduate and graduate teaching, and supervise PhD students.

The school is seeking candidates in the fields of data science and machine learning, including application of these techniques in natural language processing, speech recognition, information retrieval, and similar fields.

EPFL offers internationally competitive salaries, significant start-up resources, and outstanding research infrastructure.

To apply, please follow the application procedure at <https://academicjobsonline.org/ajo/jobs/6192>

The following documents are requested in PDF format: cover letter, curriculum vitae including publication list, brief statements of research and teaching interests, names and addresses (including e-mail) of 3 references for junior positions and 6 for senior positions. Screening will start on **December 1, 2015**. Further questions can be addressed to:

Prof. James Larus

Recruiting Committee

School of Computer and Communication Sciences

EPFL

CH-1015 Lausanne

recruiting.ic@epfl.ch

For additional information on EPFL, please consult:
<http://www.epfl.ch> or <http://ic.epfl.ch>

EPFL is an equal opportunity employer.

STATE UNIVERSITY OF NEW YORK AT BUFFALO

Department of Computer Science and Engineering

The State University of New York at Buffalo Department of Computer Science and Engineering invites candidates to apply for **multiple tenured and tenure-track faculty positions** beginning in the 2016-2017 academic year. We invite candidates from all areas of computer science and engineering, especially Computer Vision and Pattern Recognition, Artificial Intelligence, all aspects of Big Data, including Cyber Security, Cyber Physical Systems (or Internet of Things) and Human Computer Interaction. Applicants must have a Ph.D. in computer science or a related area by August 2016 and demonstrate potential for excellence in research, teaching, service and mentoring. We are particularly looking for candidates who can operate effectively in a diverse community of students and faculty, and share our vision of helping all constituents reach their full potential.

Applications will be accepted from **October 15, 2015 to January 15, 2016. Applicants must submit their application electronically via www.ubjobs.buffalo.edu.** Any questions can be directed to Search Committee Co-Chairs, Prof. Rohini Srihari and Chang-wen Chen at cse-recruit@buffalo.edu. The University at Buffalo is an Equal Opportunity Employer.

Computer Science and Engineering Department

Housed in a new \$75M building, and as a part of School of Engineering and Applied Sciences, the Computer Science and Engineering department offers both BA and BS degrees in Computer Science and a BS in Computer Engineering (accredited by ABET), a combined 5-year BS/MS program, a minor in Computer Science, and two joint programs (BA/MBA and Computational Physics) as well as MS and PhD programs.

The department has 37 tenured/tenure-track faculty, 4 teaching faculty, and approximately 750 undergraduate majors, 470 masters students, and 160 PhD students. Eighteen faculty including 16 junior faculty have been hired since 2010, and we are continuing to expand. Two members of our faculty currently hold key university leadership positions and seven members of our faculty are IEEE and/or ACM Fellows. Our faculty members are actively involved in cutting-edge research and successful interdisciplinary programs and centers devoted to biometrics; bioinformatics; biomedical computing; computational and data science and engineering, document analysis and recognition; high performance computing; information assurance and cyber security; embedded, networked and distributed systems, and sustainable transportation. Our annual research expenditure exceeds 4.6 Millions on average over the last five years.

The State University of New York at Buffalo (UB)

The State University of New York at Buffalo (UB) is New York's largest and most comprehensive public university, with approximately 20,000 undergraduate students and 10,000 graduate students.

City and Region

Buffalo is the second largest city in New York state, and was rated the 10th best place to raise a family in America by Forbes magazine in 2010 due to its short commutes and affordability. Located in scenic Western New York, Buffalo is near the world-famous Niagara Falls, the Finger Lakes, and the Niagara Wine Trail. The city is renowned for its architecture and features excellent museums, dining, cultural attractions, and several professional sports teams, a revitalized downtown waterfront as well as a growing local tech and start-up community. Buffalo is home to Z80, a start-up incubator, and 43 North, the world's largest business plan competition.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Lecturer Position Available

The State University of New York at Buffalo Department of Computer Science and Engineering invites candidates to apply for a non-tenure track lecturer position beginning in Fall 2016. We invite candidates from all areas of computer science and computer engineering who have a passion for teaching to apply. We are particularly looking for candidates who can operate effectively in a diverse community of students and faculty, and share our vision of helping all constituents reach their full potential.

The department has a strong commitment to hiring and retaining a lecturer for this career-oriented position, renewable for an unlimited number of 3-year terms. Lecturers are eligible for the in-house titles of Teaching Assistant Professor, Teaching Associate Professor and Teaching Professor. The department encourages and supports lecturers to conduct research in addition to teaching and service. Current lecturers have been conducting successful research with both internal and external funding. The department is undertaking both research and education initiatives in several exciting areas related to Internet of Things (IoT) including Big Data, cloud computing, cyber security and cyber physical systems and embedded systems. This lecturer position also has opportunities to greatly contribute to the development of new certificate and degree problems at both the undergraduate and M.S. levels in the department.

Ideally, applicants should have a PhD degree in computer science, computer engineering, or a related field, by August 15, 2016. Exceptional applicants with M.S. degree will also be considered. The ability to teach at all levels of the undergraduate curriculum is essential, as is potential for excellence in teaching, mentoring, service and research. A background in computer science and computer engineering education, a commitment to K-12 outreach, and addressing the recruitment and retention of underrepresented students are definite assets.

Duties include teaching and development of undergraduate Computer Science and Computer Engineering courses (with an emphasis on lower-division), advising undergraduate students, as well as participation in department and university governance (service). Contribution to research is encouraged.

Review of applications will begin on January 15, 2016, but will continue until the position is filled. **Applicants must submit their application electronically via www.ubjobs.buffalo.edu.**

Any questions can be directed to Search Committee Co-Chairs, Prof. Rohini Srihari and Chang-wen Chen at cse-recruit@buffalo.edu. The University at Buffalo is an Equal Opportunity Employer.

The Department, School and University

Housed in a new \$75M building, and as a part of School of Engineering and Applied Sciences, the Computer Science and Engineering department offers both BA and BS degrees in Computer Science and a BS in Computer Engineering (accredited by ABET), a combined 5-year BS/MS program, a minor in Computer Science, and two joint programs (BA/MBA and Computational Physics) as well as MS and PhD programs.

The department has 37 tenured/tenure-track faculty, 4 teaching faculty, and approximately 750 undergraduate majors, 470 masters students, and 160 PhD students. Eighteen faculty including 16 junior faculty have been hired since 2010, and we are continuing to expand.

The State University of New York at Buffalo (UB) is New York's largest and most comprehensive public university, with approximately 20,000 undergraduate students and 10,000 graduate students.

City and Region

Buffalo is the second largest city in New York state, and was rated the 10th best place to raise a family in America by Forbes magazine in 2010 due to its short commutes and affordability. Located in scenic Western New York, Buffalo is near the world-famous Niagara Falls, the Finger Lakes, and the Niagara Wine Trail. The city is renowned for its architecture and features excellent museums, dining, cultural attractions, and several professional sports teams, a revitalized downtown waterfront as well as a growing local tech and start-up community. Buffalo is home to Z80, a start-up incubator, and 43 North, the world's largest business plan competition.

tute of Technology seeks tenure-track faculty at all levels. This year, our focus application areas include urban systems, biomedical and health, materials and manufacturing, and national security. Applicants must have an outstanding record of research, a sincere commitment to teaching, and interest in engaging in substantive interdisciplinary research with collaborators in other disciplines.

Georgia Tech is located in the heart of metro Atlanta, a home to more than 5.3 million people and nearly 150,000 businesses, a world-class airport, lush parks and green spaces, competitive schools and numerous amenities for entertainment, sports and restaurants that all offer a top-tier quality of life. From its diverse economy, global access, abundant talent and low costs of business and lifestyle, metro Atlanta is a great place to call "home." Residents have easy access to arts, culture, sports and nightlife, and can experience all four seasons, with mild winters that rarely require a snow shovel.

For best consideration, applications are due by December 16, 2015, submitted online through <https://academicjobsonline.org/ajo/jobs/6009>.

The application material should include a full academic CV, teaching and research statements, a list of at least three references and up to three publications. Georgia Tech is an Affirmative Action/Equal Opportunity Employer. Applications from women and under-represented minorities are strongly encouraged.

For more information about Georgia Tech's School of Computational Science and Engineering please visit: <http://www.cse.gatech.edu/>

Macalester College **Assistant Professor**

Applications are invited for a tenure-track Computer Science position at Macalester College to begin Fall, 2016. Candidates must have or be completing a PhD in CS and have a strong commitment to both teaching and research in an undergraduate liberal arts environment. Areas of highest priority include computer and data security and privacy, mobile and ubiquitous computing, human-computer interaction, and visualization. See <http://www.macalester.edu/mscs> for details. Evaluation of applications will begin December 1. Apply URL: <https://academicjobsonline.org/ajo/jobs/5794>

New York University **Courant Institute of Mathematical Sciences** **Department of Computer Science** **Faculty**

The department expects to have several regular faculty positions beginning in September 2016 and invites candidates at all levels. We will consider outstanding candidates in any area of computer science.

Faculty members are expected to be outstanding scholars and to participate in teaching at all levels from undergraduate to doctoral. New appointees will be offered competitive salaries and startup packages, with affordable housing within a short walking distance of the department. New York University is located in Greenwich Village, one of the

most attractive residential areas of Manhattan.

The department has 35 regular faculty members and several clinical, research, adjunct, and visiting faculty members. The department's current research interests include algorithms, cryptography and theory; computational biology; distributed computing and networking; graphics, vision and multimedia; machine learning and data science; natural language processing; scientific computing; and verification and programming languages.

