

COMMUNICATIONS

CACM.ACM.ORG OF THE ACM 04/2014 VOL.57 NO.04

Security and Privacy for Augmented Reality Systems



Area Map: Nearest 24-hour Pharmacy



Mortgage Payment: Overdue



Tuesday, 3PM: Job Recruiter Appt.



Calorie Count: Black Coffee, 2



Message Center Pin No.: 1314



Keychain Password: kelf367c22



Most Frequent Site: WatchMyDogAllDay.com

Who Does What in a MOOC?

Multipath TCP

small data where $n = me$

New Technologies Replace Animal Testing

ANITA BORG INSTITUTE

GRACE HOPPER

CELEBRATION OF WOMEN IN COMPUTING



GHC Scholarship applications close on Wednesday, April 16

GHC ABIE Award nominations close on Thursday, May 15

Registration opens June 2

Grace Hopper Celebration • October 8-11, 2014

Phoenix, AZ



ANITA BORG INSTITUTE

WOMEN TRANSFORMING TECHNOLOGY



ACM Books



MORGAN & CLAYPOOL
PUBLISHERS

Publish your next book in the ACM Digital Library

ACM Books is a new series of advanced level books for the computer science community, published by ACM in collaboration with Morgan & Claypool Publishers.

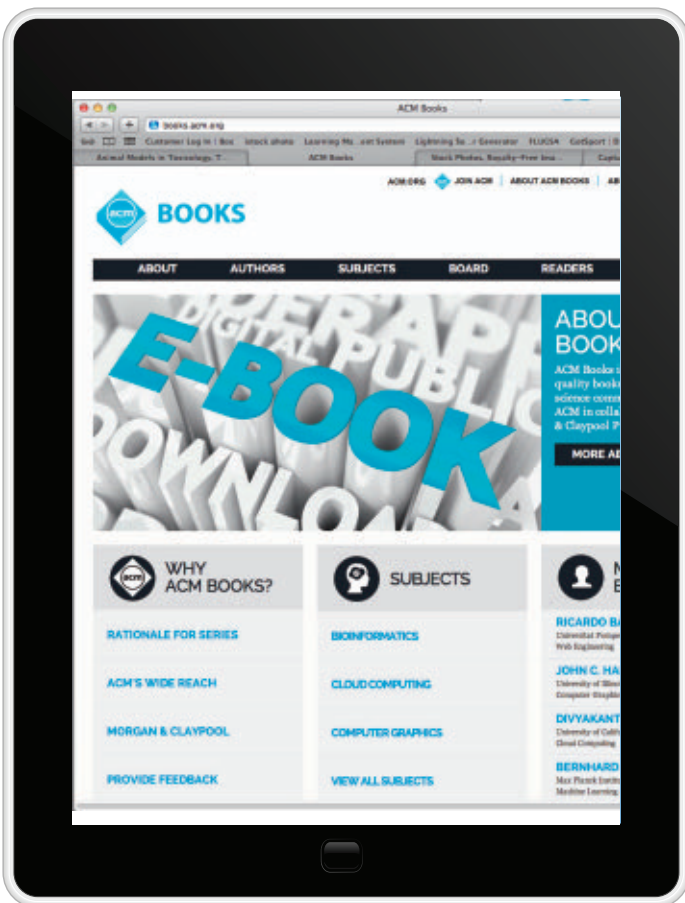
I'm pleased that ACM Books is directed by a volunteer organization headed by a dynamic, informed, energetic, visionary Editor-in-Chief (Tamer Özsu), working closely with a forward-looking publisher (Morgan and Claypool).

—Richard Snodgrass, University of Arizona

books.acm.org

ACM Books

- ◆ will include books from across the entire spectrum of computer science subject matter and will appeal to computing practitioners, researchers, educators, and students.
- ◆ will publish graduate level texts; research monographs/overviews of established and emerging fields; practitioner-level professional books; and books devoted to the history and social impact of computing.
- ◆ will be quickly and attractively published as ebooks and print volumes at affordable prices, and widely distributed in both print and digital formats through booksellers and to libraries and individual ACM members via the ACM Digital Library platform.
- ◆ is led by EIC M. Tamer Özsu, University of Waterloo, and a distinguished editorial board representing most areas of CS.



Proposals and inquiries welcome!

Contact: **M. Tamer Özsu**, Editor in Chief
booksubmissions@acm.org



Association for
Computing Machinery

Advancing Computing as a Science & Profession

Departments

- 5 **Letter from *Communications'* Contributed/Review Articles Co-Chairs**

A Front Row Seat to *Communications'* Editorial Transformation

By Alfred Aho and Georg Gottlob

- 7 **From the President**
The Internet Governance Ecosystem
By Vinton G. Cerf

- 9 **Letters to the Editor**
Code That Missed Mars

- 10 **BLOG@CACM**
Eyes Forward
Mark Guzdial considers why computing education lags behind other sciences, while Daniel Reed weighs balancing immediate research needs against future uncertainty.

- 29 **Calendar**

- 108 **Careers**

Last Byte

- 112 **Future Tense**
Re: Search
For some, data collecting will always be more rewarding than data mining.
By Ken MacLeod

News



- 13 **Using Patient Data for Personalized Cancer Treatments**
Personalized Cancer Treatments Stutter, But Researchers See Mellifluous Future
Patient information databases eventually will help improve health outcomes and support development of new therapies.
By Chris Edwards

- 16 **Speech-To-Speech Translations Stutter, But Researchers See Mellifluous Future**
The practical need for accurate instant or simultaneous machine translations continues to grow as applications multiply.
By Paul Hyman

- 20 **New Models in Cosmetics Replacing Animal Testing**
A European law spurs scientists to develop computational simulations capable of predicting the toxicity of cosmetics.
By Gregory Mone

Viewpoints

- 24 **Technology Strategy and Management MOOCs Revisited, With Some Policy Suggestions**
Assessing the rapidly evolving realm of massive open online courses.
By Michael A. Cusumano

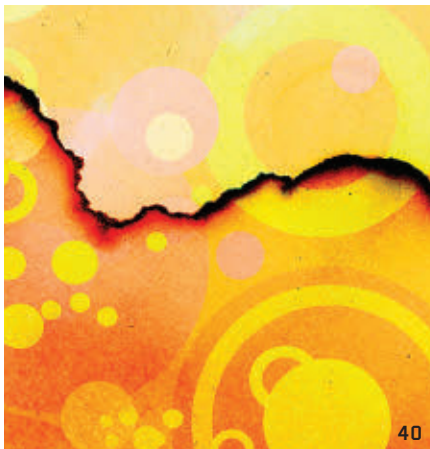
- 27 **Global Computing**
Thinking Outside the Continent
Encouraging the opportunities for digital innovation and invention to flourish in a variety of social environments.
By Michael L. Best

- 30 **Kode Vicious**
This Is the Foo Field
The meaning of bits and avoiding upgrade bogdowns.
By George V. Neville-Neil

- 32 **Viewpoint**
small data, where n = me
Seeking personalized data-derived insights from analysis of our digital traces.
By Deborah Estrin

- 35 **Viewpoint**
Is Multicore Hardware for General-Purpose Parallel Processing Broken?
The current generation of general-purpose multicore hardware must be fixed to support more application domains and to allow cost-effective parallel programming.
By Uzi Vishkin

Practice



40

40 **Rate-Limiting State**
The edge of the Internet is an unruly place.
By Paul Vixie

44 **Major-League SEMAT—**
Why Should an Executive Care?
Becoming better, faster, cheaper, and happier.
By Ivar Jacobson, Pan-Wei Ng, Ian Spence, and Paul E. McMahon

51 **Multipath TCP**
Decoupled from IP, TCP is at last able to support multihomed hosts.
By Christoph Paasch and Olivier Bonaventure

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



About the Cover:
The wonders and benefits of augmented reality applications, made possible via headsets/glasses, smartphones, and other mobile devices, also come with some very real security and privacy concerns. This month's cover story (p. 88) explores the risks and argues that now is the time to address

these issues while the technology is still young. Cover photo by Michael Zhang.

Contributed Articles



58

58 **Who Does What in a Massive Open Online Course?**
Student-participation data from the inaugural MITx (now edX) course—6.002x: Circuits and Electronics—unpacks MOOC student behavior.
By Daniel T. Seaton, Yoav Bergner, Isaac Chuang, Piotr Mitros, and David E. Pritchard

66 **Formally Verified Mathematics**
With the help of computational proof assistants, formal verification could become the new standard for rigor in mathematics.
By Jeremy Avigad and John Harrison

76 **Unifying Functional and Object-Oriented Programming with Scala**
Scala unifies traditionally disparate programming-language philosophies to develop new components and component systems.
By Martin Odersky and Tiark Rumpf

Review Articles



88

88 **Security and Privacy for Augmented Reality Systems**
AR systems pose potential security concerns that should be addressed *before* the systems become widespread.
By Franziska Roesner, Tadayoshi Kohno, and David Molnar

Research Highlights

98 **Technical Perspective**
A 'Reasonable' Solution to Deformation Methods
By Joe Warren

99 **Bounded Biharmonic Weights for Real-Time Deformation**
By Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine-Hornung



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

John White
Deputy Executive Director and COO
 Patricia Ryan
Director, Office of Information Systems
 Wayne Graves
Director, Office of Financial Services
 Russell Harris
Director, Office of SIG Services
 Donna Cappel
Director, Office of Publications
 Bernard Rous
Director, Office of Group Publishing
 Scott E. Delman

ACM COUNCIL

President
 Vinton G. Cerf
Vice-President
 Alexander L. Wolf
Secretary/Treasurer
 Vicki L. Hanson
Past President
 Alain Chesnais
Chair, SGB Board
 Erik Altman
Co-Chairs, Publications Board
 Jack Davidson and Joseph Konstan
Members-at-Large
 Eric Allman; Ricardo Baeza-Yates;
 Radia Perlman; Mary Lou Soffa;
 Eugene Spafford
SGB Council Representatives
 Brent Hailpern; Andrew Sears;
 David Wood

BOARD CHAIRS

Education Board
 Andrew McGettrick
Practitioners Board
 Stephen Bourne

REGIONAL COUNCIL CHAIRS

ACM Europe Council
 Fabrizio Gagliardi
ACM India Council
 Anand S. Deshpande, PJ Narayanan
ACM China Council
 Jianguang Sun

PUBLICATIONS BOARD

Co-Chairs
 Jack Davidson; Joseph Konstan
Board Members
 Ronald F. Boisvert; Marie-Paule Cani;
 Nikil Dutt; Roch Guerrin; Carol Hutchins;
 Patrick Madden; Catherine McGeoch;
 M. Tamer Ozsu; Mary Lou Soffa

ACM U.S. Public Policy Office

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

Computer Science Teachers Association

Chris Stephenson,
 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
 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

Editorial Assistant

Zarina Strakhan

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

Public Relations Coordinator

Virginia Gold

Publications Assistant

Emily Williams

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

acmhelp@acm.org

Letters to the Editor

letters@cacm.acm.org

WEBSITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/guidelines

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

Marc Najork and William Pulleyblank

Board Members

Hsiao-Wuen Hon; Mei Kobayashi;
 Michael Mitzenmacher; Rajeev Rastogi;
 Marc Snir

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;
 Seymour Goodman; Mark Guzdial;
 Rachelle Hollander; Richard Ladner;
 Carl Landwehr; Carlos Jose Pereira de Lucena;
 Beng Chin Ooi; Loren Terveen;
 Marshall Van Alstyne; Jeannette Wing

Q PRACTICE

Co-Chairs

Stephen Bourne and George Neville-Neil

Board Members

Eric Allman; Charles Beeler; Bryan Cantrill;
 Terry Coatta; Stuart Feldman; Benjamin Fried;
 Pat Hanrahan; Tom Limoncelli;
 Marshall Kirk McKusick; Erik Meijer;
 Theo Schlossnagle; Jim Waldo

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

CONTRIBUTED ARTICLES

Co-Chairs

Al Aho and Georg Gottlob

Board Members

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

RESEARCH HIGHLIGHTS

Co-Chairs

Azer Bestavros and Gregory Morrisett

Board Members

Martin Abadi; Amr El Abbadi; Sanjeev Arora;
 Dan Boneh; Andrei Broder; Stuart K. Card;
 Jeff Chase; Jon Crowcroft; Alon Halevy;
 Maurice Herlihy; Norm Jouppi;
 Andrew B. Kahng; Xavier Leroy; Kobbi Nissim;
 Mendel Rosenblum; David Salesin;
 Guy Steele, Jr.; David Wagner;
 Margaret H. Wright

WEB

Chair

James Landay

Board Members

Gene Golovchinsky; Marti Hearst;
 Jason I. Hong; Jeff Johnson;
 Wendy E. MacKay

ACM Copyright Notice

Copyright © 2014 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 acmhelp@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



DOI:10.1145/2582611

A Front Row Seat to *Communications'* Editorial Transformation

For the past five years, we have been co-chairs on *Communications'* editorial board for the Contributed Articles and Review Articles sections. In this short span of time we have

seen a dramatic shift in the kinds of articles submitted to and published in these popular sections of the magazine. This shift has not been by accident.

To appreciate the ever-changing dynamics of *Communications'* readership, we should start at the beginning. ACM was founded in a meeting at Columbia University in 1947 as the Eastern Society for Computing Machinery. It was formed as an outgrowth of the increasing interest in computing that was dawning at that time. ("Eastern" was dropped from the name shortly thereafter).

In the beginning, ACM members were primarily academic researchers interested in the science, development, construction, and application of computing machinery. As we all know, that early interest in computing has since spread explosively to virtually all aspects of human endeavor and has transformed today's world into an intimately connected information society. Indeed, ACM has evolved into "an international scientific and educational organization dedicated to advancing the art, science, engineering, and application of information technology, serving both professional and public interests."

In 2005 David Patterson, then ACM's president, formed a task force co-chaired by Stu Feldman and Mary Jane Irwin to study new directions for *Communications* in the 21st century. The motivation for commissioning this task force was growing dissatis-

faction with what was being published in the magazine at that time.


The broadening of ACM's membership to include researchers and practitioners presented challenges for what kinds of articles to publish in *Communications*. By 2005, ACM's magazine—*ACM Queue*—had a devoted following among many of the practitioners in the field. At the same time, scholars in management information systems viewed *Communications* as the top publication venue for their field. But many computer science researchers felt *Communications* no longer enjoyed the reputation of being the leading scientific publication in the field, a reputation it had enjoyed many decades earlier.

The task force recommended a *Science* magazine-like format for *Communications*. *Science* is a huge success for the American Association for the Advancement of Science. It appeals to scientists across many different fields of science and each issue has a collection of departments ranging from news to perspectives, which invites a wide readership. Most importantly, virtually all scientists consider *Science* among the most, if not *the* most, prestigious publication in all of science.

Communications was refocused to follow the *Science* model. The sections of each issue now include Letters, News, Viewpoints, Practice, Contributed Articles, Review Articles, Research Highlights, and often a Last Byte. A website for the magazine was

also established to not only highlight the print matter but also serve as a news and scientific research tool.

It is still too early to determine whether *Communications* will enjoy the same preeminent scientific reputation in computer science as *Science* enjoys in the biological and physical sciences, but the early indicators are very encouraging. Editor-in-chief Moshe Vardi has made the scientific and technical quality of the entire editorial board his primary concern. All the departments are staffed with preeminent researchers and practitioners who have access to the best reviewers in the field.

As co-chairs of the Contributed Articles/Review Articles sections, we would like to mention the articles we now receive are representative of the expansive reach of information technology. In addition to cherishing scientific excellence in what we publish, we look for original articles that are suggestive of the impact that IT can have on all areas of science and society. Our editorial board also insists on clarity in exposition so the significance of the results being reported can be appreciated by a universal audience. Please continue to submit your best work to *Communications*! 

Alfred Aho and **Georg Gottlob** serve as co-chairs on *Communications'* Editorial Board. Aho (aho@cs.columbia.edu) is the Lawrence Gussman Professor of Computer Science at Columbia University, New York, NY. Gottlob (georg.gottlob@cs.ox.ac.uk) is a professor of informatics at Oxford University, Oxford, England, U.K.

Copyright held by Author(s).



HERE'S TO ADI, RON AND LEN.
FOR GIVING US RSA PUBLIC-KEY
CRYPTOGRAPHY.

We're more than computational theorists, database managers, UX mavens, coders and developers. We're on a mission to solve tomorrow. ACM gives us the resources, the access and the tools to invent the future. Join ACM today and receive 25% off your first year of membership.

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.

[ACM.org/KeepInventing](https://www.acm.org/KeepInventing)



Association for
Computing Machinery

The Internet Governance Ecosystem

OVER THE PAST decade, the Internet and its governance has become the topic of major discussion, debate, and controversy. In infancy, the Internet was a research project sponsored by the U.S. Department of Defense's Advanced Research Projects Agency. It expanded with the participation of the U.S. Department of Energy, NASA and, especially, with the NSF's involvement.

In the mid-1980s, U.S. government Internet research policy rested with the Federal Research Internet Coordinating Committee, which later evolved into the Federal Networking Council, forming a part of the Federal Coordinating Council for Science, Engineering and Technology, which later re-formed into the National Science and Technology Council (NSTC). Within NSTC is the Committee on Technology that has a National Information Technology Research and Development subcommittee managed by the National Coordination Office. A related presidential advisory committee called the President's Information Technology Advisory Committee was formed during the Clinton administration and was later folded into the President's Committee of Advisors on Science and Technology.^a

Many of the R&D and operational functions associated with Internet networks originated as research contracts with U.S. agencies and other governments worldwide. Over time, the various agencies have shed responsibilities for public Internet policy in favor of academic or private-sector organizations, which emerged or were created to fulfill particular functions. The Internet Activities Board was formed in the U.S. to oversee the research leading to the Internet's technical evolution. It became the Internet Architecture Board (IAB) as the Internet Society was created to provide an institutional home and financial support for the IAB and the Internet Engineering and Research Task Forces (IETF and IRTF). Regional Internet Reg-

istries (RIRs) were created to help manage the IP address space: the African Network Information Center, the Asia/Pacific Network Information Center, the American Registry of Internet Numbers, the Latin American and Caribbean Network Information Center, and the Réseaux IP Européens—Network Coordination Center.

In 2003, the World Summit on the Information Society (WSIS) was convened in Geneva. While it was not quite clear what an information society was, many pointed to the Internet as a prototype for its infrastructure. The question then became who was in charge of the Internet; the assembled diplomats had trouble accepting the idea it was highly collaborative and decentralized. They soon focused on the Internet Corporation for Assigned Names and Numbers (ICANN)—created by a White House initiative as a private-sector entity responsible for administering key parts of the Internet's unique space.

Incorporated within ICANN was the Internet Assigned Numbers Authority (IANA) that had been the responsibility of ARPANET/Internet pioneer Jonathan B. Postel^b for years. IANA managed the top-level allocation of Internet address space to RIRs, the so-called root zone of top-level domains of the Domain Name System (DNS), and the registration of key parameters of the Internet protocols specified by the IETF.

The WSIS convened a Working Group on Internet Governance to develop a definition of Internet governance; identify public policy issues; and develop a common understanding of the roles of governments, existing international organizations, and other forums as well as the private sector and civil society from developing and developed countries.

Internet governance was still a focal point at the second WSIS meeting in 2005, where the Internet Governance Forum (IGF) was created for multistakeholders to elaborate further. The first IGF meeting took place in 2006 and the group continues to meet annually.

Today there are multiple efforts to understand the players, roles, and responsibilities manifested in the widespread implementation and operation of the Internet. ICANN President Fadi Chehadé set up four strategic advisory panels to address Identifier Technology Innovation, Multistakeholder Innovation, a Public Responsibility Framework, and ICANN's Role in the Internet Governance Ecosystem. Chehadé also convened a high-level panel to develop principles for Internet governance. Later this month, Netmundial will convene in São Paulo, Brazil.^c The World Economic Forum, together with the U.K.'s Chatham House and Canada's Centre for International Governance, announced a two-year investigation into the way governments use data, including data found on the Internet, for surveillance purposes. Finally, the ITU will convene its quadrennial meeting later this year in South Korea.

The ICANN panel observed that the Internet's ecosystem is vast, diverse, evolving, and dynamic. It recognized that there exists a web of relationships exist among the ecosystem's many players and it would be most valuable to document these relationships. Moreover, it recommended the parties in these relationships consider forming Affirmations of Commitments (AOCs) to recognize one another's responsibilities and make mutual commitments to reinforce them. The panel also urged that means be established to resolve disagreements between the AOC parties. The resulting Web of Commitments would provide an adaptable and evolvable framework for Internet governance. It remains to be seen whether such a vision can be realized but it is fair to say the multistakeholder, cooperative, and collaborative nature of the Internet's development has been a major source of its resilience and its ability to absorb new applications and players since its conception 40 years ago and should form the basis for its future evolution.

^c <http://netmundial.br/> and <http://1net.org/>

Vinton G. Cerf, ACM PRESIDENT

Copyright held by Author.

^a For a summary of the U.S. role in Internet development, see <http://bit.ly/1hJiXRS/>

^b http://en.wikipedia.org/wiki/Jon_Postel

CALL FOR PAPERS

2014 ACM International Conference on Business Analytics

October 8-9, 2014, Houston Texas

<http://bac.acm.org>

CONFERENCE AT A GLANCE

Hosted by its Special Interests Groups (SIG) Board, this is the first ACM-sponsored Conference to focus on *Business Analytics*. Key questions today deal with organizing and managing massive volumes of data effectively, the evolution of analytics techniques and software tools to support complex analytical processes, and how business analytics impacts and changes business organizations and their competitive situations. This conference will address these and related issues.

Conference Board

Dr. Charles K. Davis, Chair
University of St. Thomas

Ms. Kimberly L. Bullock
ExxonMobil Chemical Company

Dr. Jonathan L.S. Byrnes
Massachusetts Institute of Technology

Dr. Wynne Chin
University of Houston

Dr. Charlene A. Dykman
University of St. Thomas

Dr. Gerald D. Everett
IBM (retired)

Dr. Wagner A. Kamakura
Rice University

Dr. Wayne L. Winston
Indiana University



Paper Submissions by: May 30, 2014



Welcoming Submissions On:

- **Theoretical Foundations of Business Analytics.** Theories and Frameworks, Epistemology of Big Data, Design and Functionality
- **Economic Impacts of Business Analytics.** Innovation through Analytics, Revenue & Profit Analytics, Potential of 'Big Data', Impact Assessments, Advertising Analytics, Investment Analytics
- **Functional Area Analytics.** Marketing Analytics, Sales Analytics, Financial Analytics, Human Resources Analytics, Production and Operations Analytics, Oil & Gas Analytics, Healthcare Analytics, Travel Analytics, Sports Analytics, Educational Analytics
- **Business Analytics Roles & Methods.** Best Practices for Business Analytics, Impact of Analytics on IT and non-IT Roles, Advanced Analytics
- **Big Data & Data Warehousing.** IT Operations, Data Mining, Data-Driven Management, Big Data Analytics
- **Business Intelligence & Decision Systems.** Decision Support Systems, Operational Analytics, Stochastic Modeling, Machine Learning, Cognitive Systems, Enterprise Business Intelligence
- **Business Analytics Technologies.** Predictive Analytics, Analytics as a Service, Visualization, Distributed Parallel Architectures, Analytics Engines
- **Data Sources and Data Collection.** Data Acquisition, Web Analytics, Scaling Business Analytics
- **Defining and Controlling Analytics Projects.** Staffing, Analytics Project Management
- **Analytics and Strategy.** Corporate Planning Models, Problem Finding, Planning Analytics
- **The Future of Business Analytics.** Organizational Impacts, Analytics as a Service, Digital Management, Cloud-based Analytics

Collage Photos Clockwise from Top Left: Sam Houston Statue - Hermann Park (by Another Believer CC-BY-SA), Houston Skyline (by Henry Chan CC-BY-SA), The Galleria (by Postoak CC-BY-SA), NASA Mission Control, (Public Domain) Houston Ship Channel – Port of Houston (Public Domain), and Melcher Hall – The University of Houston's Bauer College of Business (by RJN2 CC-BY-SA). The Pumpjack at Sunset photo at the left is Public Domain.

Code That Missed Mars

GERARD J. HOLZMANN'S article "Mars Code" (Feb. 2014) demonstrated a nonblocking implementation of concurrent double-ended queues, or deque¹ (previously shown to be incorrect by Doherty²) to not work through an application of Holzmann's own Spin model checker. However, the demonstration seemed too shallow. Assuming the writer process of the test driver is allowed to *pushRight(0)* and *pushRight(1)* before the reader process gets a chance to run, then the value of *rv* returned by the first succeeding *popRight()* in the reader process would definitely not be 0 and the *assert(rv == i)* would fail because *i* is 0; that is, the test driver is incorrect and could fail, even with a correct implementation of concurrent dequeues.

Holzmann's demonstration included a description of a failing run of his test driver exercising the concurrent deque implementation. Even though Holzmann clearly left out some details—the initialize function and complete output of the model checker—the failing run he described was definitely much simpler than the failure described by Doherty,² indicating the failure detected was in the test driver, not in the concurrent deque implementation.

References

1. Detlefs, D.L. et al. Even better DCAS-based concurrent dequeues. In *Distributed Algorithms, LNCS Vol. 1914*, M. Herlihy, Ed. Springer-Verlag, Heidelberg, Germany, 2000, 59–73.
2. Doherty, S. *Modelling and Verifying Non-blocking Algorithms That Use Dynamically Allocated Memory*. Master's thesis, Victoria University, Wellington, New Zealand, 2004.

Thorkil Naur, Odense, Denmark

Author's Response:

Naur is correct, and I thank him for his keen observation. The test driver used for the example was flawed. If we make the required changes we can show it takes minimally three processes to expose the bug in the original DCAS algorithm,¹ as was also shown in Doherty.² A corrected version of the example, with the model-checking result, is available from the author.

Gerard J. Holzmann, Pasadena, CA

Yes, Teach Everybody to Code

As explored in Esther Shein's news story "Should *Everybody* Learn to Code?" (Feb. 2014), educators should indeed teach everybody to code, even if not all become programmers. After all, throughout the English-speaking world, we aim to teach everybody to write good English, even though we do not expect all of them to write novels for a living.

William Clocksin, Hatfield, U.K.

Learn from Long-Term U.K. MOOC Experience

I keep reading about U.S. initiatives involving massively open online courses, or MOOCs, and computer science education in schools, as in Andrew McGettrick's Letter from the Chair of Education Board "Education, Always" (Feb. 2014) and Tim Bell's Viewpoint "Establishing a Nationwide CS Curriculum in New Zealand High Schools" (Feb. 2014). Here, I would like to point out the U.K. has had a distance-education university—the Open University, founded 1971—that has made ample use of appropriate technology and is well worth looking at if you want to benefit from a long-running, successful, high-quality system; for a condensed history of this so-called "University of the Air," see <http://www.open.ac.uk/about/main/the-ou-explained/history-the-ou>. I would also like to point to England's more recent but equally successful campaign called "Computing At School" to introduce and scale out teaching computer science for all schoolchildren; see <http://www.computingatschool.org.uk/>. All can likewise share quite a bit of useful experience there, too.

Jon Crowcroft, Cambridge, England

Toward Multidisciplinary Design Thinking

In his Viewpoint "Toward a Closer Integration of Law and Computer Science" (Jan. 2014), Christopher S. Yoo raised an important point about how the law and technological change interact but emphasized only one dimension of what

could be called the "social embedding" of technology. Legal concerns are an important aspect of software design, especially if the software stores and processes sensitive personal data about users. However, in order to increase the acceptability and acceptance of a software product, more aspects must be considered during development. Some (such as data privacy and usability) are well represented in most projects. Others (such as users trust in technology, incentives to participate in collective activities, inclusion of users with disabilities, and ethical and sociological challenges) have only begun to attract attention due to recent technological advancements (such as context-aware services, self-adaptive systems, and autonomously acting agents). The crucial point for software developers is these aspects of social embedding could lead to conflicting software design requirements, so should be addressed together in a systematic and integrated development process. Because society demands it, truly multidisciplinary design thinking will become increasingly important in the future.

Kurt Geihs, Kassel, Germany

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.

© 2014 ACM 0001-0782/14/04 \$15.00

Is it Safe to Migrate Servers to the Cloud...Yet?

Reducing the Software Value Gap

The Community Source Approach to Software Development: The Quali Case

Understanding Empirical Hardness of NP Problems

And the latest news on the longevity of information, how the digital era confounds courts, and how computers are changing biology.

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:10.1145/2581795

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

Eyes Forward

Mark Guzdial considers why computing education lags behind other sciences, while Daniel Reed weighs balancing immediate research needs against future uncertainty.