Collaborative research with industry is facilitated by geographic proximity to computer science activities at AT&T, Facebook, Google, IBM, Bell Labs, NEC, and Siemens.

Please apply at <https://cs.nyu.edu/webapps/facapp/register>

To guarantee full consideration, applications should be submitted no later than December 1, 2015; however, this is not a hard deadline, as all candidates will be considered to the full extent feasible, until all positions are filled. Visiting positions may also be available.

New York University is an equal opportunity/affirmative action employer.

North Dakota State University **Assistant/Associate Professor**

Computer Science Department seeks to fill a tenure-track Assistant/Associate Professor position in Software Engineering starting Fall, 2016 or thereafter.

PhD required. NDSU offers degrees at all levels in Computer Science and Software Engineer-



LEHIGH
UNIVERSITY

Multiple Tenure-Track Positions in the **Department of Computer Science and Engineering**

Lehigh is recruiting for four new tenure-track faculty positions in Computer Science and Engineering to start in August 2016 as part of its recently announced Data X strategic initiative. Applications are invited for (1) one position in core computer systems; (2) one position in data-rich consumer analytics; (3) one position in large-scale digital media production, distribution, and analysis; and (4) one position in connected health. The first of these is at open rank, while the other three positions are at the Assistant Professor level. Outstanding candidates at all ranks and in all areas of computer science and engineering will be considered. See <http://www.cse.lehigh.edu> for further details on the CSE Department, and <http://www.lehigh.edu/datax> for details on the interdisciplinary Data X initiative.

Successful applicants will hold a Ph.D. in Computer Science, Computer Engineering, or a closely related field. Candidates must demonstrate a strong commitment to quality undergraduate and graduate education, and the potential to develop and conduct high-impact research programs with external support. Applicants should have an interest in teaching core courses in computer science, interdisciplinary Data X courses, and courses in their own research areas.

The faculty of the Computer Science and Engineering department maintains an outstanding international reputation in a variety of research areas, and includes ACM and IEEE fellows as well as six NSF CAREER award winners.

Lehigh University is a private, highly selective institution that is consistently ranked among the top national research universities by U.S. News & World Report. Located in Bethlehem, Pennsylvania, Lehigh is 80 miles west of New York City and 50 miles north of Philadelphia, providing an accessible and convenient location that offers an appealing mix of urban and small town lifestyles.

Applications can be submitted at Academic Jobs Online (<https://academicjobsonline.org/ajo/Lehigh/CSE>) and should include a cover letter, vita, both teaching and research statements, and contact information for at least three references. Review of applications will begin in November 2015 and will continue until the positions are filled.

Lehigh University is an affirmative action/equal opportunity employer and does not discriminate on the basis of age, color, disability, gender, gender identity, genetic information, marital status, national or ethnic origin, race, religion, sexual orientation, or veteran status. Lehigh is a 2010 recipient of an NSF ADVANCE Institutional Transformation Grant for promoting the careers of women in academic science and engineering. The University provides comprehensive benefits including domestic partner benefits (see also <http://www.lehigh.edu/worklifebalance/>). Lehigh Valley Inter-regional Networking & Connecting (LINC) is a newly created regional network of diverse organizations designed to assist new hires with dual career, community and cultural transition needs: please contact infcdap@lehigh.edu for more information. Questions concerning this search may be sent to faculty-search@cse.lehigh.edu.

ing. Research and teaching excellence is expected, normal teaching load is three courses/year. The department has 16 Faculty; 4 Lecturers; 200 graduate students (Master's and Ph.D.) and 350 BS/BA undergraduate majors.

NDSU is ranked in the top 108 universities in the Carnegie category of "Very High Research Activity". Fargo is a clean, growing metropolitan area of 250,000 that consistently ranks near the top in national quality-of-life surveys. We have low levels of crime and pollution, excellent schools, short commutes, and proximity to the Minnesota lake country. There is a symphony, an opera, a domed stadium, a community theater, three colleges, a research technology park and many other amenities. See <https://jobs.ndsu.edu/> for more information.

NDSU is an EEO/AA-MF/ Vet/ Disability.

Job closing will be November 15, 2015 or until position is filled.

Princeton University **Department of Computer Science** **Tenure Track Position(s)**

The Department of Computer Science at Princeton University invites applications for faculty positions at the Assistant Professor level. We are accepting applications in all areas of Computer Science. Applicants must demonstrate superior research and scholarship potential as well as teaching ability.

A PhD in Computer Science or a related area is required. Candidates should expect to receive their PhD before Fall, 2016. More senior appointments may be considered for extraordinary candidates at the associate and full professor levels. Successful candidates are expected to pursue an active research program and to contribute significantly to the teaching programs of the department. Applicants should include a CV and contact information for at least three people who can comment on the applicant's professional qualifications.

For full consideration we recommend that applicants apply by December 1, 2015, though we will continue to review applications past that date.

Princeton University is an equal opportunity employer. All qualified applicants will receive consideration for employment without regard to race, color, religion, sex, national origin, disability status, protected veteran status, or any other characteristic protected by law. This position is subject to the University's background check policy.

You may apply online at: <http://jobs.cs.princeton.edu/>

Qatar University **Assistant/Associate/Full Research Professor** **in Cyber Security**

Qatar University invites applications for research faculty positions at all levels.

Candidates will cultivate and lead research projects at the KINDI Center for Computing Research in the area of Cyber Security.

Qatar University offers competitive benefits package including a 3-year renewable contract, tax free salary, free furnished accommodation, and more.

Apply by posting your application on the QU online recruitment system at careers.qu.edu.qa under "College of Engineering"



Faculty Positions **Department of Computer Science**

The Department of Computer Science at Virginia Tech (www.cs.vt.edu) seeks applicants for tenure-track faculty positions in three areas: interactive computing, cyber security, and data analytics. Candidates should have a Ph.D. in Computer Science or related field at the time of appointment, a rank-appropriate record of scholarship and collaboration in computing and interdisciplinary areas, sensitivity to issues of diversity in the campus community, and will be required to teach at the undergraduate and/or graduate levels. The position requires occasional travel to professional conferences and meetings.

Tenure-track Assistant Professor in Interactive Computing – Blacksburg, VA

Strong candidates from any area related to interactive computing are encouraged to apply. Exceptional candidates at higher ranks will also be considered. Candidates who complement existing strengths in human-computer interaction, graphics, intelligent user interfaces, visualization, visual analytics, human interaction with big data, augmented reality, tangible interfaces, human-robot interaction, game design, creativity support, or computing in the arts and humanities are especially encouraged. Candidates have opportunities for collaboration in the interdisciplinary Center for Human-Computer Interaction (hci.vt.edu) which includes 30 faculty across campus, the Institute for Creativity, Arts, and Technology (icat.vt.edu) housed in the new Moss Center for the Arts, and the Discovery Analytics Center (dac.cs.vt.edu). Applications must be submitted online to <http://jobs.vt.edu> for posting #TR0150108. Applicant screening will begin on November 20, 2015 and continue until each position is filled. Inquiries should be directed to **Dr. Chris North, Search Committee Chair**, north@vt.edu.

Tenure-track Assistant Professor in Cyber Security – Blacksburg, VA

Candidates with expertise in cyber security, including technologies for and applications in information security, network security, and trustworthy computing are encouraged to apply. Candidates focusing on security issues of cyber-physical systems, embedded systems, sensor networks, robotics, Internet of Things (IoT), etc. are especially encouraged. The candidate will join the CS department and also participate in an interdisciplinary team of five faculty in Advanced Manufacturing and share common space and equipment, leveraging established labs and the Commonwealth Center for Advanced Manufacturing (www.ccam-va.com/), a public-private partnership in Virginia. There is an active group of cyber security faculty in CS and ECE departments collaborating in research as well as graduate and undergraduate education (see: www.cyber.vt.edu/). Applications must be submitted online to <http://jobs.vt.edu> for posting #TR0150107. Applicant screening will begin on November 20, 2015 and continue until the position is filled. Inquiries should be directed to **Dr. Ali Butt, Search Committee Chair**, butta@cs.vt.edu.

Associate/Full Professor in Data Analytics – National Capital Region (NCR)

Candidates with research depth and breadth in data analytics, data mining, "big data", or data science are encouraged to apply. Candidates working at the intersection of data analytics and cyber-security and at the intersection of data analytics and urban computing are especially encouraged. Candidates should present a proven ability to initiate and sustain collaborations within computing as well as with application specialists. The department is home to the Discovery Analytics Center (dac.cs.vt.edu) that leads "big data" research on campus. The successful candidate will contribute to the research and graduate programs in the NCR and collaborate with faculty at Virginia Tech's campus in Blacksburg, VA. The NCR campus (www.ncr.vt.edu) is located near the Washington D.C./Falls Church area and houses the Virginia Tech Research Center (www.ncr.vt.edu/arlington) in Arlington, VA. Applications must be submitted online to <http://jobs.vt.edu> for posting #TR0150106. Applicant screening will begin on November 20, 2015 and continue until the position is filled. Inquiries should be directed to **Dr. Naren Ramakrishnan, Search Committee Chair**, naren@cs.vt.edu.

The Department of Computer Science has 40 research oriented tenure-track faculty and ~10 postdocs/research faculty. There are a total 12 NSF/DOE CAREER awardees in the department. Research expenditures for FY2015 were \$412 thousand per tenure-track faculty member (i.e., a total of \$15.5 million); total research funding at the beginning of FY2015 was \$43.4 million. BS, MS, and PhD degrees are offered, with a growing enrollment of over 610 undergraduate majors (14% women) and over 270 PhD/MS students. In 2010, CS@VT was ranked 5th in the country in recruiting quality of CS undergrads by the *Wall Street Journal*. The department is in the College of Engineering, whose undergraduate program was ranked 8th and graduate program was ranked 12th among public engineering schools in 2014 by *U.S. News & World Report*.