Mark Guzdial
"We May Be 100 Years Behind in Making Computing Education Accessible to All"

<http://cacm.acm.org/blogs/blog-cacm/171475-we-may-be-100-years-behind-in-making-computing-education-accessible-to-all/fulltext>
January 17, 2014

Just how far behind other STEM disciplines are we in computing education? Unlike mathematics and the sciences, we do not have teachers in every school. We do not have a wide range of well-defined, standards-based curricula for elementary and primary levels with supportive materials available to every teacher. In the U.S., there are few pre-service teacher professional development programs available at schools of education, and few states can offer a credential or license to teach that says *computer science teacher*. How long does it take to build up all that stuff?

Here is one way of measuring the gap:

► The National Council of Teachers of Mathematics (NCTM) is the professional organization of mathematics teachers and administrators in the U.S. It was formed in 1920.

► The American Association of Physics Teachers (AAPT) is the professional organization for physics teachers in the U.S. It was formed in 1930.

► The Computer Science Teachers Association (CSTA) was formed in 2005.

The gap may be more than 75 or 85 years, though, because NCTM and AAPT were formed when much of the school infrastructure was already created. There were physics and mathematics teachers in most schools in the U.S. when AAPT and NCTM were formed. There were physics education and mathematics education faculty helping prepare the next generation of teachers already. We in computing education do not have those advantages even today.

It is worth considering what that presence in schools buys those other disciplines. Why would we want to be in school anyway? One of the explicit roles for schools in the U.S. (and probably many other nations) is to ensure equal access. The first item in the mission of the U.S. Department of Education is to "Strengthen the Federal commitment to assuring access to equal educational opportunity for every individual." Thomas Jefferson argued the purpose of public schools was to "diffuse knowl-

edge more generally through the mass of the people." *We want computing education to be in schools to give everyone the opportunity to pursue computer science.*

Computer science education today is mostly accessed by males who are white or Asian. My wife and colleague, Barbara Ericson, has been getting a significant amount of press for her analysis of the results of the AP CS exam in 2013. There were three states in which no females took the exam. *USA Today* had a stunning visualization of just how more gender-skewed CS is compared to all other AP exams. In 11 states, no black students took the exam. In eight states, no Hispanic students took the exam. *Yahoo News* had an amusing piece where they pointed out that kids of Wyoming might be underrepresented in CS—not a single student took the exam in that state. CS is not fully accessible in schools yet.

When I talk to my colleague computer scientists about how far behind we are, they sometimes suggest the way to close the gap is to "create an alternative system." They suggest we build a computing education system built around "city recreation departments, after-school programs," and programs like Khan Academy and MOOCs. Certainly, building an alternative education program, especially one wholly or significantly online, is an important and interesting research endeavor. But rebuilding an *entire* education system as a *shortcut* to getting *one subject* into schools seems unlikely to be easier or shorter.

Chris Stephenson pointed out in her blog post (<http://bit.ly/1nuCjca>)

that university CS faculty do not understand what schools do and how they do it. What is worse, I fear computer scientists do not understand the *necessity* of traditional face-to-face schools for providing access. Students who know about computing and have the resources to succeed at it are frequently privileged (see Philip Guo's excellent post on what technical privilege means, at <http://bit.ly/1bNlUua>). As seen in the recent University of Pennsylvania study of MOOCs, students who succeed at MOOCs tend to already be well educated. Most people, especially underprivileged students, need a teacher today. Being able to *reach* an online education resource is not the same as access if you need a teacher and the infrastructure of school to help you *understand*.

Someday, we might provide students online supports that are equivalent to an in-person teacher. That is a research goal. Today, we do not know how to make a new school framework that still achieves the goal of giving everyone access. Today, if we want more people to have access to computer science education, we need face-to-face schools and in-person teachers. We need to work within the existing school framework to make that happen. And we have a long way to go.



Daniel Reed
"Deferred Maintenance on the Future"

<http://cacm.acm.org/blogs/blog-cacm/171143-deferred-maintenance-on-the-future/fulltext>

January 6, 2014

In straitened financial times, time horizons shrink. This observation is self-similar across scales, applying equally to individuals and families, small businesses and corporations, and countries and economic blocs. If you find yourself struggling to pay bills, even after eliminating luxuries, then you defer some purchases, often painfully. Indeed, if you are homeless, cold, and hungry, physical needs shrink time to the here and now—the next meal and a warm place to sleep trump all else. That is something worth remembering about those less fortunate as we face a near-record cold across much of the continental U.S.

When times are difficult, as an individual, you keep driving that aging car or truck, even as its reliability declines and the risks of major failure increase. As a small business owner, you defer that infrastructure upgrade, making do with what you have. As a CEO, you avoid risks, focusing on expense reduction and weathering the financial storm. As a country, you collectively focus on the short term, avoiding or militating the effects of recession, prioritizing short-term expenditures over long-term investments.

These sacrifices are natural and rational—in the short term. If continued too long, however, they ultimately lead to calamity and loss, as individuals suffer, infrastructure fails, and the future becomes shrouded in a miasma of unfulfilled dreams. For all our physical wants and needs, we are creatures of dreams.

Make no mistake; balancing the immediate, pressing, and real needs of the here and now against the uncertain and ill-defined future is a difficult task, made no easier by a cacophony of competing petitioners, each with compelling arguments and considerable needs. Yet it is precisely such a time when wisdom and foresight are required; it is the very definition of leadership. Although the needs of the present are real, dreams of the future must not be sacrificed on the altar of exigency.

Telling the Future the Past

Today in the U.S. we face difficult challenges, with a growing backlog of deferred intellectual maintenance. Overworked and sleep-deprived drivers are steering many of our vehicles of discovery on balding tires across potholed roads. Stripping away the metaphor, we are struggling to sustain appropriate investments in basic research infrastructure and facilities operations, and our dispirited researchers face ever-diminishing odds of research funding as they work to keep laboratories operational and students and post-doctoral research associates funded.

It is worth remembering that the Computer Science and Telecommunications Board (CSTB) of the U.S. National Academies released and updated the famous "tire tracks" diagram (<http://bit.ly/1bYhSlK>) illustrating the path from basic computing research ideas to major industries. In almost

every case, the time from discovery to major societal impact was a decade or more, yet few could have imagined that impact at the time of discovery.

Today's ubiquitous smartphone or tablet has its roots in Engelbart's 1968 "mother of all demos" (<http://bit.ly/1lEdNGe>), and a host of other advances in microprocessors, memory and storage systems, Web services, and wired and wireless broadband communications. The same is true of cloud computing, advanced robotics, streaming multimedia, global positioning systems, and supercomputing. Each capability is the evolving outgrowth of decades of basic and applied research by tens of thousands of dedicated and passionate researchers. Their dreams of what might be became the computing and communications infrastructure that underpins today's society.

The impact of basic research is no less profound in a host of other domains, and today's choices have long-term implications for national and international competitiveness. As history has repeatedly shown, investment in the future—basic research—is integral to economic recovery and long-term growth. Yet by its very definition, the intellectual and pragmatic outcomes of specific research projects and directions are unpredictable. It is only in retrospect that we see the clear and unmistakable benefits—in medicine and public health, in design and manufacturing, in energy production and efficiency, and yes, in computing and communications.

The Road Ahead

The past speaks urgently to the present about the future. It whispers about what could be, about dreams deferred and opportunities lost, about innovation and economic success, and about creativity.

It is time to put some new tires on the vehicle of scientific discovery and head out to the future. As Kerouac noted in *On the Road*, there's "Nothing behind me, everything ahead of me, as is ever so on the road." We do not know what we will find, but the journey itself is the destination. It leads to the future and a better world. ■

Mark Guzdial is a professor at the Georgia Institute of Technology. Daniel Reed is Vice President for Research and Economic Development at the University of Iowa.

© 2014 ACM 0001-0782/14/04 \$15.00



CHI PLAY 2014

The ACM SIGCHI Annual Symposium on
Computer-Human Interaction in Play

Important Dates

8 May 2014, 5:00pm PT

Full papers, demos, workshops, doctoral consortium

26 June 2014, 5:00pm PT

Student competition, courses, panels and works-in-progress

Organizing Committee

Conference Chair

Lennart Nacke, University of Ontario
Institute of Technology, Canada

Technical Program Chair

Nicholas Graham, Queen's University,
Canada

Papers and Proceedings Chairs

Florian "Floyd" Mueller, RMIT University,
Australia
Regan Mandryk, University of Saskatche-
wan, Canada

Works-In-Progress, Videos and Demos Chairs

Peta Wyeth, Queensland University of
Technology, Australia
Paul Cairns, York University, UK

Student Game Design Competition Chairs

Vero vanden Abeele, University of Leuven,
Belgium
Bieke Zaman, University of Leuven,
Belgium

Industry Case Studies and Panels Chairs

Anders Drachen, Game Analytics, Denmark
Ben Medler, Electronic Arts, USA

Courses and Tutorials Chairs

Regina Bernhaupt, IRIT, University Paul
Sabatier, Toulouse III, France
Bill Kapralos, University of Ontario Institute
of Technology, Canada

Workshops Chairs

Zach Touns, New Mexico State University,
USA
Georgios Christou, European University
Cyprus, Cyprus

Doctoral Consortium Chairs

Drew Davidson, Carnegie Mellon Universi-
ty, USA
Eelke Folmer, University of Nevada, Reno,
USA

Local Arrangements Chairs

Pejman Mirza-Babaei, University of Ontario
Institute of Technology, Canada
Andrew Hogue, University of Ontario
Institute of Technology, Canada

Key Topics Include

- Game Interaction
- Novel Game Control
- Games User Research
- Gamification
- Persuasive Games
- Games for Health
- Games for Learning
- Player Experience
- Game Evaluation Methods
- Social Game Experiences
- Serious Games
- Tools for Game Creation
- Developer Experiences
- Industry Case Studies

www.chiplay.org

Using Patient Data for Personalized Cancer Treatments

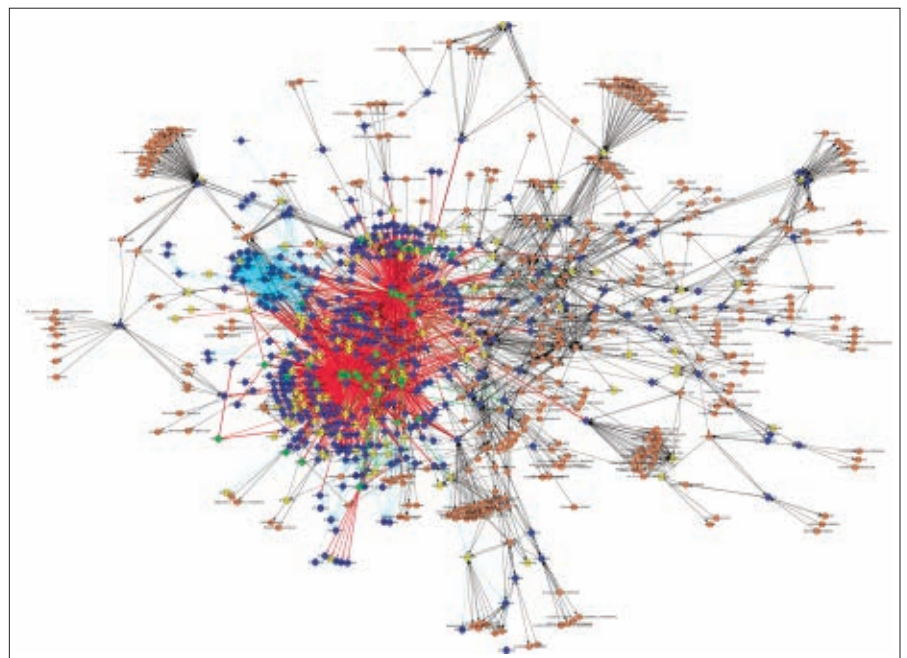
Patient information databases eventually will help improve health outcomes and support development of new therapies.

HEALTH ORGANIZATIONS IN Europe and the U.S. are pushing ahead with plans to compile massive databases of patient information, which they anticipate will not just improve treatments and survival chances for cancer sufferers, but will drive the development of new therapies. These moves will involve many more participants in massive clinical trials than today, and will allow medical centers in the developed world to expand their reach globally.

One organization, the American Society of Clinical Oncology (ASCO), aims to have ready by 2015 the production version of its CancerLinQ ‘learning health system,’ building on the experience of a 170,000-record prototype unveiled in 2013. At a White House event in November 2013, Thomas Kalil, deputy director for technology and innovation at the White House Office of Science and Technology Policy, identified CancerLinQ as one of a number of healthcare systems that would aggregate large quantities of data to improve treatment.

“There are huge opportunities to both improve health outcomes and lower costs,” Kalil said, claiming CancerLinQ would allow every patient’s experience to help inform future can-

cer care. “Currently only 3% of patients participate in clinical trials. ASCO is committed to figuring out how to use the data, while protecting patient confidentiality of the other 97% to make



In 2011, researchers at Columbia University Medical Center built this model of the gene regulation network in mammalian cells that they used for studies into the genetic variability of cancer.

our healthcare system much more of a learning system, to improve the quality of care and accelerate the development of new therapeutics.”

Dr. Peter Campbell, head of cancer genetics and genomics at the Wellcome Trust Sanger Institute, said at the Oncology Forum 2013 held in Amsterdam, The Netherlands, last autumn that aggregating data into large, online databases would help drive the advent of personalized medicine, providing an opportunity to treat serious illnesses more effectively. “We are standing on the cusp of an era where we can characterize all patients and what drives their cancer.”

Cancer is the primary target, partly because of its prevalence. According to the American Cancer Society, cancer remains the second most common cause of death in the U.S., accounting for almost one in four deaths. The European situation is similar; Tonio Borg, European Union commissioner for health, says, “We expect that in the EU, one in three men and one in four women will be affected by cancer before reaching the age of 75. Cancer is not something that only affects others; it happens to everybody.”

The other reason for using population-scale databases to collate and process patient information is due to cancer’s nature. Campbell says cancers have huge variations that result from the many different ways in which the DNA of tumor cells can mutate in different patients. In breast cancers, for example, analysis of tumors reveals the most commonly mutated or deleted genes were

found in just 10% of affected patients. “There is a long tail of other mutations, each affecting just a few percent or less of patients,” Campbell explains.

With diseases such as cancer, genetic changes cause some biological feedback loops and other processes to break down in unpredictable ways. Researchers have found that even within the same tumor, cells may be altered in different ways, so a therapy that works for a large group of patients may be utterly ineffective for the one sitting in the doctor’s office.