Early applications are encouraged. We welcome applications from women or minorities. Salary for suitably qualified applicants is competitive and commensurate with experience. Selected candidates must pass a criminal background check prior to employment.

Virginia Tech is an AA/EEO employer; applications from members of underrepresented groups are especially encouraged.

San Diego State University
Department of Computer Science
Chair of Computer Science

Department of Computer Science at SDSU seeks candidates for the Chair position with a PhD in Computer Science or a closely related field, and a sustained record of supported research. The Department is a dynamic and growing unit looking for a visionary Chair to lead it into its next phase of expansion.

We strive to build and sustain a welcoming environment for all. SDSU is seeking applicants with commitment to working effectively with individuals from diverse backgrounds and members of underrepresented groups.

For more details and application procedures, please apply via <http://apply.interfolio.com/31841>.

SDSU is a Title IX, equal opportunity employer. A full version of this ad can be found at: <http://cs.sdsu.edu/>

South University of Science and Technology (SUSTC)
Professor/Associate Professor/Assistant Professorship in Computer Science

The University

Established in 2012, the South University of Science and Technology (SUSTC) is a public institution funded by the municipal of Shenzhen, a special economic zone city in China. Shenzhen is a major city located in Southern China, situated immediately north of Hong Kong Special

Administrative Region. As one of China's major gateways to the world, Shenzhen is the country's fast-growing city in the past two decades. The city is the high-tech and manufacturing hub of southern China, home to the world's third-busiest container port, and the fourth-busiest airport on the Chinese mainland. A picturesque coastal city, Shenzhen is also a popular tourist destination and was named one of the world's 31 must-see tourist destinations in 2010 by The New York Times. By the end of 2013, there were over 10 million permanent residents in the city.

The South University of Science and Technology is a pioneer in higher education reform in China. The mission of the University is to become a globally recognized institution which emphasizes academic excellence and promotes innovation, creativity and entrepreneurship. The teaching language at SUSTC is bilingual, either English or Putonghua.

Set on five hundred acres of wooded landscape in the picturesque Nanshan (South Mountain) area, the new campus offers an ideal environment suitable for learning and research. The new campus occupies more than 1,940,000 square meters, and the total construction area will be approximately 630,000 square meters.

Call for Application

The South University of Science and Technology now invites applications for the faculty position in Computer Science Department which is currently under rapid construction. It is seeking to appoint a number of tenured or tenure track positions in all ranks. Candidates with research inter-

ests in all mainstream fields of Computer Science will be considered. These positions are full time posts. SUSTC adopts the tenure track system, which offers the recruited faculty members a clearly defined career path.

Candidates should have demonstrated excellence in research and a strong commitment to teaching. A doctoral degree is required at the time of appointment. Candidates for senior positions must have an established record of research, and a track-record in securing external funding as PI.

As a State-level innovative city, Shenzhen has chosen independent innovation as the dominant strategy for its development. It is home to some of China's most successful high-tech companies, such as Huawei and Tencent. As a result, SUSTC considers entrepreneurship is one of the main directions of the university, and good starting supports will be provided for possible initiatives. SUSTC encourages candidates with intention and experience on entrepreneurship to apply.

Terms & Applications

To apply, please send curriculum vitae, description of research interests and statement on teaching to cshire@sustc.edu.cn.

SUSTC offers competitive salaries, fringe benefits including medical insurance, retirement and housing subsidy, which are among the best in China. Salary and rank will commensurate with qualifications and experience. More information can be found at <http://talent.sustc.edu.cn/en>

Candidates should also arrange for at least three letters of recommendation sending directly



Florida International University is a comprehensive university offering 340 majors in 188 degree programs in 23 colleges and schools, with innovative bachelor's, master's and doctoral programs across all disciplines including medicine, public health, law, journalism, hospitality, and architecture. FIU is Carnegie-designated as both a research university with high research activity and a community-engaged university. Located in the heart of the dynamic south Florida urban region, our multiple campuses serve over 55,000 students, placing FIU among the ten largest universities in the nation. Our annual research expenditures in excess of \$100 million and our deep commitment to engagement have made FIU the go-to solutions center for issues ranging from local to global. FIU leads the nation in granting bachelor's degrees, including in the STEM fields, to minority students and is first in awarding STEM master's degrees to Hispanics. Our students, faculty, and staff reflect Miami's diverse population, earning FIU the designation of Hispanic-Serving Institution. At FIU, we are proud to be "Worlds Ahead!" For more information about FIU, [visit fiu.edu](http://visit.fiu.edu).

The School of Computing and Information Sciences (SCIS) seeks exceptionally qualified candidates for tenure-track and tenured faculty positions at all levels as well as non-tenure track faculty positions at the level of Instructor, including visiting instructor appointments. SCIS is a rapidly growing program of excellence at the University, with 30 tenure-track faculty members and over 2,000 students, including over 80 Ph.D. students. SCIS offers B.S., M.S., and Ph.D. degrees in Computer Science, an M.S. degree in Telecommunications and Networking, an M.S. degree in Cybersecurity, and B.S., B.A., and M.S. degrees in Information Technology. SCIS has received over \$22M in the last four years in external research funding, has six research centers/clusters with first-class computing and support infrastructure, and enjoys broad and dynamic industry and international partnerships.

Open-Rank Tenure Track/Tenured Positions (Job ID# 508676)

SCIS seeks exceptionally qualified candidates for tenure-track and tenured faculty positions at all levels. We seek well-qualified candidates in all areas; researchers in the areas of computer systems, cybersecurity, cognitive computing, data science, health informatics, and networking are particularly encouraged to apply. Preference will be given to candidates who will enhance or complement our existing research strengths.

Ideal candidates for junior positions should have a record of exceptional research in their early careers. Candidates for senior positions must have an active and proven record of excellence in funded research, publications, and professional service, as well as a demonstrated ability to develop and lead collaborative research projects. In addition to developing or expanding a high-quality research program, all successful applicants must be committed to excellence in teaching at both the graduate and undergraduate levels. An earned Ph.D. in Computer Science or related disciplines is required.

Non-tenure track instructor positions (Job Opening 507474)

We seek well-qualified candidates in all areas of Computer Science and Information Technology. Ideal candidates must be committed to excellence in teaching a variety of courses at the undergraduate level. A graduate degree in Computer Science or related disciplines is required; significant prior teaching and industry experience and/or a Ph.D. in Computer Science is preferred.

HOW TO APPLY:

Qualified candidates for open-rank faculty positions are encouraged to apply to (Job Opening ID #508676); and candidates for instructor positions are encouraged to apply to (Job Opening ID # 507474). **Submit applications at facultycareers.fiu.edu** and attach cover letter, curriculum vitae, statement of teaching philosophy, research statement, etc as individual attachments. Candidates will be required to provide names and contact information for at least three references who will be contacted as determined by the search committee. To receive full consideration, applications and required materials should be received by December 31st, 2015. Review will continue until position is filled.

If you are interested in a visiting appointment please contact the department directly by emailing Dr. Mark Weiss at Weiss@cis.fiu.edu. All other applicants should apply by going to facultycareers.fiu.edu.

FIU is a member of the State University System of Florida and an Equal Opportunity, Equal Access Affirmative Action Employer. All qualified applicants will receive consideration for employment without regard to race, color, religion, sex, national origin, disability status, protected veteran status, or any other characteristic protected by law.

to the above email account. The search will continue until the position is filled.

For informal discussion about the above posts, please contact Professor Yifan Chen, Interim Head of Department of Computer Science, by phone +86-755-88018506 or email: chen.yf@sustc.edu.cn.

Swarthmore College Computer Science Department Assistant Professor

The Computer Science Department invites applications for one tenure-track position and multiple visiting positions at the rank of Assistant Professor to begin Fall semester 2016.

Swarthmore College has a strong institutional commitment to excellence through diversity and inclusivity in its educational program and employment practices. The College actively seeks and welcomes applications from candidates with exceptional qualifications, particularly those with demonstrated commitments to a more inclusive society and world.

Swarthmore College is a small, selective, liberal arts college located 10 miles outside of Philadelphia. The Computer Science Department offers majors and minors at the undergraduate level.

Applicants must have teaching experience and should be comfortable teaching a wide range of courses at the introductory and intermediate level. Candidates should additionally have a strong commitment to involving undergraduates

in their research. A Ph.D. in Computer Science at or near the time of appointment is required.

For the tenure-track position, we are particularly interested in applicants whose areas will complement and broaden our program, including theory and algorithms, programming languages, and systems areas. Strong applicants in other areas will also be considered.

For the visiting position, strong applicants in any area will be considered.

For the tenure-track position, priority will be given to complete applications received by December 15. For the visiting position, priority will be given to complete applications received by February 15. Applications for both positions will continue to be accepted after these dates until the positions are filled.

Applications should include a cover letter, vita, teaching statement, research statement, and three letters of reference, at least one (preferably two) of which should speak to the candidate's teaching ability. In your cover letter, please briefly describe your current research agenda; what would be attractive to you about teaching in a liberal arts college environment; and what background, experience, or interests are likely to make you a strong teacher of Swarthmore College students.

Tenure-track applications are being accepted online at <https://academicjobsonline.org/ajo/jobs/6161>.

Visiting applications are being accepted online at <https://academicjobsonline.org/ajo/jobs/6173>. Candidates may apply for both positions.

The College of New Jersey Assistant or Associate Professor - Three Positions

The Department of Computer Science at The College of New Jersey (TCNJ) invites applications for three tenure-track Assistant or Associate Professor faculty positions in computer science starting August 2016. One position may be filled at the Associate Professor rank.

For each position, we are seeking broadly trained applicants from all areas of computer science, who have the potential to establish a highly visible undergraduate research program and who are passionate about teaching computer science.

The successful candidates must be able to teach computer science courses at all levels of the curriculum and be able to develop innovative, upper-level courses that complement and diversify the department's offerings.

Teaching and research are mutually supportive activities at TCNJ. Candidates should be strongly committed to the teacher-scholar model, to maintaining both high-quality teaching and an active and productive research program involving highly motivated undergraduates, and to seeking external funding consistent with this mission. Faculty members serve as academic advisors and have service responsibilities within the Department and College.

Candidates must have a Ph.D. in Computer Science, or a closely related field; demonstrated excellence in teaching; and a strong commitment to the education of, and research with, undergraduates.

ROCHESTER INSTITUTE OF TECHNOLOGY Computing and Information Sciences Multiple Openings for Fall 2016

The B. Thomas Golisano College of Computing and Information Sciences at the Rochester Institute of Technology invites applications and nominations for the following faculty positions:

- Four full-time tenure-track/tenured positions to be part of a cluster hire in **cybersecurity** (two positions in **Computing Security**, one in **Computer Science**, and one in **Software Engineering**). Candidates will be considered at the assistant, associate, and full professor ranks (#2031BR).

RIT views cybersecurity as an interdisciplinary field that exists at the intersection of society and technology. RIT is making a major investment in cybersecurity, and the cluster hires will further strengthen the Institute's research competency and profiles in related disciplines. While successful candidates will be appointed to different departments, they will be expected to collaborate across departments and colleges, and bring discipline perspectives and contributions to cybersecurity from their home departments. They will be expected to play critical roles in advancing research and education in cybersecurity, and lead efforts in building a dedicated interdisciplinary research center on campus.

We are also seeking candidates for the following openings:

- Three tenure-track assistant professor positions in **Computer Science** (#2032BR and #2033BR [2])
- One tenure-track assistant professor position in **Computing Security** (#2022BR)

- One tenure-track assistant or associate professor position in **Software Engineering** (#2019BR)
- Two lecturer positions in **Software Engineering** (#2027BR and #2028BR)
- One visiting lecturer position in **Computer Science** (#1905BR)
- One postdoctoral researcher position in **Human Computer Interaction** in the **Information Sciences and Technologies** department (#1735BR)
- One postdoctoral researcher position in **Software Engineering** (#1976BR)

Candidates should visit <http://careers.rit.edu/faculty> and refer to the BR numbers listed above for specific information about the positions and the application process. Refer to www.rit.edu for information about RIT and the B. Thomas Golisano College of Computing and Information Sciences.

The B. Thomas Golisano College of Computing and Information Sciences is home to the departments of Computer Science, Software Engineering, Information Sciences and Technologies, and Computing Security, and the School of Interactive Games and Media, offering a variety of BS and MS degrees as well as the Ph.D. in Computing and Information Sciences. The college has 110 faculty and over 3000 undergraduate and 800 graduate students.

RIT does not discriminate. RIT is an equal opportunity employer that promotes and values diversity, pluralism, and inclusion. For more information or inquiries, please visit *RIT/TitleIX* or the U.S. Department of Education at *ED.Gov*

R·I·T

-Selected as one of the top colleges to work for by The Chronicle of Higher Education

To Apply:

To apply, please submit the following materials electronically and consolidated into a **single PDF file**, by email to cs@tcnj.edu (for any materials larger than five megabytes, please provide a download link). **Please put the candidate's name and "Tenure Track Position" in the email subject line.**

- ▶ Letter of application
- ▶ Curriculum vitae
- ▶ Statement of teaching philosophy
- ▶ Evidence of exemplary undergraduate teaching, including examples of relevant work from previous students
- ▶ Statement of the applicant's plan for continued scholarly research and professional activity
- ▶ Copies of sample publications to demonstrate scholarship and professional accomplishments
- ▶ Copies of all undergraduate and graduate transcripts.

In addition, a complete application requires three confidential letters of recommendation, including:

- ▶ At least one commenting on teaching effectiveness, and
- ▶ At least one commenting on scholarly research and professional engagement.

Letters must be emailed directly by the recommenders to cs@tcnj.edu with the name of the candidate in the subject line. Letters forwarded by the applicant will not be considered to be part of the completed package.

Review of **complete** applications will begin on

December 1, 2015. Applications will be accepted and reviewed until the positions are filled.

To enrich education through diversity, The College of New Jersey is an Equal Opportunity Employer. The College has a strong commitment to achieving diversity among faculty and staff, and strongly encourages women and members of underrepresented groups to apply.

The University of Michigan, Ann Arbor
Department of Electrical Engineering and Computer Science
Computer Science and Engineering Division
Faculty Positions

Applications and nominations are solicited for multiple faculty positions in the Computer Science and Engineering (CSE) Division. Highly qualified candidates from all areas of computer science and computer engineering will be considered for positions at all levels and are encouraged to apply. Candidates whose interests fall in the areas of data analytics and data management will also be considered for affiliation with the Michigan Institute for Data Science (MIDAS), a new multi-disciplinary Institute at the University of Michigan (midas.umich.edu).

Qualifications include an outstanding academic record, a doctorate or equivalent in computer science or computer engineering, and a strong commitment to teaching and research. The college is especially interested in candidates who can contribute, through their research, teaching, and/or service, to the diversity and ex-

culture of the academic community.

The University of Michigan is a Non-Discriminatory/Affirmative Action Employer with an Active Dual-Career Assistance Program.

Applications must be received by January 1, 2016. To apply, please complete the form at: <https://www.eecs.umich.edu/eecs/etc/csejobs/>

Electronic applications are **strongly** preferred, but you may alternatively send resume, teaching statement, research statement and names of three references to:

Professor Atul Prakash, Chair, CSE Faculty Search
 Department of Electrical Engineering and Computer Science
 University of Michigan
 2260 Hayward Street
 Ann Arbor, MI 48109-2121

Trinity College, Hartford, Connecticut
Assistant Professor of Computer Science

Applications are invited for a tenure-track position in computer science at the rank of Assistant Professor to start in the fall of 2016. Candidates must hold a Ph.D. in computer science at the time of appointment. We are seeking candidates with teaching and research interests in applied areas associated with Big Data (such as database and information systems, data mining and knowledge discovery, machine learning and cloud computing), but other related areas will be seriously considered. Applications should be submitted to: <https://trincoll.edu>



Multiple Tenure-Track or Tenured Faculty Positions in Computer Science

The Department of Computer Science at National University of Singapore (NUS) invites applications for several tenure-track or tenured faculty positions. We have positions dedicated to cyber security or big data analytics as well as positions open to all areas of computer science. While our main focus is on the assistant professor level, we also welcome exceptional candidates at the associate and full professor levels. For applications at the assistant professor level, candidates should demonstrate excellent research potential and a strong commitment to teaching. Candidates at more senior levels should have an established record of outstanding research achievements.

The Department of Computer Science at NUS is highly ranked internationally. It enjoys ample research funding, moderate teaching load, excellent facilities, and extensive international collaborations. The department covers all major research areas in computer science and boasts a thriving PhD program that attracts the brightest students from the region and beyond. More information is available at <http://www.comp.nus.edu.sg/>.

NUS offers highly competitive salaries and is situated in Singapore, an English-speaking cosmopolitan city and a meeting point of many cultures, both the east and the west. Singapore offers high-quality education, healthcare, and extremely low tax rates.

Interested candidates are invited to send, via electronic submission, the following materials to the Chair of the CS Faculty Search Committee, Prof. David Hsu, at csrec@comp.nus.edu.sg:

- A cover letter that clearly indicates main research interests
- Curriculum Vitae
- A teaching statement
- A research statement

Please arrange for

- at least 3 references

to be sent directly to the same e-mail address or provide the contact information.

Application review will commence on October 1, 2015 and continue until the positions are filled. To ensure maximal consideration, please submit your application by December 15, 2015.



Announcement of an open position at the Faculty of Informatics, Vienna University of Technology, Austria

**PROFESSOR
of
SECURITY**

The Vienna University of Technology (TU Wien) invites applications for a professor position at the Faculty of Informatics. Excellent junior scientists are explicitly encouraged to apply.

The successful candidate will have an outstanding research record in the field of computer security. He/She should have a comprehensive research agenda, preferably spanning multiple subareas of the field. We expect the successful candidate to establish links to several of the main research areas of the Faculty of Informatics. Experience in raising funds and in managing scientific research projects is highly appreciated.

We offer excellent working conditions in an attractive research environment in a city with an exceptional quality of life.

For a more detailed announcement and information on how to apply, please go to: www.informatik.tuwien.ac.at/vacancies

Application deadline: **December 31, 2015**

peopleadmin.com/. Consideration of applications will begin on December 15, 2015. Trinity College is an Equal-Opportunity/Affirmative-Action employer.

University of Alabama
Tenured/Tenure-Track Faculty Position,
Computer Science
All Areas of Computer Science

The Department of Computer Science at the University of Alabama invites applications for a tenure-track faculty position at either the Assistant or Associate level, to begin August 2016. Outstanding candidates in all areas of computer science will be considered.

Located in Tuscaloosa, AL, the University of Alabama is the capstone of higher education in the State. The student-centered research institution is also the State's largest, with roughly 36,000 students. Housed in the College of Engineering, the Computer Science Department has 22 faculty members (14 tenured/tenure track faculty), and approximately 500 undergraduate and 50 graduate students.