The dream of personalized or stratified medicine, says Campbell, is to prescribe treatments for a patient’s specific condition and “melt away the cancer.” The problem is obtaining enough data to work out how different treatments fare under different conditions.

“I would say that interventional clinical trials are underpowered to detect gene-drug interactions. If a particular mutation is only found in one in 700 patients, you need to screen that many just to find one participant for a clinical trial for a drug that targets it,” Campbell explains. “We do not have enough patients to study. We need several thousand patients, maybe tens of thousands of patients. For any given tumor type, we need a database of 10,000 to 20,000 patients, and with 50 to 100 common tumor types, that means access to the records of at least one million patients.”

Dr. Sandra Swain, former ASCO president and medical director of the Washington Cancer Institute at Med-Star Washington Hospital Center, says

significant groups of patients are not represented well in the existing clinical-trial structure. Seniors, for example, “have diseases associated with getting older; that changes how we need to evaluate the patient in front of us.”

Another issue is how medical practitioners can sift through data available in principle, but inaccessible in practice.

To improve physicians’ access to relevant medical data, the University of Texas MD Anderson Cancer Center is using an IBM supercomputer running artificial intelligence software originally written to allow such a machine to compete on the “Jeopardy!” game show. IBM’s Watson supercomputer uses statistical natural-language processing techniques to work out which pieces of information are relevant to a problem. The Oncology Expert Adviser software is being used to analyze patient data, as well as research data from cancer trials at the Center.

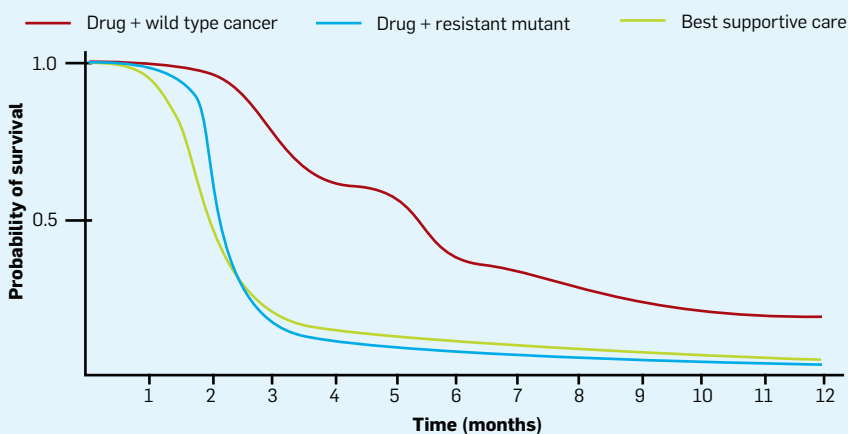
Dr. Hagop Kantarjian, chair and professor in leukemia at MD Anderson, says the center’s Watson-based system “will be a transformational tool. Today when we see patients with cancer, we rely on our memory and our limited amount of knowledge to work out the next best step.” By presenting information from a wide range of sources, Kantarjian says, the Oncology Expert Adviser could be “a quantum leap in the way we treat patients and how we provide care and knowledge.”

For its CancerLinQ prototype, ASCO used an approach based on a combination of open-source software and the Galileo Analytics’ Galileo Cosmos data-mining software, which accessed and analyzed the preprocessed data. Preprocessing software uses statistical functions and an artificial neural network to learn, structure, and map data fields on the original records to the format the system needs.

Dr. Clifford Hudis, president of ASCO and chief of the breast cancer medicine service at New York’s Memorial Sloan-Kettering Cancer Center, said CancerLinQ lets physicians see the results of interventions on other patients who fit a given profile, to help them determine the most appropriate course of action. “The clinician isn’t just a robot doing what the computer says,” he stresses.

Kantarjian says of the Oncology Expert Adviser, “This kind of data gathering and analytical tool will allow discov-

The Kaplan-Meier plot of survival chances versus time, commonly used to gauge treatment efficacy for patients, shows how a cancer mutating from a frequently encountered form may develop increasing resistance to standard drug treatment.



eries that we cannot do today. Suppose a group of patients take a medicine that is not for cancer but is, say, a heart medicine, and that medicine also influences the sensitivity of their particular tumor to the effect of the cancer treatments. We are going to discover through these analytical tools very quickly that this helps the patients, and we can apply that to future treatment programs.”

Dr. Emile Voest, chair of the Center for Personalized Cancer Treatment at the medical research institute UMC Utrecht in The Netherlands, which has its own data analysis system, warns of putting too much faith in findings extracted from large-scale databases to prescribe customized treatments “off-label” without additional research.

“Some people say 75% of patients can now be treated with something because of the genetic information we now have. I tend to disagree because, frankly, we do not know; we can only postulate. It may work, but I am very much against treating patients off-label outside of clinical studies because we are going to make a mess of it,” says Voest.

An earlier attempt to use genetic data for treatment selection failed. Clinical trials started in 2007 after researchers at Duke University published results suggesting tests using arrays of chemical DNA probes could help select viable treatments. Later research found those results could not be reproduced, and a group of 30 bioinformaticians and statisticians urged the National Cancer Institute (NCI) to suspend those trials, which it did.

If not used directly to create customized off-label treatments in the doctor’s office, the mined data is likely to be utilized to inform basic medical research, particularly in the emerging field of systems biology, which uses computer modeling to predict biological behavior.

Says Andrea Califano, chair of the department of systems biology at Columbia University, “Instead of using statistical associations, we are starting to go towards a model-driven type of science where we create regulatory models of cells.”

Systems biology regards biological cells as a form of signaling network, a dynamic system in which proteins and genes interact in complex feedback loops. A number of cancer scientists believe combination therapies may

Even within the same tumor, cells may be altered in different ways, so a therapy that works for many patients may be utterly ineffective for the one sitting in the doctor’s office.

provide the key to dealing with cancers currently difficult to treat directly.

The complex network inside a cell can provide a way to attack cancers more effectively. A single drug may target only one part of the network, perhaps inactivating a single protein, but that is only effective if the cancerous cell is disabled by that step. In many cases, changes made to the network by knocking a protein out of action will uncover new signaling pathways that help the cell to survive. Computer modeling can help identify the most productive targets, and combinations of them, to increase the probability of the treatment’s success.

Campbell argues, “Not only are many genes involved; many are not easily actionable. Many of them are essential: you cannot drug them without toxicity. For example, kidney cancer is almost entirely driven by tumor-suppressor genes, and those are notoriously hard to drug.

“A further issue is that many of these genes interact with each other,” Campbell adds. “You can see patterns of genes that are co-mutated with each other. When pairs of genes are significantly mutated, they often interact strongly; that is clearly important and will probably play out in most cancers. These secondary mutations will probably have a strong impact on the patient’s treatment.”

Although the aim of large-scale medical learning systems is to have automated data collection and parsing, the reality is more complex. Dr. Peter Johnson, chief clinician at Cancer Research U.K., says full automation cannot be guaranteed, based on his experience with the UK pilot scheme. “What we are

aiming to do is get automated data collection and extraction, but the complexity is such that a lot of the collection and extraction is manual,” says Johnson.

As CancerLinQ moves closer to production, ASCO expects some manual processing will be necessary, at least in early versions of the system. Says Dr. Robert Hauser, senior director of ASCO’s quality department, “Oncology is very complicated, and therefore will require a lot of manual interpretation and intervention in the beginning. We believe over time that manual intervention efforts will decrease as the system learns.”

As these systems grow to incorporate millions of patient records, organizations such as MD Anderson see them as a way to extend their reach globally. MD Anderson president Dr. Ronald DePinho points to a recent Institute of Medicine report that catalogued large disparities in cancer care across the U.S. Outside the U.S., DePinho says, “We have medical deserts on a global scale.

“Our mission is to end cancer, here and around the world. Our platform allows us to democratize MD Anderson care. We can go where the patients are, to places where a doctor hasn’t read a paper for 10 years.”

Collecting and processing the data is the beginning of a process that could link cancer treatment around the world. □

Further Reading

Basso, K., Margolin, A.A., Stolovitzky, G., Klein, U., Dalla-Favera, R., Califano, A.

Reverse engineering of regulatory networks in human B cells, *Nature Genetics* 37, 382-390 (2005)

Gonzalez-Angulo, A.M., Hennessy, B.T.J., Mills, G.B. Future of Personalized Medicine in Oncology: A Systems Biology Approach *Journal of Clinical Oncology*, vol 28, no. 16, June 1 (2010)

Tsimberidou, A.M., Ringborg, U., Schilsky, R.L. Strategies to overcome clinical, regulatory, and financial challenges in the implementation of personalized medicine 2013 ASCO Educational Book <http://meetinglibrary.asco.org/EdBookTracks/2013%20ASCO%20Annual%20Meeting> [Open Access]

Godman, B., et al.

Personalizing health care: feasibility and future implications *BMC Medicine*, 2013, 11:179 <http://www.biomedcentral.com/1741-7015/11/179> [Open Access]

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

© 2014 ACM 0001-0782/14/04 \$15.00

Speech-To-Speech Translations Stutter, But Researchers See Mellifluous Future

The practical need for accurate instant or simultaneous machine translations continues to grow as applications multiply.

WHILE COMPUTER SCIENTISTS have yet to build a working “universal translator” such as the one first described in the 1945 science-fiction novella “First Contact” and later employed by the crew of the Starship Enterprise on “Star Trek,” the hurdles to creating one are being cleared. That is because the practical need for instant or simultaneous speech-to-speech translation is increasingly important in a number of applications.

Take, for example, the hypergrowth of social networking and Skype chats that demand bidirectional, reliable, immediate translations. Similarly, when natural disasters strike, the problem of aid workers struggling to communicate with the stricken who often speak other languages can become overwhelming.

When tourists travel to remote areas; when businesses want to speed commerce; when doctors who speak only one language need to talk with patients who only speak another; when immigration control is unable to conduct interviews because of a language barrier; all of these are good examples of why instant or simultaneous speech-to-speech translation has become more than just a good idea.

Indeed, in the European Union, where there are 24 official languages (up from 11 in 2004) and where the European Commission employs approximately 3,000 staff translators and interpreters, the cost of maintaining the EU’s policy of multilingualism was 1.1 billion euros (\$1.48 billion) last year.

“Simultaneous interpreting by machine, once further developed, could facilitate communication in certain



Carnegie Mellon computer science professor Alexander Waibel (far right), one of the developers of speech translation app Jibbigo, looks on as a U.S. Marine uses the app to communicate with a Thai native.

contexts of simple communication where a professional human interpreter is not available,” says Susanne Altenberg, head of the Unit for Multilingualism Support at the European Parliament in Brussels. “Some aspects of speech-to-speech translation could also assist human interpreters in their work.”

However, the difficulty with machine interpretation that goes out live, immediately, and ensures rapid oral communication, says Altenberg, is that such revision is not possible as it is with translated texts, and current methods are not yet sufficiently reliable.

While Altenberg says it is difficult to predict how long it will take to develop a fully reliable speech-to-speech translation capability, “we follow developments with interest. Simultaneous interpreting is a very complex task involving almost all human cognitive and emotional capabilities. It is quite a challenge for computers to imitate this, especially with 552 language combinations [in the EU] and in a highly political context such as ours.”

The single highest hurdle to a fully reliable speech-to-speech translation technology is that language is inherently ambiguous, says Jaime Carbonell,

director of the Language Technologies Institute at Carnegie Mellon University's School of Computer Science. For instance, in English, the word "line" can mean a geometrical line, a queue where people stand in front of one another, a rope (as in "tangled up in the lines"), a railroad line, an actor's line in a script, and so on. We may not think of "line" as having multiple meanings but, for translation purposes, there are actually 16 of them.

"Traditionally, the most difficult problem—and it remains so—is that you need to use context to pick the right meaning, to resolve the ambiguity so that the correct words and phrases are chosen in the target language," Carbonell explains.

Yet pinpointing which solution is currently the 'state of the art' depends on whom you ask.

At IBM's Thomas J. Watson Research Center, Salim Roukos, senior manager of Multilingual NLP Technologies and CTO Translation Technologies, describes the speech-to-speech translation system on which he and his team are working as a budding technology consisting of three parts: a speech recognition system that converts the audio of the speaker's source language into written text; a text-to-text system to translate the text into the target language, and a text-to-speech system to synthesize audio from the written target language. All three components need to behave really well, individually and together, to achieve a good result.

However, there have always been two major difficulties with such systems, says Roukos.

The first involves speech recognition. "When I don't articulate, when I speak quickly or drop words, if I have an accent, that makes speech recognition harder and the machine does not do as well," he explains. "The error rate can increase from a few percent to tens of percents."

Yet that is no longer the greatest challenge, he says. In a recent evaluation of his team's work, Roukos says speech recognition has improved by about 40%, compared to a year ago, in terms of the reduction of word errors. "We still have a ways to go," he says, "but we have gotten to that point in our work, which is based on what we

"When I don't articulate, when I speak quickly or drop words, if I have an accent, that makes speech recognition harder and the machine does not do as well."

call convolutional neural networks, where speech recognition is not the toughest problem."

The more recent challenge is tackling out-of-vocabulary (OOV) words. Because languages have dialects and speakers frequently use slang, a system's translation from the source language to the target language is often inaccurate. That is why Roukos and his team introduced the concept of a dialogue manager on both sides of the conversation. For example, if the speaker says a word the dialogue manager does not recognize, it will play to the speaker the audio that corresponds to that word, and then ask whether the word is a name; if it is not a name, it will request a synonym, a paraphrase or, in extreme cases, a spelling of the word.

The most recent metrics show that, about 80% of the time, the system is able to detect when it fails to recognize the input and then interacts with the user. Roukos expects to be able to improve that to 90%–95% in the next few years.

"What we are doing now is using machine-mediated speech-to-speech translation, in which the machine is taking an active role in helping the communication across the two languages," says Roukos. "This is brand-new, the state of the art, but a work in progress."

Carnegie Mellon's Carbonell identifies the commercialization of speech-to-speech translation as one of the most recent developments in the field. For instance, in August 2013, Face-

ACM Member News

ZYDA LEADS USC CS GAMES PROGRAM TO THE TOP



"I make change and I make my next position for the job that needs me," said Michael Zyda, founding

director of the University of Southern California's GamePipe Laboratory, and a professor of engineering practice in USC's department of computer science.

In 2004, at age 50, Zyda, an ACM Distinguished Speaker, opted to reinvent himself by leaving his posts as computer science professor and founding director of the Modeling, Virtual Environments and Simulation (MOVES) Institute at the Naval Postgraduate School in Monterey, CA, where he served as principal investigator and development director of the "America's Army" PC game. He went to USC to launch the Joint Games Program, now called USC Games; within five years, he had made it the world's top computer science games program.

USC Games now has 80 engineers (including 30 interactive game designers, as well as 150 artists from outside art schools) who build approximately seven games a year. At the end of every fall and spring semester, it hosts USC Games Demo Day, attracting hundreds of top gaming industry professionals.

Zyda and his students designed *Black Ops 1* and *Black Ops 2*; *Grand Theft Auto 5* ("it made \$1 billion in three days," he recalls); *Modern Warfare 3*, and *Farmville*. "You don't make a Triple A game title in America without a USC Games alumnus," Zyda said, adding, "over 90% of our students have jobs before graduation."

Zyda is a triple-threat career as a computer science professor, game designer (he co-holds the patent on the Nintendo Wii U console's nine-axis sensor), and as an expert witness in gaming industry patent litigation cases.

What does he do for fun? "I swim 2,000 meters freestyle in the USC pool every day for 40 minutes, about 27 miles a month," Zyda laughed.

—Laura DiDio



Association for
Computing Machinery

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

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

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

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



book acquired the team and technology of Pittsburgh-based Mobile Technologies, a speech recognition and machine translation startup that had spun off from Carnegie Mellon and which developed the app Jibbiggo. The app allows users to select from more than 25 languages, record a voice in that language, then have a translation displayed on screen and read aloud in a language of your choosing.

“Speaking of commercialization, Google, with its Google Translate, does an excellent job of using context to determine which is the speaker’s intended meaning,” says Carbonell. “That is because Google has access to so much more data than do others, so it can get better statistics to determine, in the context of a group of words or phrases, what is the most likely meaning so that it can translate accordingly. That is Google Translate’s strength.”

Indeed, the Google Translate Android app—which allows users to speak into a phone, translates the words spoken into a different language, then allows a second user to respond in their own language—began 10 years ago as a third-party software product that translated just eight major languages. Today, the app can translate 72 different languages—from Afrikaans to Yiddish—and processes over a billion translations daily. That allows Google to gather enough data to create dictionaries automatically by learning from that data.

Meanwhile, Google is researching new ways to resolve the translation problem so it might learn more effectively.

“What we have done is to simplify the translation method,” says Franz Och, who heads up the Google Translate team. “Instead of the typical method of feeding whole translated documents into the learning process, we are able to seed our vector system with just a little bit of information—about 5,000 words from Google Translate for which translations are known—and the system can then find parallels between words in different languages in documents that have not yet been translated. Basically, we have made the move from parallel texts, which is what we call texts for which we have specific translation, to just comparable text ... and then use that

“We are closer than ever to translation that is so quick and natural it feels like human translation.”

to learn translation information.”

It is important to note, says Och, that the vector system is not a translation system in itself. “It is just interesting in that it can find these parallels without our feeding it documents that have already been translated,” he adds. “So there is some potential here for this vector approach to help with our existing machine translation system.”

During the past three years, researchers at Microsoft Research (MSR) report having dramatically improved the potential of real-time, speaker-independent, automatic speech recognition by using deep neural networks (DNNs).

At Interspeech 2011, MSR demonstrated how to apply DNNs to large-vocabulary speech recognition and reduce the word error rate for speech by over 30% compared to previous methods, recalls Frank Seide, principal researcher and research manager at MSR’s Speech Group. “This means that rather than having one word in four or five incorrect, now the error rate is one word in seven or eight,” he says. “While still far from perfect, this is the most dramatic change in accuracy since 1979 and, as we add more data to the training, we believe we will get even better results.”

Seide’s team also found using DNNs helps recognition engines deal with differences in voices and accents and the conditions under which the speech was captured, like microphone type and background noise.

“We also determined that our work in DNNs can learn across languages,” Seide adds. “In other words, example data of one language can help to improve accuracy for another language.

This is very important since speech recognizers, as part of their ‘training,’ must be exposed to extremely large amounts of example speech data—on the order of thousands of hours of speech—which has to be painstakingly transcribed down to the last stutter. These significant improvements have aided in the progression of new recognition scenarios such as the possibility of broad-scale speech-to-speech translation.”

Microsoft recently applied MSR’s DNN technology to the company’s Bing Voice Search app for Windows Phone, providing a 12% improvement in word error rate overall compared to the previous Bing system.

Going forward, what are the next steps for researchers working on speech-to-speech translation?

At MSR, the goal is to improve recognition and translation of language as people use it. “People don’t speak in the same way they write,” says Chris Quirk, senior researcher at MSR’s Natural Language Processing Group. “They do not even write like they used to; look at social media sites such as Facebook and Twitter. Being the translation service for Facebook has given us a unique view on this rapid change, driving us to broaden our systems toward the language of today and tomorrow. Clearly, we have to find new and different data sources.”

At IBM, Roukos’ team will concentrate on improving their system’s ability to detect “low-confidence regions” where the system does not rec-

ognize certain words and/or phrases, does not know how to translate them, or is not sure it translated them correctly. “The ability of the system to detect when it does not know what the input is, and therefore is interacting with the user to clarify or paraphrase, is our core focus.”

Meanwhile, at Google, Och is reluctant to predict how long it will be before a close-to-foolproof method exists for simultaneous or instant language translation. “If you asked that of people in AI research any time since the 1950s, the answer would be ‘in about five years,’ so that is what I will say is, in about five years. But it is true we really are closer than ever to translation that is so quick and natural it feels like real human translation.

“For some language pairs—like Spanish to English—we are already pretty close; people judge our translations to be sometimes as good as, or better than, a human translation. Of course, that is partly because human translations are not always that great. Ideally, machine translations would be even more consistently high-quality; at least, that is our goal.”

Further Reading

“Simultaneous Translation By Machine,” a video posted June, 2012 by KITinformatik at <http://www.youtube.com/watch?v=q2amqJmmDm4>

“Exploiting Similarities Among Languages For Machine Translation,” Tomas Mikolov, Quoc V. Le, and Ilya Sutskever, September 2013, the Cornell University Library, <http://arxiv.org/abs/1309.4168>

“Learning The Meaning Behind Words,” a blog by Tomas Mikolov, Ilya Sutskever, and Quoc Le, published August 2013 by Google, at <http://google-opensource.blogspot.com/2013/08/learning-meaning-behind-words.html>

“Breaking Down The Language Barrier – Six Years In,” a blog by Franz Och, published April, 2012 by Google, at <http://googleblog.blogspot.com/2012/04/breaking-down-language-barriersix-years.html>

E. Matusov, G. Leusch, R. E. Banchs, N. Bertoldi, D. Déchelotte, M. Federico, M. Kolss, Y. S. Lee, J. B. Mariño, M. Paulik, et al.,

“System Combination For Machine Translation Of Spoken And Written Language,” September 2008, IEEE, http://ieeexplore.ieee.org/xpl/login.jsp?tp=&ar_number=4599393&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4599393

“Speech Recognition Breakthrough for the Spoken, Translated Word,” a video posted Nov. 7, 2012 by Microsoft Research, at http://www.youtube.com/watch?feature=player_embedded&v=Nu-nlQqFCKg

“Speech Recognition Leaps Forward,” an article by Janie Chang, published August, 2011 by Microsoft Research, at <http://research.microsoft.com/en-us/news/features/speechrecognition-082911.aspx>

Google Translate, an app developed by Google, updated Nov., 2013, at <https://play.google.com/store/apps/details?id=com.google.android.apps.translate>

Bing Translator, an app developed by Microsoft Research, updated Dec., 2013, at <http://www.windowsphone.com/en-us/store/app/translator/2cb7cda1-17d8-df11-a844-00237de2db9e>

Paul Hyman is a science and technology writer based in Great Neck, NY.

© 2014 ACM 0001-0782/14/04 \$15.00

Education

ACM Report Urges Expansion of CS Education

A new report from ACM found few states positioned to provide students with the computer science (CS) education required for rewarding careers and to ensure the needs of the future workforce are met.

The report, *Rebooting the Pathway to Success: Preparing Students for Computing Workforce Needs in the United States*, urges state education and business leaders and public policy officials to work together to develop

comprehensive CS education and workforce development plans. The report provides recommendations to help these leaders create pathways that will expose all K–12 students to computer science, provide expanded access to more rigorous CS courses, offer increased opportunities for students to pursue post-secondary degrees, and align education pathways with computing careers.

“By 2020, one of every two jobs in science, technology,

engineering, and mathematics (STEM) will be in computing,” said Bobby Schnabel, chair of ACM’s Education Policy Committee. “This concentration of computing positions in STEM makes it imperative for K–12 students in academic and career technical education programs to gain more opportunities to learn computer science.”

The report calls on colleges and universities to play a role

in expanding opportunities for computer science education by recognizing rigorous computer science courses in their admissions processes. Higher education institutions also can reduce barriers to degree completion by adopting systemwide agreements that allow students to transfer course credits to fulfill their computing degrees efficiently.

The full report is available at pathways.acm.org.

New Models in Cosmetics Replacing Animal Testing

A European law spurs scientists to develop computational simulations capable of predicting the toxicity of cosmetics.

LAST YEAR, A long-planned European Union ban on using animals to test the safety of cosmetic ingredients went into effect. The motivation for the new rule stems in part from concerns about the ethics of testing the safety of chemicals on living creatures, but scientists say there is more at play here than a moral conundrum.

Reliance on animal testing, according to several experts, has actually hindered the evaluation of many chemicals and ingredients inside and outside the cosmetics industry. Animal-based tests take too long and are too expensive, they say, often requiring several years and millions of dollars or more to carry out.

The significant physiological differences between humans and the mice, rats, and other animals used to evaluate the safety of chemicals also can limit the validity of the results. “We need a better way to test toxic chemicals,” says toxicologist Kristie Sullivan, director of regulatory testing issues at the non-profit Physicians Committee for Responsible Medicine.

In 2009, industry trade group Cosmetics Europe, along with a division of the European Commission on Research & Innovation, launched a \$68-million research initiative to develop lab technologies and computational models capable of predicting the toxicity of chemicals in humans. Although cosmetics companies are anxiously awaiting these alternatives, the so-called Safety Evaluation Ultimately Replacing Animal Testing (SEURAT) program was not meant to produce a quick solution. The scientists involved with SEURAT, which is made up of five individual research clusters and involves more than 70 universities, research groups, and biotechnology companies, say these new technologies will take at least five more years to develop.



A test subject? Not in Europe, which banned the use of animal testing in the development of cosmetic ingredients last year.

As the researchers push the boundaries of their fields to build these new tests, one thing is clear. “The future direction clearly has to involve computation,” says Elmar Heinzle, a chemical engineer at the University of Saarland in Germany and a leader of NOTOX, one of the SEURAT research clusters.

Mechanisms of Action

Pharmaceutical companies and regulatory agencies have been using computational tools to evaluate toxicity for years. Generally, these methods look at the structure of the chemical compound or molecule under consideration, run it up against a database of chemicals with known toxicological effects, and search for substances with similar chemical structures. If a new chemical or ingredient lines up with one that has proven toxic in previous tests, this suggests the new compound could be similarly troublesome.

Instead of relying on structural similarities, NOTOX technologies will base predictions on actual biological mechanisms of action. Heinzle says scientists have reached a consensus that there are similarities in terms of how toxic chemicals act within the body. “Although the whole systems are very, very complex and the number of possible interactions is huge,” he says, “we have found that the number of pathways that leads to adverse outcomes is very limited.”

The NOTOX plan is to analyze these pathways in the lab, picking out key events such as a molecule binding to a particular protein, and then using this data to bolster virtual or *in silico* models. A combination of laboratory and computer modeling would then be used to evaluate whether a new chemical would spark reactions that match those critical, harm-inducing events. Heinzle acknowledges this approach leaves open the possibility of a molecule acting via

an unknown mechanism, but he adds the same holds true for animal testing, and that it would be difficult to eliminate all risk. The combination of powerful models and robust lab tests, though, will reduce that risk to safe levels.

The scientists say the time is right for the new approach, in part because of more advanced electron microscopes and other imaging technologies that allow scientists to capture the inner architecture of organs and their cells. Additionally, *in vitro* or cell-culture-based studies of toxicity have vastly improved. In the past, scientists often relied on short-lived, two-dimensional cultures of human liver cells to test for possible toxic effects; now scientists not only have more accurate 3D cell cultures at their disposal, but these cultures also last longer, allowing lab researchers to study the effects of a substance over a longer period of time.

Virtual Body

Cosmetics present a unique challenge for the researchers. Although generally designed to cling to the surface of skin and not move through the body like a drug, the compounds in cosmetics can still seep inside. An ingredient in a facial lotion, especially if it is lipophilic (attracted to fats), might move through the upper layer of skin into the sub-mucosa, which contains fatty tissue and small blood vessels; some of the compound might then leak into those vessels, travel through the bloodstream, heart, and lungs, and eventually reach the liver.

Modeling this complex chain of events is a difficult task for the researchers. “In the near future, one of the big challenges will be understanding exposure,” says Mark Cronin, a computational toxicologist at Liverpool John Moores University in the U.K., and the leader of one of the SEURAT research groups. “If you apply cosmetics to your skin, how much will get to a particular organ? How much will get into your blood? How much will get to the liver? And then, we need to be able to predict whether or not that will be toxic.”

To do so, the SEURAT scientists will eventually interweave models of each of these systems—the skin, heart, lungs, and more—to trace that path. In these first few years of the program, several NOTOX researchers have been focusing on developing a virtual model

“*In vitro* and *in silico* testing will play a much larger part in how we assess chemicals in the future.”

of the liver, the main detoxifying organ in the human body. Dirk Drasdo, a biophysicist and bioinformatics expert at the French research agency INRIA, toxicologist Jan Hengstler of the University of Dortmund, and their colleagues are working toward a more realistic interpretation of the liver, from the shape and architecture of the organ down to the individual hepatocytes (liver cells).

The model is still in its early stages, but Drasdo, Hengstler, and their co-workers have already shown its potential. In a 2010 paper in the *Proceedings of the National Academy of Sciences* (see Hoehme et al. below), the pair and a team of colleagues detailed an early version that showed how the liver would respond when exposed to a known toxic compound. The model’s findings matched up with laboratory tests of the same exposure, verifying its accuracy, but it also revealed a never-before-seen process that proved central to the liver’s ability to heal itself. While this result showed the potential of a virtual liver, the ultimate goal is a model that can recreate the actual processes taking place inside. “This would be as if you could look into the liver and follow the fate of all the molecules you are interested in,” Drasdo says.