For additional details, visit <http://cs.ua.edu/> or contact Dr. Susan Vrbosky (faculty.search@cs.ua.edu). To apply, visit: <http://facultyjobs.ua.edu/postings/37597>. Review of applications will begin November 30 and will continue until the position is filled. The University of Alabama is an equal opportunity/affirmative action employee. Women and minority applicants are particularly encouraged to apply.

University of California, Santa Barbara
Department of Computer Science
Assistant Professor

The University of California, Santa Barbara invites applicants for two tenure-track appointments at the Assistant Professor level in the Department of Computer Science, effective July 1, 2016. Applicants must have a Ph.D. in computer science or a closely related field. We are looking for exceptional individuals from all areas of computer science who have the potential to become outstanding research and education leaders. We are especially interested in the broad areas of Big Data, focusing on large-scale data gathering, mining and learning, and Physical Computing, focusing on tight integration of computational and physical systems. Qualified candidates will be considered for appointment to the Eugene Aas Endowed Chair.

The department seeks individuals with promise and a proven track record of research excellence, a deep commitment to educating undergraduate and graduate students, and professional leadership abilities. We are particularly interested in candidates who value cross-disciplinary and collaborative research.

UC Santa Barbara is a premier public research university. The Department of Computer Science is part of the College of Engineering, in which all five departments are ranked among the top ten by the National Research Council (NRC). With 34 full-time faculty and more than 120 doctoral students currently, the Department of Computer Science has witnessed a rapid rise over the past ten years, both in size and stature. We are looking for candi-

dates who will add to the department's dynamic culture and stimulating intellectual environment.

Additional information about the department and our academic programs can be found at <http://www.cs.ucsb.edu>.

Primary consideration will be given to applications received by December 15, 2015, however the position will remain open until filled. **Applications should be submitted electronically as PDF documents to:**

<https://recruit.ap.ucsb.edu/apply/JPF00544>

Applications must include a detailed resume, research and teaching statements, three reference letters and three publications.

The department is especially interested in candidates who can contribute to the diversity and excellence of the academic community through research, teaching and service. The University of California is an Equal Opportunity/Affirmative Action Employer and all qualified applicants will receive consideration for employment without regard to race, color, religion, sex, sexual orientation, gender identity, national origin, disability status, protected veteran status, or any other characteristic protected by law.

University of Georgia
Assistant or Associate Professor

Internet of Things (IoT) tenure-track opening in the Department of MIS in the Terry College of Business at The University of Georgia. Search for position # 30012 at apps.itos.uga.edu/ach



Computer Science
Department
Assistant Professor

The Computer Science Department in the College of Engineering and Applied Sciences at the University at Albany SUNY is searching for tenure-track faculty beginning either January 2016 or Fall 2016. While the search is primarily at the Assistant Professor level, we will consider senior applicants with appropriate credentials. We seek candidates with research expertise in cyber-security and privacy, an area that the University has identified as a priority for development.

Applicants must have a Ph.D. in Computer Science, Computer Engineering, Informatics, or a closely related discipline.

For a complete job description and application procedures, visit:
<https://albany.interviewexchange.com/jobofferdetails.jsp?JOBID=53468>

Questions regarding the position may be addressed to CSCyber@albany.edu.

The Computer Science Department offers Bachelor's, Master's and Ph.D. degrees. For additional information, please visit <http://www.cs.albany.edu/>.

The University at Albany is an EO/AA/IRCA/ADA



COLUMBIA ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Open Rank Faculty Position in the
Department of Electrical Engineering

Columbia Engineering is pleased to invite applications for faculty positions in the Department of **Electrical Engineering** at Columbia University in the City of New York. Applications at the assistant professor, and in exceptional cases, at the associate professor and full professor levels, will be considered.

Applications are sought in all areas of Electrical Engineering. Candidates must have a Ph.D. or its professional equivalent by the starting date of the appointment. Applicants for this position at the Assistant Professor and Associate Professors without tenure must demonstrate the potential to do pioneering research and to teach effectively. Applicants for this position at the tenured level (Associate or Full Professor) must have a demonstrated record of outstanding research accomplishments, excellent teaching credentials and established leadership in the field.

The successful candidate is expected to contribute to the advancement of their field and the department by developing an original and leading externally funded research program, and to contribute to the undergraduate and graduate educational mission of the Department. Columbia fosters multidisciplinary research and encourages collaborations with academic departments and units across Columbia University. The Department 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.

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 **December 1st, 2015** will receive full consideration.

Applicants can consult <http://www.ee.columbia.edu/ee-faculty-positions> and <http://www.ee.columbia.edu> for more information about the position and the department.

Columbia is an affirmative action/equal opportunity employer with a strong commitment to the quality of faculty life.

University of Illinois Springfield Assistant Professor, Computer Science

The Computer Science Department at the University of Illinois at Springfield (UIS) invites applications for an Assistant Professor, tenure track position to begin August, 2016. A Ph.D. in Computer Science or closely related field is required. Candidates who have completed all course work and have successfully defended the dissertation will be considered if such documentation is provided. The position involves graduate and undergraduate teaching, supervising student research, and continuing your research. Many of our classes are taught online. All areas of expertise will be considered, but the ability to teach core computer science is of special interest for the Department. The UIS Computer Science Department offers innovative programs in software engineering and information systems security. In the fall 2014, there were over 1,000 majors in online and on campus bachelor's and master's computer science degree programs. Students may choose to emphasize software engineering or information security. We are planning to expand our offerings in the area of big data/data analytics. Applications must be submitted online at <https://jobs.uis.edu/job-board>. Applicants should navigate to the Faculty category and Computer Science position. Applicants should create a profile, complete the online application, and upload a cover letter, curriculum vita, statements of teaching philosophy and research interests, and contact information for three references. Letters of reference and transcripts may be requested at a later date. Review of applications will begin on

January 15, 2016 and continue until the position is filled or the search is terminated. Only those applicants who complete the online application and upload all required materials will be considered. To maintain the integrity and confidentiality of the search, applicants are asked to upload their own materials. Contact Person: Bobbie Fults. Email: bfult1@uis.edu, tel: 217-206-6770

Located in the state capital, the University of Illinois Springfield is one of three campuses of the University of Illinois. You are encouraged to visit the university web page at <http://www.uis.edu> and the department web page at <http://csc.uis.edu>. UIS is an affirmative action/equal opportunity employer with a strong institutional commitment to recruitment and retention of a diverse and inclusive campus community. Women, minorities, veterans, and persons with disabilities are encouraged to apply.

University of Illinois at Urbana-Champaign Department of Electrical and Computer Engineering (ECE) Positions in Computing

The Department of Electrical and Computer Engineering (ECE) at the University of Illinois at Urbana-Champaign invites applications for faculty positions at all levels and in all areas in computing, broadly defined, with particular emphasis on big data and its applications, including data analytics; data center and storage systems; parallel, high-performance, and energy-efficient computing; reliable and secure computing; distributed

computing; bio-inspired computing; verification; wired/wireless networking; social networking; mobile, wearable sensing & applications; and computational genomics. From the transistor and the first computer implementation based on von Neumann's architecture to the Blue Waters petascale computer – the fastest computer on any university campus – ECE ILLINOIS faculty have always been at the forefront of computing research and innovation. Applications are encouraged from candidates whose research programs specialize in core as well as interdisciplinary areas of electrical and computer engineering. The department is engaged in exciting new and expanding programs for research, education, and professional development, with strong ties to industry. The ECE Department has recently settled into its new 235,000 sq. ft. net-zero energy design building, which is a major campus addition with maximum space and minimal carbon footprint.

Qualified senior candidates may also be considered for tenured full Professor positions as part of the Grainger Engineering Breakthroughs Initiative (<http://graingerinitiative.engineering.illinois.edu>), which is backed by a \$100-million gift from the Grainger Foundation to support research in big data and bioengineering, broadly defined. In addition, the University of Illinois is home to Blue Waters petascale computer, which is supported by the National Science Foundation and developed and operated by the University of Illinois' National Center for Supercomputing Applications. Qualified candidates may be hired as Blue Waters Professors who will be provided substantial allocations on and expedited access



Florida Institute of Technology
High Tech with a Human Touch™

TWO FACULTY POSITIONS

In

“Human-Computer Interaction” and “Modeling and Simulation”

Florida Institute of Technology

(www.fit.edu)

School of Human-Centered Design, Innovation and Arts

School of Human-Centered Design, Innovation and Arts at Florida Institute of Technology in Melbourne, Florida, invites applications for two full time assistant professor positions to begin January 2016: one in **human-computer interaction**; and another in **modeling and simulation** and connected disciplines such as computer-aided design and virtual engineering. The school strives to provide excellent research, teaching and service to the university community and to the world in human-centered design, cognitive engineering and human-systems integration. Areas of research expertise within the school include advanced interaction media, creativity, design thinking, modeling and simulation, complexity analysis, industrial design, organization design and management, life-critical systems. Applicants should have a Ph.D. degree in human-computer interaction, modeling and simulation, computer science or related areas. Of particular interest are candidates having an outstanding research record, demonstrated interest/experience in teaching at the undergraduate and graduate levels and supervising graduate students. Experience beyond Ph.D. is preferred.

Our school was recently created as an extension of the Human-Centered Design Institute. It gathers several domains of expertise including aeronautics, space, nuclear engineering, medicine, automotive, education and culture. Our proximity to NASA Kennedy Space Center and our location on the Space Coast offer a great environment for hard work and fun! Our school contributes to the education and training of socio-technical leaders of the 21st century. It combines strong theoretical knowledge and proactive hands-on endeavors. Graduate students are involved in research and innovation projects and are strongly encouraged to publish. We also welcome students from the other colleges to our transversal design platform.