Outlook


The liver is only one important aspect of the overall toxicological picture, and even the predictive model Drasdo describes remains a relatively distant goal. “To really establish this as a solid tool could take another five to 10 years,” he notes.

In the more immediate future, the SEURAT effort has other goals to achieve. Mark Cronin of the University of Liverpool notes that one of the tasks

has been to create good databases upon which the computer models can draw. “That might sound trivial,” he says, “but organizing the toxicological data has turned out to be very challenging.”

The range of scientific and logistical hurdles remaining is not good news to the cosmetics companies. When the ban went into effect last year, Cosmetics Europe, the industry group funding the research, protested that it was too soon to eliminate animal testing as the alternatives were not yet robust enough. In the meantime, they are effectively hamstrung; if they identify a new ingredient that has not previously been proven safe, they basically have to wait, as there are few ways to validate its safety now.

Still, experts say the research is progressing faster than they expected. Regardless of the exact timing, the scientists insist some combination of laboratory-based and virtual work is the future of testing.

“I don’t think there is any going back at this point in time. I think the train has left the station,” says toxicologist Bette Meek of the University of Ottawa. “*In vitro* and *in silico* testing will play a much larger part in how we assess chemicals in the future. It will happen; it is just a question of how quickly.” 

Further Reading

Towards the replacement of *in vivo* repeated dose systemic toxicity testing. *The Annual Scientific Report of the SEURAT Research Initiative*. Vol. 3, 2013. <http://www.seurat-1.eu/>

Niklas, J., Bucher, J., et al. Quantitative evaluation and prediction of drug effects and toxicological risk using mechanistic multiscale models. *Molecular Informatics*, Nov. 2012.

Gunness, P., Mueller, D., et al. 3D organotypic cultures of human HepaRG cells: A tool for *in vitro* toxicity studies. *Toxicological Sciences*, 133:1, 2013.

Hoehme, S., et al. Prediction and validation of cell alignment along microvessels as order principle to restore tissue architecture in liver regeneration. *PNAS*, June 8, 2010.

NOTOX
A video introduction to the computer-modeling-focused research cluster: <http://notox-sb.eu/film>

Gregory Mone is a Boston, MA-based writer and the author of the novel *Dangerous Waters*.

Association for Computing Machinery

Global Reach for Global Opportunities in Computing



Dear Colleague,

Today's computing professionals are at the forefront of the technologies that drive innovation across diverse disciplines and international boundaries with increasing speed. In this environment, ACM offers advantages to computing researchers, practitioners, educators and students who are committed to self-improvement and success in their chosen fields.

ACM members benefit from a broad spectrum of state-of-the-art resources. From Special Interest Group conferences to world-class publications and peer-reviewed journals, from online lifelong learning resources to mentoring opportunities, from recognition programs to leadership opportunities, ACM helps computing professionals stay connected with academic research, emerging trends, and the technology trailblazers who are leading the way. These benefits include:

Timely access to relevant information

- *Communications of the ACM* magazine
- *ACM Queue* website for practitioners
- Option to subscribe to the *ACM Digital Library*
- ACM's *50+ journals and magazines* at member-only rates
- *TechNews*, tri-weekly email digest
- *ACM SIG conference* proceedings and discounts

Resources to enhance your career

- **ACM Tech Packs**, exclusive annotated reading lists compiled by experts
- **Learning Center** books, courses, webinars and resources for lifelong learning
- Option to join **36 Special Interest Groups (SIGs)** and **hundreds of local chapters**
- **ACM Career & Job Center** for career-enhancing benefits
- *CareerNews*, email digest
- **Recognition of achievement** through Fellows and Distinguished Member Programs

As an ACM member, you gain access to ACM's worldwide network of more than 100,000 members from nearly 200 countries. ACM's global reach includes councils in Europe, India, and China to expand high-quality member activities and initiatives. By participating in ACM's multifaceted global resources, you have the opportunity to develop friendships and relationships with colleagues and mentors that can advance your knowledge and skills in unforeseen ways.

ACM welcomes computing professionals and students from all backgrounds, interests, and pursuits. Please take a moment to consider the value of an ACM membership for your career and for your future in the dynamic computing profession.

Sincerely,

A handwritten signature in black ink, appearing to read "Vint Cerf". The signature is fluid and cursive, written over a white background.

Vint Cerf

President

Association for Computing Machinery



Association for
Computing Machinery

Advancing Computing as a Science & Profession



Association for
Computing Machinery

Advancing Computing as a Science & Profession

membership application & digital library order form

Priority Code: AD13

You can join ACM in several easy ways:

Online
<http://www.acm.org/join>

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

Fax
+1-212-944-1318

Or, complete this application and return with payment via postal mail

Special rates for residents of developing countries:

<http://www.acm.org/membership/L2-3/>

Special rates for members of sister societies:

<http://www.acm.org/membership/dues.html>

Please print clearly

Name _____

Address _____

City _____ State/Province _____ Postal code/Zip _____

Country _____ E-mail address _____

Area code & Daytime phone _____ Fax _____ Member number, if applicable _____

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

I agree with the Purposes of ACM:

Signature _____

ACM Code of Ethics:

<http://www.acm.org/about/code-of-ethics>

choose one membership option:

PROFESSIONAL MEMBERSHIP:

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

STUDENT MEMBERSHIP:

- ACM Student Membership: \$19 USD
- ACM Student Membership plus the ACM Digital Library: \$42 USD
- ACM Student Membership PLUS Print CACM Magazine: \$42 USD
- ACM Student Membership w/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.

payment:

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc. in US dollars or foreign currency at current exchange rate.

- Visa/MasterCard American Express Check/money order

Professional Member Dues (\$99 or \$198) \$ _____

ACM Digital Library (\$99) \$ _____

Student Member Dues (\$19, \$42, or \$62) \$ _____

Total Amount Due \$ _____

Card # _____ Expiration date _____

Signature _____

RETURN COMPLETED APPLICATION TO:

Association for Computing Machinery, Inc.
General Post Office
P.O. Box 30777
New York, NY 10087-0777

Member dues, subscriptions, and optional contributions are tax-deductible under certain circumstances. Please consult with your tax advisor.

All new professional members will receive an ACM membership card.

Questions? E-mail us at acmhhelp@acm.org
Or call +1-800-342-6626 to speak to a live representative

Satisfaction Guaranteed!



DOI:10.1145/2580941

Michael A. Cusumano

Technology Strategy and Management

MOOCs Revisited, With Some Policy Suggestions

Assessing the rapidly evolving realm of massive open online courses.

IT IS APPROXIMATELY one year since I wrote a column on massive open online courses (see “Are the Costs of ‘Free’ Too High in Online Education?” *Communications*, April 2013). Since then, we have seen many more analyses on the subject.^a I also received several responses to my column, from positive to negative,^{3,5} and served this past year on an MIT task force examining the future of education (see <http://future.mit.edu/>).

Probably the most disturbing response to my column came from a professor in the U.S. who had decided to teach his course on one of the major MOOC platforms. He thought MOOCs would be the future and did not want to be left behind. Yet, he confessed regret that he might be contributing

to the “tragedy of the commons”: He feared his individual decision would not be good in the long run for his university or for the education profession. The image that came immediately to my mind was of the natives on Easter Island who cut down the last tree. They got fuel for another day but eventually their civilization collapsed. Did they know what they were doing?

I expressed two main concerns in

The breadth of MOOC offerings is growing but also leaves considerable room for traditional university education.

my April 2013 column:

► Free online courses might set a threshold price of “zero” for college education and seriously undermine the economic models of private colleges and universities that rely on tuition. The scenario in my mind was what happened to other industries affected by platform dynamics and the Internet, such as newspapers, magazines, books, music, video, and software products. These digital goods have close to a marginal cost of zero for reproduction and distribution but this does not mean that they have a zero cost of production or zero value.

► Many colleges and universities would offer online courses but eventually find it difficult to subsidize free education, as MIT has already experienced with Open Courseware. Then, we might be left with only a few wealthy universities and MOOC platforms dominating the online education industry.

With regard to my first concern—threats to the economics of tuition-dependent educational institutions—it

^a See, in particular, T. Lewin, “After Setbacks, Online Courses are Rethought” *The New York Times*, (Dec. 11, 2013); <http://nyti.ms/1CPTLw> and S. Adams, “Are MOOCs Really a Failure?” *Forbes.com*; <http://onforb.es/1CMCM4>.

is still much too early to gauge the impact of MOOCs or digital technologies more broadly. Nonetheless, university administrators seem to understand the economic challenges very clearly and are already making some adaptations to the free MOOCs model. For example, MIT and Harvard jointly launched EdX in 2012 after providing \$60 million in funding. The courses currently remain free, though some courses charge a fee for an ID-certified certificate. EdX in the future is likely to charge for credentials such as certificates of completion. Some of these courses may be eligible for degree credit at some institutions, though EdX does not at present offer transcripts. EdX is also licensing some materials for a fee to other institutions. Coursera is heading down a similar path, that is, to charge for credentials or grading. Udacity already charges for grading. In other words, a business model is emerging.

A business model is critical because education always costs something to produce. In the residential world, the most elite institutions, which include MOOC pioneers such as MIT, Harvard, Stanford, Princeton, and the University of Pennsylvania, set the price for tuition. Today this is over \$50,000 per year. These prices are then copied by other institutions. However, only the elite schools have large enough endowments and diverse sources of revenue that allow them to give significant financial aid to needy students as well as to subsidize experiments such as free MOOCs. The average MIT student, for example, pays only half the nominal tuition rate. Moreover, net tuition has not exceeded more than 15% of MIT's revenues in recent decades; we rely much more heavily on research funding as well as endowment income.⁴ Another economic challenge is that MOOCs are more expensive to create and produce. They resemble movie productions far more than traditional college classes, especially if they require small armies of teaching assistants to be effective.

With regard to my second concern—that a few Web platforms, led by the most elite institutions, would dominate the MOOCs movement—this does seem to have occurred. Again, however, we see some adaptations. There are three main MOOC platforms: EdX, which is non-profit, as well as the for-profit Coursera (founded by

two Stanford professors in 2012, with \$85 million in venture capital) and the for-profit Udacity (established in 2012 by Sebastian Thrun, a formerly tenured professor at Stanford and Google Fellow, and two other partners, with substantial venture funding). However, many universities and colleges now contribute content to the two main platforms. As of early 2014, EdX counted 27 institutions among its members and Coursera 108 (see <https://www.edx.org/> and <https://www.coursera.org/>).

Enthusiasm for MOOCs one year later also seems dimmer because data so far suggests they are unlikely to replace in-person education anytime soon. For

State in small classes of 100 students, found the online students did much more poorly than regular students, even with teaching assistants. (EdX also did an experiment with San Jose State and got somewhat better results, supporting the argument that MOOCs can work well when combined with live instruction in a “blended” education model.^b) Udacity is now trying to work with companies to offer vocational training rather than college classes. In particular, it has partnered with AT&T and Georgia Tech to offer a three-semester master's degree in computer science for \$6,600—one-seventh the tuition rate for out-of-state students.¹



example, the *New York Times* recently gave front-page coverage to a University of Pennsylvania Graduate School of Education report involving a million students. Only about 4% of those who registered completed a MOOC. Half of the registered students never even viewed a lecture. In addition, MOOCs do not seem to be educating the impoverished third-world masses; rather, they are providing continuing education to relatively wealthy students of working age, some 80% of whom already have college degrees.² Other experiments, such as between Udacity and San Jose

The breadth of MOOC offerings is growing but also leaves considerable room for traditional university education. The first MOOCs were based on large undergraduate introductory lectures. Putting these types of courses online has many advantages: Students can learn at their own pace, there is no need to keep giving the same lectures year after year, students can view the lectures from different locations or

^b Email comment by Sanjay Sarma, MIT professor of mechanical engineering and director of digital learning, January 11, 2014.

institutions, among other benefits. Yet having access to live instructors also helps students learn, as the San Jose State experiments suggest. It is possible to have interactive Web classes (the online and for-profit University of Phoenix has done this for years), but these become increasingly difficult as the number of students rise. In addition, many advanced classes and seminars do not adapt well to the MOOCs format.

In short, we have made considerable progress finding ways to balance laudable educational goals and technological progress with economic and pedagogical realities. Nonetheless, there remain several questions we still need to resolve. I have been taking notes the past year and consulting with my task force colleagues, and this is my current list:

► Should MOOCs aimed at general education remain free? I think this is possible and desirable. They will require subsidies to produce and deliver, and salaries for the faculty and teaching assistants. However, the high-traffic MOOC platforms can generate indirect revenue to offset some costs, such as by selling ads or lists of CVs, or licensing content. Wealthy universities and colleges as well as foundations and governments also can contribute some funding. Venture capitalists are involved as well, though it is anyone's guess whether their investments will pay off.

► Should MOOCs with a credential, grade, or credit toward a college degree be free? I think not because I still believe "free" in the long run will damage the economic model of the many non-profit educational institutions that rely on tuition. There is another purpose as well to setting a price on these courses. If there are even very modest charges, it is likely the number of students who register for MOOCs will drop dramatically. However, the number of students who complete the courses should also rise dramatically. We need to run more experiments. I would try to set the price of a credentialed or graded MOOC to balance these two goals—providing education to people who cannot come to a college campus versus making enough money to cover costs plus some excess to invest (such as in new course development or infrastructure).

► What about institutions or individual faculty that want to emphasize MOOCs' philanthropic potential? Surely, we can still offer education for free or at very low cost to many students around the world through scholarships or tuition waivers, just as we already do for traditional students.

► How should institutions treat MOOCs in terms of degree credit, apart from tuition charges? Some schools recognize courses taken at other schools in order to waive requirements but not to accelerate completion of a degree; other schools accept transfer credit toward a degree but with some limits. I would treat internal MOOCs as regular classes and external MOOCs from accredited institutions, as long as they come with grades and credit, like any other college classes where a student applies for transfer credit.

► Should a student be able to get a college degree solely through taking MOOCs? I think the answer here is a qualified yes, but I would treat the degrees more like we currently treat extension school degrees—give them a specific designation. It is already possible for students to get regular college and advanced degrees (even Ph.D.'s!) fully online from some institutions, with or without MOOCs. The big question for me is whether a student who only takes, say, EdX or Coursera classes, should get *the identical degree* as a student who physically attended Harvard, Stanford, MIT, Berkeley, Princeton, Penn, Michigan, or other universities where only a fraction of the applicants are admitted? At present, there is too much variance in the quality of the students and the educational experience is not the same. They should not get the same degrees. Nonetheless, EdX, Coursera, and other MOOC platforms may themselves evolve into degree-granting institutions.