Applications must consist of a cover letter, current curriculum vitae, copies of recent publications, a statement of interest and research achievements, and evidence of teaching effectiveness. Candidates must also arrange to have three letters of reference sent directly to:

Dr. Guy A. Boy, University Professor and Dean
School of Human-Centered Design, Innovation and Arts
Florida Institute of Technology
Melbourne, FL 32901 USA
gboy@fit.edu

Applications should reach the department no later than October 30, 2015. All inquiries about the position should be directed to Dr. Boy (gboy@fit.edu). For additional information, please visit our website at <http://research.fit.edu/hcdi/>.

Florida Institute of Technology is committed to employment equity.



ADVERTISING IN CAREER OPPORTUNITIES

How to Submit a Classified Line Ad: Send an e-mail to acmm mediasales@acm.org. Please include text, and indicate the issue/ or issues where the ad will appear, and a contact name and number.

Estimates: An insertion order will then be e-mailed back to you. The ad will by typeset according to CACM guidelines. NO PROOFS can be sent. Classified line ads are NOT commissionable.

Rates: \$325.00 for six lines of text, 40 characters per line. \$32.50 for each additional line after the first six. The MINIMUM is six lines.

Deadlines: 20th of the month/2 months prior to issue date. For latest deadline info, please contact:

acmm mediasales@acm.org

Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at:

<http://jobs.acm.org>

Ads are listed for a period of 30 days.

For More Information Contact:

**ACM Media Sales
at 212-626-0686 or
acmm mediasales@acm.org**

to the supercomputer. To be considered as a Blue Waters Professor, candidates need to mention Blue Waters as one of their preferred research areas in their online application, and include a reference to Blue Waters in their cover letter.

Please visit <http://jobs.illinois.edu> to view the complete position announcement and application instructions. Full consideration will be given to applications received by December 15, 2015, but applications will continue to be accepted until all positions are filled.

The University of Illinois conducts criminal background checks on all job candidates upon acceptance of a contingent offer.

Illinois is an EEO Employer/Vet/Disabled www.inclusiveillinois.illinois.edu.

University of North Florida School of Computing Jacksonville, Florida Assistant Professor

A tenure-earning position at the **Assistant Professor** level in the School of Computing at the University of North Florida is available beginning August 2016. The successful candidate must have an earned Ph.D. (or will have earned a Ph.D. degree by August 2016) in computer science, information systems, information sciences, information technology, or equivalent from an accredited institution. A demonstrated ability to establish/conduct a fundable research program and supervise research in computing and information science related topics at both the undergraduate and graduate levels

will be expected from the successful candidate.

Preferred areas of research include big data, machine learning, mobile computing, cloud computing, and cybersecurity. The ability to mentor and guide students through theses work in a vibrant graduate program is essential. Teaching responsibilities will include teaching both undergraduate and graduate courses in computing. All four School of Computing undergraduate programs are accredited by ABET. Collegiality, active service and participation in governance in accordance with the vision and mission of the school, college and university are also expected.

Applicants must complete an online application at www.unfjobs.org and upload the following documents to be considered for this position: a letter of interest, a curriculum vitae, a list of three references with contact information, a statement of purpose detailing teaching experience, and your vision for research. A pre-employment background check is performed on candidates selected for employment.

If you have any questions about this position, please contact Dr. Bob Roggio at broggio@unf.edu. Initial review of applications will begin on October 15, 2015.

The University of North Texas Department of Computer Science and Engineering Assistant/Associate Professor and Lecturer

The University of North Texas (UNT) invited applications for the following faculty positions in the **Department of Computer Science and Engineering (CSE)**.

Tenure or tenure track positions:

Position 1: Assistant or Associate Professor in **Computer Security**, including cryptography, cloud and network security, mobile security, intrusion detection, secure hardware and software systems, and vulnerability and threat analysis.

Position 2: Assistant or Associate Professor in **Computer Systems**, including computer architecture, real-time operating systems, runtime systems and virtualization, memory and storage systems, distributed systems, embedded systems, resilient, secure and survivable systems, and performance measurement and tuning.

The candidates are expected to teach CSE undergraduate and graduate courses, develop a strong research program funded by external sources, support and mentor graduate students, and provide service to the University and the profession. Minimum qualifications include an earned doctorate in computer science, computer engineering or a closely related field. For the Assistant Professor position, a strong publication record and the potential to succeed in securing research funding and mentoring graduate students are required. For the senior positions, a sustained record of providing mentoring to junior faculty, advising graduate students, providing service to the University and profession, and securing external funding for research activities with current research funding are also required. Post-doctoral research experience or industrial research experience is preferred.

Non-tenure track position:

Position 3: Lecturer, Senior Lecturer, or Principal Lecturer with a three-year appointment renew-

udjobs
Employer of Choice

Faculty Position in Strategic Computing and Information Sciences Department of Electrical and Computer Engineering

The Department of Electrical and Computer Engineering at the University of Delaware invites applications for a tenure-track Assistant/Associate Professor position in the areas of i) strategic computing, including high performance computing and big data analytics and ii) information sciences, including but not limited to the areas of machine learning, computational sensing and imaging, information theory and communications. Successful applicants will share our vision to grow the department into a leader in research and educational programs. ECE initiatives are supported by over 40,000 square feet of departmental facilities and fueled with over \$10M/year in research expenditures.

The University of Delaware combines a rich history in engineering with a commitment to undergraduate education and scholarly excellence. The University ranks among the top 100 universities in federal R&D support for science and engineering, with external funding exceeding \$200 million annually. Supported by state-of-the-art facilities, research is conducted across all seven colleges and numerous interdisciplinary institutes and centers. The main campus in Newark, Delaware, provides the amenities of a vibrant college town with convenient access to the major cities of the East Coast. The newly opened 194,000-square-foot Harker Interdisciplinary Science and Engineering Laboratory greatly expands opportunities and resources for interdisciplinary research and active learning, and the recently acquired 272-acre Science, Technology and Advanced Research (STAR) campus offers additional opportunities for academic and commercial collaborative research and development.

Candidates must have a Ph.D. degree in electrical or computer engineering, computer science or a related discipline. Applicants should submit a curriculum vita, a one-page statement of teaching experience and interests, a 3-6-page research proposal and a list of at least four references. The University of Delaware values diversity and is supportive of the needs of dual-career couples; women and minorities are especially encouraged to apply.

To submit applications please visit apply.interfolio.com/31482

Equal Employment Opportunity

The University of Delaware is an Equal Opportunity Employer which encourages applications from minority group members, women, individuals with a disability and veterans. The University's Notice of Non-Discrimination can be found at <http://www.udel.edu/aboutus/legal/notice.html>. Employment offers will be conditioned upon successful completion of a criminal background check. A conviction will not necessarily exclude you from employment.

Dare to be first.



TEMPLE UNIVERSITY

Department of Computer and Information Sciences Junior and Senior Tenure-Track Faculty Positions

Applications are invited for tenure-track, open rank, faculty positions in the CIS department at Temple University.

The junior position is in the systems area, including but not limited to: computer architecture and languages, storage systems, distributed computing, software systems, and embedded computing.

The senior position for Associate, Full, or Endowed Chair Professor is open to all areas of computer science/engineering. Applicants for the senior position are expected to have an outstanding and continuing track record of research and funding.

The CIS Department has two undergraduate degree programs, one in Computer Science (CS) and one in Information Science and Technology (IS&T), a master's program in CIS, a master's program in IS&T, and a PhD program in CIS. The department, recently moved to a new Science Education and Research Center, has undergone considerable growth in research in the past few years, during which research funding, publication rates, and the number of Ph.D. students has significantly increased. Located in Philadelphia, the 5th-largest city in the U.S., Temple University is a comprehensive research institution that serves more than 39,000 students. It is the 26th largest university in the United States and is one of the nation's leading centers of professional education.

Please submit applications with all requested information online at <https://academicjobsonline.org/ajol/jobs/6035>. For further information check <http://www.cis.temple.edu> or send email to **Dr. Krishna Kant**, Chair, Faculty Search Committee at kkant@temple.edu with Subject=TTPosition. Review of candidates will begin on October 15, 2015. Candidates are encouraged to apply by this deadline to receive priority consideration. Review continues until the positions are filled. Senior position applicants, who are interested in the general future directions of the department, can also directly contact the department chair **Dr. Jie Wu** at jiewu@temple.edu with Subject=SeniorPosition.

able annually, depending on performance and the availability of funding. The primary responsibility is teaching at the undergraduate and graduate level. Additional expectations include participating in departmental activities, assisting with ABET accreditation, and providing career guidance to undergraduate students.

The Computer Science and Engineering department is home to 1,139 bachelors students, 151 masters students and 90 Ph. D. students. The UNT CSE department is the home department of the interdisciplinary Center for Information and Computer Security and the lead academic institution of the NSF Industry/University Cooperative Research Center for Net-centric and Cloud Software and Systems. Additional information about the department and the centers are available at the websites: www.cse.unt.edu, www.cics.unt.edu, and netcentric.cse.unt.edu, respectively.

Application Procedure:

All applicants must apply online to: <https://faculty-jobs.unt.edu>. Submit nominations and questions regarding the tenure track position in computer security (system identification number 6001152) to Dr. Ram Dantu (Ram.Dantu@unt.edu), the tenure track position in computer systems (system identification number 6001153) to Dr. Bill Buckles (Bill.Buckles@unt.edu), and lecturer position (system identification number 6001154) to Dr. Phil Sweany (Philip.Sweany@unt.edu).