These are brief policy suggestions for some difficult questions. When it comes to education, there are also larger issues at stake, as reflected in another email message I received from a former business school dean. He too worried about the threat MOOCs might have on the business models of tuition-dependent universities. More than this, though, he worried about the *need* to threaten institutions such

as his. He thought the faculty union at his school had grown too powerful over the years and used its influence to resist curriculum innovations as well as to undermine the tenure process by limiting outside evaluations, which focused on research quality. Ultimately, he saw student education as suffering. So, whatever else they may do, MOOCs can be a useful "kick in the pants." They can persuade complacent professors and institutions to improve their educational product lest we be replaced by online videos and grading software.

My greatest concern at this point is clarity in mission. Should universities and colleges focus on educating their local tuition-paying students or on educating the world? Many professors do both, such as by writing mass-market books and articles, and they do research. But it is very difficult to do everything well. Creating and running a successful MOOC seems to be extraordinarily difficult and time consuming, and not what professors are usually trained to do. So what is the first priority? Too much attention on how to better *disseminate existing knowledge* may ultimately weaken our ability to *create new knowledge*. It would indeed be a "tragedy of the commons" if the fascination with online courses diminishes the time and commitment of our best faculty to do world-class research, which we need to create the MOOCs as well as the conventional classes of the future. ■

References

1. Chafkin, M. Udacity's Sebastian Thrun, godfather of free online education, changes course. *Fast Company* (Dec. 2013/Jan. 2014); <http://www.fastcompany.com/3021473/udacity-sebastian-thrun-uphill-climb>.
2. Lewin, T. After setbacks, online courses are rethought. *The New York Times* (Dec. 11, 2013), A1; <http://nyti.ms/1JCPTLw>.
3. Lohr, S. Beware the high cost of 'free' online courses. *The New York Times* (Mar. 25, 2013); <http://bits.blogs.nytimes.com/2013/03/25/beware-of-the-high-cost-of-free-online-courses/>.
4. Massachusetts Institute of Technology. Institute-wide Task Force on the Future of MIT Education. Preliminary Report (Nov. 21, 2013); <http://future.mit.edu/preliminary-report>.
5. Randall, E. Is the Internet sending higher education the way of the newspaper industry? *Boston Magazine*; <http://www.bostonmagazine.com/news/blog/2013/03/27/is-the-internet-sending-higher-education-the-way-of-the-newspaper-industry/>.

Michael A. Cusumano (cusumano@mit.edu) is a professor at the MIT Sloan School of Management and School of Engineering and author of *Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Uncertain World* (Oxford University Press, 2010).

Copyright held by Author/Owner(s).

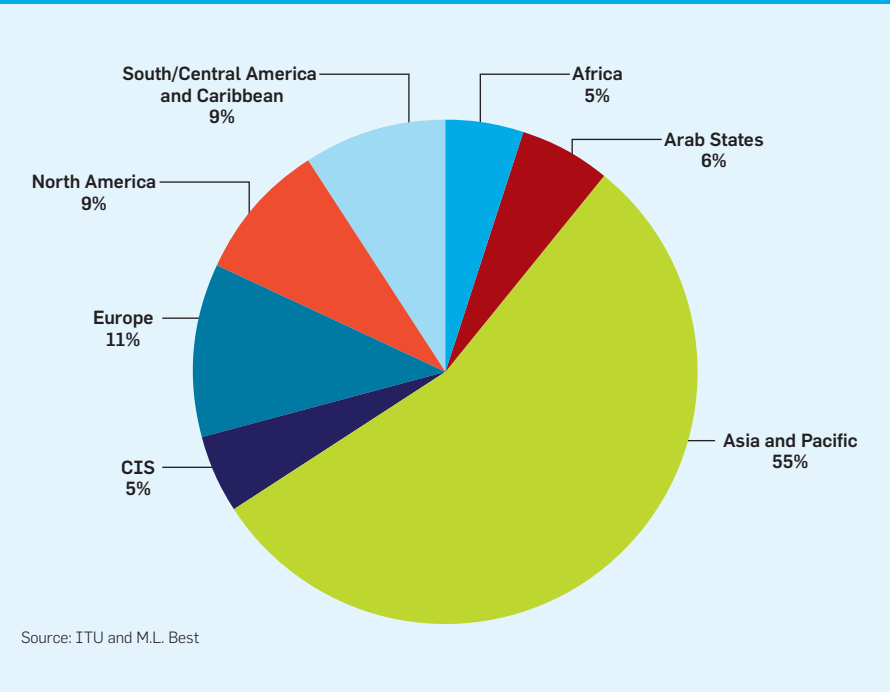
Global Computing Thinking Outside the Continent

Encouraging the opportunities for digital innovation and invention to flourish in a variety of social environments.

IN A RECENT STUDY I conducted with the International Telecommunications Union (ITU)³ we operationalized the concept of the Digital Native and proceeded to count them across the planet. Criticisms of the Digital Native premise notwithstanding (and there are many) we argued that young Digital Natives are driving the adoption and inventing the future of information and communication technologies (ICTs) in many unlikely places across the globe. In Europe or North America almost everyone is online; and this is even truer for the young there. Meanwhile, the percentage of the population digitally networked in Africa and South Asia is low, however the proportion of youth adopters is relatively high: in some parts of Africa there are two or three online youth for every online adult.

Small levels of digital penetration are one explanation why large computing multinationals and research universities routinely ignore Africa and parts of Latin America and Asia. But a look specifically at these regions' Digital Natives reveals a collection of innovators and users spearheading some of the world's most exciting ICT advances. These Digital Natives are leading through rule-breaking lateral thinking but they also simply lead in their sheer size. The number of global Digital Natives disaggregated by region shows Europe and North America as but small slivers of the worldwide pie (see Figure 1). A paradox looms: computing researchers

Figure 1. Global distribution of Digital Natives.



and practitioners design for the Global North, but the sheer numbers and opportunities for innovation are overflowing in the Global South.^a

^a Terminologies describing Africa, Latin America, and much of Asia have varied with the times. The “Third World” emerged from the Cold War, which is now as obsolete as the Soviet Union. “Developing world,” though commonly used, seems pejorative and patronizing. The term “Global South” describes a collection of countries located mostly in the southern hemisphere, while “Global North” refers to what some have called the “developed world.”

A Case of Nigerian Digital Native Innovation: The Social Media Tracking Centers

Nigeria is Africa's most populous country, a nation of enormous possibility and seemingly endless challenges. The deeply contested political environment has played out in a series of fraught elections. During the lead-up to national elections in 2011 the country seemed to hang on tenterhooks. Nigerian Digital Native activists, calling themselves the “Facebook generation,” invented ways to use social media and mobile

ACM Transactions on Accessible Computing



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

www.acm.org/taccess
www.acm.org/subscribe



Association for
Computing Machinery



Figure 2. Social media tracking center in Abuja, Nigeria, in 2011. The monitor depicts histogram visualizations of trending reports created by Aggie software.

apps to help ensure a free and fair election. The Social Media Tracking Center (SMTC) and the Aggie social media aggregator software were results of this.^b

The Aggie system, developed primarily at Georgia Tech, combs social media sources from Twitter, Facebook, Google+, Ushahidi, and other social media platforms. The data is streamed in real time to Aggie, which presents trends grouped around voting logistics, violence, political parties, and so forth. The SMTC team, based in Nigeria's capital city of Abuja, watched these trends (see Figure 2), detecting possible election irregularities or occasions of violence that warranted further attention. Reports categorized the incidents, which were relayed to the election commission, police, and other relevant stakeholders.

Approximately 750,000 reports were analyzed through the SMTC system during the three-week election period. Social media activity peaked during the April 16, 2011 Presidential election. When violence erupted in the North of the country, Aggie received nearly 50 reports a second. The system has been replicated for elections in Liberia, Ghana, and Kenya, with results that are only now being analyzed robustly⁵ though initial results show great promise.

Computing Innovation and the Global South

Computing practitioners and researchers often design for the Global North

^b Many groups collaborated on this including Enough is Enough, the Shehu Musa Yar'Adua Foundation, the MacArthur Foundation, and other donors, as well as partners such as Georgia Tech and Harvard University.

but much of the action is in the Global South. The preceding case study demonstrates the energy of African Digital Natives driving opportunity and invention in the Global South. Young people in these contexts will continue to surface opportunity for exploration and invention, in areas such as:

► **Networks and infrastructure.** The Global South is seeing exponential growth in networks and infrastructure (while the rest of the world moves with-in replacement and upgrade cycles). Broadband penetration is on the rise. But ubiquitous cloud-computing capable access is still a goal not entirely met; the network reality for most people feels like it is partially cloudy with occasional thunderstorms.⁴ All sorts of design challenges and research opportunities, from intermittent connectivity to community clouds, demand our attention.

► **HCI.** Usability and interface design are not common fields of work in the Global South, but this is not for want of opportunity. Africa is the world's most linguistically rich continent, but many Africans lack print literacy in their native language. Non-print interfaces (voice or visual) might provide rich innovation spaces. We continue to deploy *personal* computers into places where the technology is mostly shared and not kept by a single person.¹ Do we need a *community* computer instead? What does the desktop metaphor mean in a context that does not value or use desks? Why do we rely on the QWERTY keyboard for languages that do not include the 'Q', 'W', or 'E'?

► **Channels.** A lot of discussion has focused on the prominent rise of mobile phone use in low-income coun-

tries and thus whether mobile phones are the technological “winners.”⁶ The global rise of mobile phone networks, now usually with data support, is clear; and the desirability of mobility itself is also clear. Similarly, low-cost laptop initiatives have captured considerable attention—sometimes suggesting these particular systems will solve all the problems of development. In reality, neither mobile phones nor laptops are the perfect appliances for all situations. We need to better understand what the best design and form factors are for end-user appliances regardless of the network or distribution model. Do we need to design an entirely new appliance, something with a more appropriate display or input device or better suited to end-user sharing for instance?

► **Software engineering.** Is there a software engineering approach unique to the Global South? It is an odd question, but one that keeps surfacing as I collaborate in Africa or Asia. In any engineering practice the cultural contexts matter, as does the training and engineering capacities. So why should not there be software engineering practices unique to and particularly designed for the African (or for that matter the Nigerian) context? This approach might hybridize some of the flexibility of agile methods (in particular to help focus on testing and to combat feature bloat) with some of the more conventional structured methods (which resonate with some culture’s top-down traditions).

► **Sustainability.** When issues of sustainability arise in computing initiatives in the Global South they tend to focus on financial self-sustainability tethered to market forces and neo-liberal economic theory. However, there are other forms of sustainability that demand our attention: environmental, technological, social and cultural, political and institutional. Work in computer science can touch on all of these forms of sustainability. For instance, technical sustainability will be enhanced by easy-to-use systems or systems that allow for remote maintenance. Similarly, low-power-consuming devices enhance environmental sustainability.

Computing researchers and practitioners want to think outside of the box. That is easy: think outside of the

Computing practitioners and researchers often design for the Global North but much of the action is in the Global South.

continent! This column will push thinking beyond traditional borders to where the population, and the opportunities to innovate, both flourish.

At the recent ITU Global Youth Summit, representatives of the world’s Digital Natives called for more access and more invention. “The spread of information amongst young people can directly foster empowerment and innovation on a global scale,” they wrote. “Health, civic engagement, online protection, environmental protection and economic success all depend on having unfettered access to knowledge which ICTs can extend to everyone.”² By following the lead of these Digital Natives, especially those coming from the Global South, we just might help them invent, and protect, our global future. ■

References

1. Best, M.L., Garg, S., and Kollanyi, B. *Understanding and Rethinking Shared Access: How People Collaborate and Share Knowledge and Technologies in Ghanaian Cybercafés*. Global Impact Study Research Report Series. Technology and Social Change Group, University of Washington Information School, Seattle, 2013.
2. ITU. *BYND 2015: 2013 Costa Rica Declaration*. ITU, Geneva, (2013); <http://www.itu.int/en/bynd2015/Documents/bynd2015-global-youth-declaration-en.pdf>.
3. ITU. Measuring the world’s digital natives. In *Measuring the Information Society*. ITU, Geneva, (2013), 127–158.
4. Kelly, T. and Rossotto, C.M. *Broadband Strategies Handbook*. World Bank Publications, Washington, D.C., 2011.
5. Smyth, T. and Best, M.L. Tweet to trust: Social media and elections in West Africa. Presented at the Sixth International Conference on Information and Communication Technologies and Development (ICTD2013), Cape Town, South Africa, 2013.
6. *The Economist*. The real digital divide. (Mar. 10, 2005).

Michael L. Best (mikeb@cc.gatech.edu) is an associate professor at the Sam Nunn School of International Affairs and the School of Interactive Computing at Georgia Institute of Technology where he directs the Technologies and International Development Lab. He is also faculty associate of the Berkman Center for Internet & Society at Harvard University.

Copyright held by Owner/Author(s).

Calendar of Events

June 16–19

The 11th Annual International Conference on Mobile Systems, Applications, and Services, Bretton Woods, NH, Sponsored: SIGMOBILE, Contact: David Kotz, Email: kotz@cs.dartmouth.edu

June 16–20

ACM SIGMETRICS/ International Conference on Measurement and Modeling of Computer Systems, Austin, TX, Sponsored: SIGMETRICS, Contact: Sanjay Shakkottai, Email: shakkott@austin.utexas.edu

June 21–25

Innovation and Technology in Computer Science Education, Uppsala, Sweden, Sponsored: SIGCSE, Contact: Asa Cajander, Email: asa.cajander@it.uu.se

June 22–27

International Conference on Management of Data, Salt Lake City, UT, Sponsored: SIGMOD, Contact: Curtis Dyreson, Email: curtis.dyreson@usu.edu

June 23–26

ACM Web Science Conference, Bloomington, IN, Sponsored: SIGWEB, Contact: Filippo Menczer, Email: fil@indiana.edu

June 25–27

ACM International Conference on Interactive Experiences for TV and Online Video, Newcastle Upon Tyne, U.K., Sponsored: SIGCHI, Contact: Patrick Olivier, Email: p.l.olivier@ncl.ac.uk

June 25–27

19th ACM Symposium on Access Control Models and Technologies, London, ON, Canada, Sponsored: SIGSAC, Contact: Sylvia L. Obsorn, Email: sylvia@csd.uwo.ca

June 30–July 2

International Conference on Systems and Storage, Haifa, Israel, Sponsored: SIGOPS, Contact: Eliezer Dekel, Email: dekel@il.ibm.com



Article development led by **acmqueue**
queue.acm.org

Kode Vicious This Is the Foo Field

The meaning of bits and avoiding upgrade bogdowns.