Application Deadline:

The committee will begin its review of applications on November 1, 2015 and continue to ac-

cept and review applications until the positions are filled.

The University:

As the nation's 24th largest public university and the largest, most comprehensive in the Dallas-Fort Worth area, UNT is dedicated to providing an excellent educational experience to its 37,000 students while powering the North Texas region, state and nation through innovative education and research. UNT is strategically located in Denton, Texas, a vibrant city with a lively arts and music culture.

The University of North Texas is an AA/ADA/EOE committed to diversity in its educational programs.

University of Notre Dame Department of Computer Science and Engineering Assistant, Associate, or Full Professor

The Department of Computer Science and Engineering at the **University of Notre Dame** invites applications for **Assistant, Associate, or Full Professor**. Excellent candidates in all areas will be considered.

The Department offers the Ph.D. degree and undergraduate Computer Science and Computer Engineering degrees. Faculty are expected to excel in classroom teaching and to lead highly-visible research projects that attract substantial external funding.

Applicants should submit a CV, statement

of teaching and research interests, and contact information for three professional references at <http://apply.interfolio.com/30991>.

Equal Opportunity Employment Statement

The University of Notre Dame seeks to attract, develop, and retain the highest quality faculty, staff and administration. The University is an Equal Opportunity Employer, and is committed to building a culturally diverse workplace. We strongly encourage applications from female and minority candidates and those candidates attracted to a university with a Catholic identity. Moreover, Notre Dame prohibits discrimination against veterans or disabled qualified individuals, and requires affirmative action by covered contractors to employ and advance veterans and qualified individuals with disabilities in compliance with 41 CFR 60-741.5(a) and 41 CFR 60-300.5(a).

University of Rochester Department of Computer Science Faculty Positions in Computer Science: Experimental Systems and AI/HCI

The University of Rochester Department of Computer Science <https://www.cs.rochester.edu> seeks applicants for two tenure track positions in the broad areas of experimental systems and AI/HCI. Candidates must have a PhD in computer science or a related discipline. Our goal is to hire the best and the brightest who will thrive in our environment and complement our existing strengths in



IST AUSTRIA
Institute of Science and Technology

Call for Postdoctoral Fellows in EXECUTABLE BIOLOGY

Executable biology is the study of biological systems as reactive dynamic systems (i.e., systems that evolve with time in response to external events).

Are you a talented and motivated scientist looking for an opportunity to conduct research at the intersection of **BIOLOGY and COMPUTER SCIENCE** at a young, dynamic institution that fosters scientific excellence and interdisciplinary collaboration?

Apply at www.ist.ac.at/executablebiology
Deadline December 31, 2015





UNIVERSITY AT ALBANY
State University of New York

Assistant Professors Computer Engineering

The College of Engineering and Applied Sciences at the University at Albany – SUNY is seeking applicants for assistant professor tenure-track faculty positions beginning either Spring or Fall 2016 in the recently-created Computer Engineering Department. While the search is primarily at the Assistant Professor level, we will consider more senior applicants with appropriate credentials.

The successful candidates will have expertise in one or more areas of computer or electrical engineering including, but not limited to, hardware design, embedded systems, mobile and ubiquitous computing, networking, and software engineering. Applicants should be committed to teaching, research, and service in an interdisciplinary environment, and will be encouraged to participate in curriculum development.

We would welcome applications from a cluster of faculty involved in collaborative research in computer engineering.

Applicants must have a Ph.D. in Computer Engineering, Electrical Engineering, or a closely-related discipline, or anticipate completion by August 2016. For a complete job description and application procedures, please visit: <https://albany.interviewexchange.com/jobofferdetails.jsp?JOBID=55601>

The College of Engineering and Applied Sciences also includes Computer Science, Informatics, and Information Studies departments. For additional information on the College and its departments, please visit: www.albany.edu/ceas/.

The University at Albany is an EO/AA/IRCA/ADA Employer

artificial intelligence, human-computer interaction, systems, and theory. Applicants in all areas of computer science are encouraged to apply. We are especially interested in growing research strengths in interdisciplinary work that combines AI and HCI or systems, in addition to research in databases, networks, scalable systems, security, and computer vision.

Apply online at <https://www.rochester.edu/faculty-recruiting/login>

Consideration of applications at any rank will begin immediately and continue until all interview slots are filled. Candidates should apply no later than January 1, 2016 for full consideration. Applications that arrive after this date risk being overlooked or arriving after the interview schedule has been filled.

The Department of Computer Science is research focused, with a distinguished history of contributions in artificial intelligence, HCI, systems, and theory. We have a collaborative culture and strong ties to electrical and computer engineering, brain and cognitive science, linguistics, and several departments in the medical center. We also have a growing Institute for Data Science with potential for synergistic collaboration opportunities and more joint hires. Over the past decade, a third of the department's PhD graduates have won tenure-track faculty positions, and its alumni include leaders at major research laboratories such as Google, Microsoft, and IBM.

The University of Rochester is a private, Tier I research institution located in western New York State. It consistently ranks among the top 30 institutions, both public and private, in federal funding for research and development. The university has made substantial investments in computing infrastructure through the Center for Integrated Research Computing (CIRC) and the Health Sciences Center for Computational Innovation (HSCCI). Half of all undergraduates have traditionally gone on to post-graduate or professional education. The university includes the Eastman School of Music, a premiere music conservatory, and the University of Rochester Medical Center, a major medical school, research center, and hospital system. The greater Rochester area is home to over a million people, including 80,000 students who attend its 8 colleges and universities. Traditionally strong in optics research and manufacturing, it was recently selected by the Department of Defense as the hub of a \$360M-plus Integrated Photonics Institute for Manufacturing Innovation.

The University of Rochester has a strong commitment to diversity and actively encourages applications from groups underrepresented in higher education. The University is an Equal Opportunity Employer.

EOE Minorities/Females/Protected Veterans/Disabled

University of Texas at Austin
Department of Computer Science
Assistant/Associate/Full Professor positions

The Department of Computer Science of the University of Texas at Austin invites applications for tenure-track positions at all levels. Outstanding candidates in all areas of Computer Science will be considered, particularly in computer vision

and machine learning. All tenured and tenure-track positions require a Ph.D. or equivalent degree in computer science or a related area at the time of employment.

Successful candidates are expected to pursue an active research program, to teach both graduate and undergraduate courses, and to supervise graduate students. The department is ranked among the top ten computer science departments in the country.

It has 41 tenured and tenure-track faculty members across all areas of computer science. Many of these faculty participate in interdisciplinary programs and centers in the University, including the Texas Advanced Computing Center (TACC), and those in Computational and Applied Mathematics, Computational Biology, and Neuroscience.

Austin, the capital of Texas, is a center for high-technology industry, including companies such as IBM, Dell, Freescale Semiconductor, Advanced Micro Devices, National Instruments, AT&T, Intel and Samsung. For more information please see the department web page <http://www.cs.utexas.edu/>.

The department prefers to receive applications online, beginning September 4, 2015. To submit yours, please visit <http://services.cs.utexas.edu/recruit/faculty>.

Applicants for an assistant professor position must have at least three (3) referees send letters of reference directly to the address provided. Applicants for a tenured position (associate or full professor) must have at least six (6) referees send letters of reference directly.

Inquiries about your application may be directed to faculty-search@cs.utexas.edu. For full consideration of your application, please apply by January 31, 2016. Complete applications (including all reference letters) will begin being reviewed on December 15, 2015. Women and minority candidates are especially encouraged to apply. The University of Texas is an Equal Opportunity Employer.

The University of Texas at San Antonio
Department of Computer Science
Faculty Positions in Computer Science

The Department of Computer Science at The University of Texas at San Antonio invites applications for a tenured/tenure-track position at the assistant or associate professor level, starting Fall 2016. Interested candidates with research focus in one or more areas of system software, data science, high performance computing, or cloud computing are encouraged to apply.

See <http://www.cs.utsa.edu/fsearch> for information on the Department and application instructions. Screening of applications will begin immediately. Full consideration will be given to applications received by January 4, 2016, and the search will continue until the positions are filled or the search is closed. The University of Texas at San Antonio is an Affirmative Action/Equal Opportunity Employer.

Department of Computer Science
RE: Faculty Search
The University of Texas at San Antonio
One UTSA Circle
San Antonio, TX 78249-0667
Phone: 210-458-4436

Vanderbilt University
The Department of Electrical Engineering and Computer Science (EECS)
Two Tenured/Tenure-Track (T/TK) Faculty Positions in Computer Science

The Department of Electrical Engineering and Computer Science (EECS) AT Vanderbilt University seeks candidates for two tenured/tenure-track (T/TK) faculty positions in Computer Science. Appointments at all ranks will be considered; we prefer to fill at least one position at the assistant professor level. Areas of primary emphasis are Big Data and Cyber-physical Systems. The Vanderbilt CS program provides a unique, collaborative, diverse, and interdisciplinary research environment. New trans-institutional programs create opportunities for research on issues of broad significance across multiple disciplines. Successful candidates are expected to teach at the undergraduate and graduate levels, develop and grow vigorous programs of externally funded research, and enhance the School of Engineering's strategic directions in health care and medicine, security, and energy and natural resources.

Vanderbilt University is an equal-opportunity, affirmative-action employer that aspires to become a leader among peer institutions in making meaningful and lasting progress in responding to the needs and concerns of women and members of under-represented minority groups. **Applications should be submitted on-line at: <https://academicjobsonline.org/ajob/jobs/6129>**. For more information, please visit: <http://engineering.vanderbilt.edu/eeecs/>. Applications will be reviewed on a rolling basis, with an initial deadline of December 1, 2015, but will be accepted until the positions are filled.

Washington College
Department of Mathematics & Computer Science
Assistant Professor

The Department of Mathematics & Computer Science at **Washington College** in Chestertown, MD, invites applications for a tenure-track position in computer science at the rank of **Assistant Professor** beginning in August 2016. For full details visit <http://www.washcoll.edu/offices/human-resources/employment.php>. EOE M/F/D/V

Wesleyan University
Department of Mathematics and Computer Science
Assistant Professor of Computer Science

The Department of Mathematics and Computer Science at **Wesleyan University** invites applications for a **tenure track assistant professorship in Computer Science** to start in fall 2016. Wesleyan currently has a strong research group in theory, programming languages, and software engineering and a 2-1 teaching load. We encourage candidates in all areas of Computer Science to apply but the department is especially interested in databases, machine learning, natural language processing, software engineering and software systems. For description and application procedure see <http://www.wesleyan.edu/mathcs/employment.html>.

Email: cssearch@wesleyan.edu

Fax: 860-685-2571

Apply: <http://academicjobsonline.org/ajob/jobs/5939>



DOI:10.1145/2825814

Dennis Shasha

Upstart Puzzles

Auction Triplets

THERE ARE OBJECTS of four types and n people, each with a budget of \$99. The objective of each one is to acquire three objects of the same type (any type) before anyone else does; for example, if player A acquires three type 1 objects before any other player acquires any three objects of any type, A wins. Assume each auction is resolved through a highest-bid method; that is, the highest bidder pays the amount he or she bids (Vickrey auctions are a possible variant.) Every bid must be an integral number of dollars. If there is a tie, the bidder (if any) who won the immediately preceding auction gets the item if that bidder is one of those who tied. In every other case, the tie ends in a draw, and nobody takes the item in that auction.

By symmetry, there cannot be a guaranteed winning strategy, but the general challenge is to work out probabilistically good strategies, given knowledge of the sequence of item types to be auctioned.

Warm-up. Suppose there are just two players and all items are of the same type. Is there an optimal bidding strategy?

Solution to this warm-up. The best strategy is to bid \$33 every time. If you bid \$33 and your opponent bids less, you will win, and the only way for your opponent to stop you is to bid \$34 or more in the next auction. But if you keep bidding \$33, the opponent will take a second item with \$33 and spend \$67. In that case, you win by bidding \$33 thereafter.

Puzzle 1. Suppose again there are just two players and three items of type T1 and the rest of type T2. Assume player A gets the first item of type T1 for \$12. Can A force a win?

Solution. Yes, by bidding any number divisible by three between \$12



and \$36 inclusive for the next item of type T1. If A wins again, then A bids \$51 on the last T1 item. B must match to avoid losing, so B wins for, say, \$52, but now A has \$51, while B has only \$50. A bets \$17 each time for each element of T2. B cannot stop A. If B wins the second item for \$13 or more, then A does not bid for the last T1 item but can win on T2 by bidding \$29 $(= (99 - 12)/3)$ from then on.

Puzzle 2. Suppose there are two players, A and B, but player A has \$100, and player B has only \$99. Suppose further there are items of only type 1. Can A force a win?

Solution. A starts with \$34. If B ties, then A alternates randomly between \$34 and \$33. If A wins, then A bids \$33 in the next bet. A will win on ties, so B must stop A by bidding \$34. At that point, A bids \$33 again. If B ties and takes a second item, then B has only \$32, so A can continue to bid \$33 and be sure to win.

Puzzle 3. Suppose there are several players and items of only type 1, and player A has two items and $\$m$ left, whereas each other player has at most one item and at most $\$2m-1$ left. Can A force a win?

Solution. Yes. If A bets $\$m$ and someone else beats A, then that other player cannot have more than $\$m-1$ left. If after a tie, A continues to bet $\$m$ or $\$m-1$

then anyone who wins will have at most $\$m-1$ left. Eventually each player but A has $\$m-1$ left.

Now that you get the idea, here are the upstart questions for Auction Triplets.

Upstart 1. In a two-player bidding contest, if one player knows the sequence of items, but the other does not, what is the expected advantage of the player who knows? Assume each item in the sequence is generated uniformly at random across the four types.

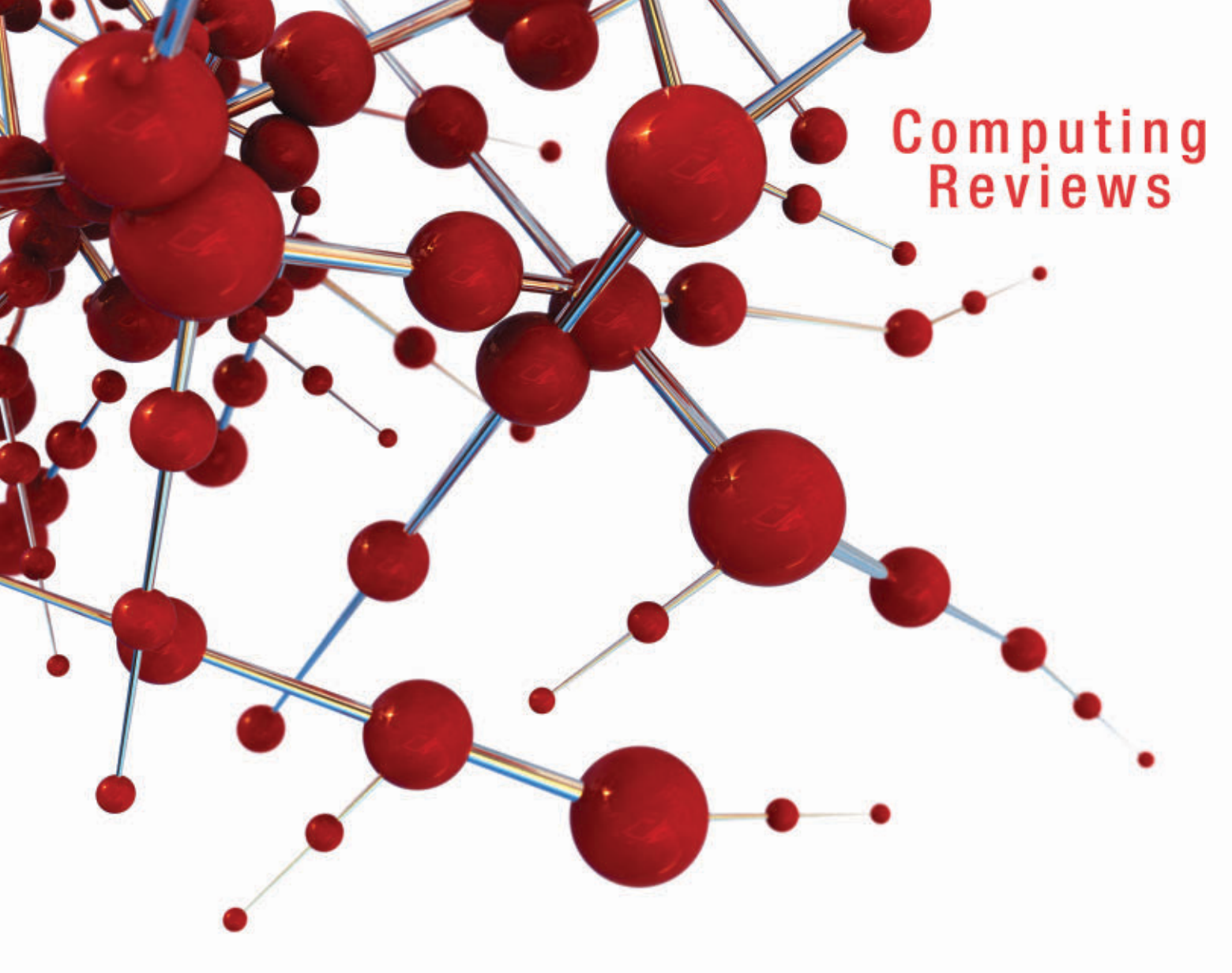
Upstart 2. If there can be an arbitrary number of types and the sequence of the types of the objects to be auctioned is known beforehand, is there any bidding strategy (which may be randomized) that gives the best chance of winning?

Upstart 3. Now generalize the setting of Upstart 1 in which one player does not know the exact sequence of types but does know the distribution of types over some horizon of future items (such as half the next 20 items are T1). 

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, as well as the chronicler of his good friend the omniheurist Dr. Ecco.

Copyright held by author.



Computing Reviews

**CONNECT WITH OUR
COMMUNITY OF EXPERTS.**

www.computingreviews.com



Association for
Computing Machinery

ThinkLoud

They'll help you find the best new books
and articles in computing.

Computing Reviews is a collaboration between the ACM and ThinkLoud.

VEE 2016

12th ACM SIGPLAN/SIGOPS international conference on

Virtual Execution Environments

Atlanta, GA April 2-3, 2016 with ASPLOS

Authors are invited to submit original papers related to virtualization across all layers of the software stack, from high-level language virtual machines down to the microarchitectural level. VEE 2016 accepts both full-length and short papers.

Abstract deadline: November 23, 2015

Paper deadline: November 30, 2015



Image: Courtesy of Chuck Koehler <https://www.flickr.com/photos/cokak/355135172/>, <https://creativecommons.org/licenses/by/2.0/>

General Chair

Vishakha Gupta-Cledat (Intel Labs)

Program Co-chairs

Donald Porter (Stony Brook University)

Vivek Sarkar (Rice University)



in cooperation with



<http://conf.researchr.org/home/vee-2016>