Dear KV,

When will someone write documentation that tells you what the bits *mean* rather than what they *set*? I have been working to integrate a library into our system, and every time I try to figure out what it wants from my code, all it tells me is what a part of it is: “This is the foo field.” The problem is that it does not tell me what happens when I set foo. It is as if I am supposed to know that already.

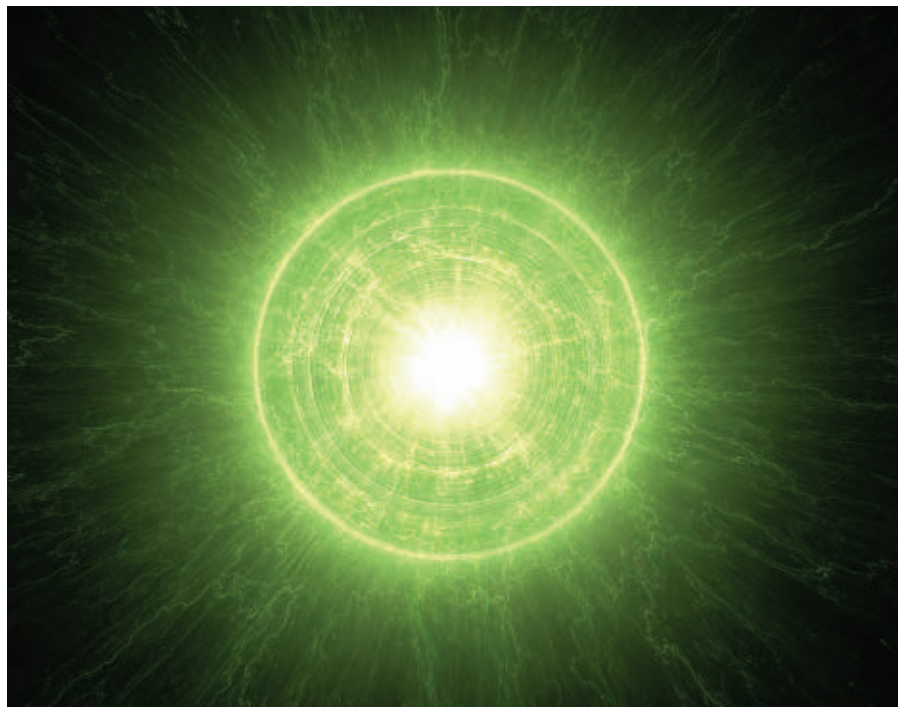
Confoosed

Dear Confoosed,

Nowhere is this problem more prevalent than in hardware documentation. I am sure Dante listed a special ring of hell for people who document this way, telling you what something is while never explaining the why or how.

The problem with that approach is assumed knowledge. Most engineers, of both the hardware and/or software persuasion, seem to assume the people they are writing documentation for—if they write documentation at all—already have the full context of the widget they are working on in their heads when they start to read the docs. The documentation in this case is a reference, but not a guide. If you already know what you need to know, then you are using a reference; if you do not know what you need to know, then you need a guide. Companies that care about their documentation will, at this point, hire a decent technical writer.

The job of a technical writer is to tease out of the engineer not only the



what of a device or piece of software, but also the *why* and the *how*. It is a delicate job, because given the incredible malleability of software, one could go on for thousands of pages about the *what*, not to mention the *why* and *how*. The biggest problem is that the *what* is the easiest question to answer, because it is in the code when dealing with software, or the VHDL (VHSIC Hardware Description Language) when dealing with hardware. The *what* can be extracted without talking to another person, and who really wants to spend the day pulling engineers' teeth to get coherent explanations about how to use their systems? Since it is

easiest to get at the *what*, most documentation concentrates on this part, often to the exclusion of the other two. Most tutorial documentation is short, and then at some point the rest of it is left as “an exercise to the reader.” And exercise it is. Have you ever tried to lift a reference manual?

Although many engineers and engineering managers now give lip service to the need for “good documentation,” they continue to churn out the same garbage that technical people have joked about since IBM intentionally left pages blank. A good writer knows that his or her job is to form in the mind of the reader a sense and an

image of what the writer is trying to communicate. Alas, programmers and engineers have rarely been known as good writers; in fact, they are most often known as atrocious writers. It turns out that writers often want to relate, in some way, to people. That is, however, not something often said about technical folk, and in fact, it is often quite the opposite. Most of us want to go off into a corner and “do cool stuff” and be left alone. Unfortunately, none of us works in a vacuum and so we must at least learn to communicate effectively with others of our ilk, if only for the sake of our own project deadlines.

Every software and hardware developer should be able to answer the following questions about systems they are developing:

1. Why did you add this? (field, feature, API)
2. How is this field, feature, or API used? Give an example.
3. Which other fields, features, and APIs are affected by using the one you are describing.

And if the answer to #1 is, “Management told me to,” then it is time to fire management, or find a new job.

KV

Dear KV,

During a recent rollout, I overheard one of our DevOps folks bemoaning the fact that upgrading our software had slowed down the overall system. This is a complaint I hear a lot, so I think it is happening more often. The problem is the folks at my gig do not do enough performance testing and just upgrade systems whenever our vendors tell them to so that they will not miss any new features, whether or not they use those features.

Bogged Down by Upgrades

Dear Bogged,

What you are really seeing are 10-year-old expectations trashed by modern hardware trends. Everyone in computing has been talking about the end of frequency scaling for at least five years, and probably more. While lots of folks sounded the warning about this problem, and talked at length about the ways in which software would have to change to meet them, not enough

I am often amazed by people who upgrade software and expect it automatically to be faster.

software has been rewritten—oh, I’m sorry, I meant *refactored*—to handle this new reality. I am often amazed by people who upgrade software and expect it automatically to be faster.

Expecting more features makes some sense, because that is what marketing and management are always going to push for in a new version of a system. The more boxes you can tick, the more money you can charge, even if the things provided are of little or no use. Given that upgrades always include new features, what makes anyone think the system provided will run any faster? Surely more code to execute means the system will run slower and not faster after the upgrade—unless you upgrade your hardware at the same time. None of which is to say that this must be the case—it is simply that it often *is* the case.

The end of frequency scaling, the ever-upward tick of CPU frequencies, was supposed to spur the software industry into building applications that took advantage of multiple cores, as transistor density is still climbing, even if clock frequency is not. Newer software does seem to take advantage of multiple cores in a system, but even when it does, another problem is presented: memory locality. Anyone who has been building software on the latest hardware knows the programs now need to know where they are running in order to get fast access to memory. In multiprocessor systems, memory is now nonuniform, meaning if my program runs on processor A but the operating system gives me memory nearer to processor B, then I am going to be very, very annoyed.

Modern operating systems are trying to handle NUMA (nonuniform memory access) correctly, but when

they get it wrong, you become—as you signed this letter—bogged down again. These are the new rules of the game programmers must contend with. Processors are not getting faster; they are splitting into parallel machines with nonuniform memory. In the current environment, we now need to worry about all the things we may have last seen in parallel-programming classes in graduate school. All programming will now be threaded programming, and we will have to deal with all that entails, plus the fact that we now need to know where our memory is coming from. My advice is to switch careers from programming to ditch digging (where at least at the end of the day you will know you did something). If you cannot switch careers, here are a few things you will need to do and check as you try to improve the responsiveness of your code:

- ▶ Learn to write correct threaded programs. Writing threaded code is hard, but there are plenty of books on this topic to help you.

- ▶ Keep your threads from sharing state whenever possible.

- ▶ Learn the APIs your operating system gives you to figure out where your thread is relative to the CPU and memory.

- ▶ Bake debugging for threaded code into your system if it is not easily available as a library. There are few things more exquisitely painful to a software engineer than tracking down a race condition with `printf()` and a spoon.

KV

Related articles on queue.acm.org

Keeping Bits Safe: How Hard Can It Be?

David S.H. Rosenthal
<http://queue.acm.org/detail.cfm?id=1866298>

Successful Strategies for IPv6 Rollouts. Really.

Thomas A. Limoncelli and Vinton G. Cerf
<http://queue.acm.org/detail.cfm?id=1959015>

Get Real about Realtime

George Neville-Neil
<http://queue.acm.org/detail.cfm?id=1466445>

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

Copyright held by Owner/Author(s).

Viewpoint small data, where $n = me$

Seeking personalized data-derived insights from analysis of our digital traces.

WE HEAR A LOT about how big data, smart devices, and all the ‘-omics’ (for example, genomics, proteomics, metabolomics, and so forth) are going to transform medicine—and they will. But there is another force that is going to change the way we think about and practice health, and that is our small data—small data derived from our individual digital traces.

Consider a new kind of cloud-based app that would create a picture of your health over time by continuously, securely, and privately analyzing the digital traces you generate as you work, shop, sleep, eat, exercise, and communicate. While there are personal devices and Internet services specifically designed for self-tracking (Fitbit, Patients like me, <http://quantifiedself.com>, and so forth), digital traces include a much richer corpus of data that we generate every day, just by virtue of our normal activities. And while the use of electronic health records is increasing, today’s systems capture data reported by clinicians, not patients; and data about clinical treatment, not day-to-day activities.

We generate these data because most of us mediate, or at least accompany, our lives with mobile technologies. As a result, we all leave a continuously updated “trail of data breadcrumbs” behind us, which together make up our digital traces. You all are generating such traces now, as

you do when you wake up and perhaps read email before you even got out of bed, or when you decide to take a walk after work instead of staying home and frequenting your refrigerator and couch.

The social networks, search engines, mobile operators, online games, and e-commerce sites we access every hour of most every day extensively use these digital traces we leave behind. They aggregate and ana-

lyze these traces to target advertisements and tailor service offerings and to improve system performance. But most services do not make these individual traces available to the person who generated them; they do not yet have a ready-made vehicle to repack-age their data about you in a useful format for you and provide it to you. But they should, because this broad but highly personalized data set can be analyzed to draw powerful infer-



PHOTOGRAPH BY LIZARD

ences about your health and well-being from your “digital behavior.”

To be clear, I am not talking about apps doing detailed medical diagnosis, and I am *not* talking about replacing the insight and role of doctors or loved ones, nor am I discounting the importance of our own self-awareness. Instead, use of these traces could serve to greatly enhance all of those with personalized data-driven insights—insights ranging from early warning signs of a problem, to indicators of gradual improvement. Ginger.io (<http://ginger.io>) refers to this sort of services as a check engine light. Another way to think of it is as a personalized “behavioral pulse.” A signal that can indicate subtle but significant changes in a person’s well-being by representing changes in day-to-day behavior, in a manner that is comfortable to share with a select number of friends or family.

Once I, as a patient and consumer, can access the data that service providers have collected and stored about me, I can then use these data to fuel apps I choose to subscribe to. For example, imagine an app that helps my doctor determine whether the new medication dosage I have been taking for the last two weeks is better for me than the previous dosage. The app could create a comparative picture of my daily function this month relative to last month by automatically analyzing motion, location, and vocabulary data plucked from my digital traces. Or, I could see, from an app running over my location traces that I get back from AT&T or Verizon, if the supplement I am taking for my early-stage arthritis is actually helping me get out and about more quickly most days; and if overall I am less sedentary than I was previously.

From chronic pain to depression to memory enhancement and Crohn’s Disease—many chronic conditions have a lot of day-to-day variability, with confounding factors. Moreover, both good and bad changes are gradual. Consequently, it is difficult for me as an individual to reliably and precisely track the effect of a new treatment based only on my subjective and selective memory. But these same health conditions have symptoms and side effects that show in our functional,

You will be the customer for the data about you; I will be the customer for the data about me.

everyday, behaviors—and for the first time really, our everyday behaviors are becoming data. While that might be disconcerting at times, it is the case; and what I am arguing for is that *we as individuals should have access to our digital traces so that we can mine them for our own purposes.*

And we can do this for the young and old alike, because while we do not usually think of elders as digital natives, they do increasingly carry cell-phones (even if only simple phones); and they increasingly use the Internet (even if only via their TV). Both simple phones and cable TV boxes are potential sources of digital traces! And, of course, as we become the elders of tomorrow we will carry with us our existing digital practices and addictions into our senior years. When I think back to my father’s final few months of life, I can identify signals that indicated that something was wrong, signals that could have shown up in his digital behavioral pulse if one had been available. He suddenly stopped sending email (and this was a man who had been using email on the Arpanet since the mid-1970s), and his daily patterns gradually changed so that he no longer shopped at the supermarket to prepare food at home for my mother, and he took shorter and shorter neighborhood walks. His declining condition was not detectable on his regular visits to his cardiologist since it did not show up in his EKGs, or traditional exchanges about how he felt, and he like others “pulled it together” for his favorite doctor. On an emergency room visit one day, the attending doctor observed nothing atypical for a 90-year-

old man; nothing in his vitals or his electronic health record communicated to the emergency room doctor that this 90-year-old man was behaving entirely differently than he was just a few weeks earlier—a behavioral pulse graph, derived from his digital traces, could have. Having access to my father’s ‘digital behavioral pulse’ would not have changed the outcome; but it would have given us the tools to track these changes and communicate them objectively to members of his medical team.

Fortunately, I have a “real” doctor in the family—my eldest sister Margo—and her insight and vigilance in keeping detailed track of my parents’ medical history and day-to-day activities effectively created a behavioral pulse for my father, but most families do not have a ‘Margo’. So, what I am suggesting is that we begin to leverage our small data to bring more vigilance and insight to everyday care. *We can think of this as new kind of medical evidence, evidence where $n=me$, because it complements traditional big- N population studies with data that are just about me (or you) over time.* And what is so compelling about this approach is that these data already exist. It does not require deployment of any new hardware, so we can start leveraging our small, $n=me$, data *now*.

So, if the raw data are there, what is left to do to make small data and $n=me$ become the standard of care? First and foremost, I do not in any way want to trivialize the work that will be needed to convert these noisy sources of data into actual insight—that is where we will see much of the iterative innovation in the coming years from the computing community in particular. But it will not happen until we can start tapping into our own data. Therefore, our first step has to be what Todd Park refers to as data liberation: we need to liberate our data from mobile and Internet services, to you and me. We need a common (open) architecture so that a rich market of apps and services can grow around our $n=me$ data in the same way the HTTP standard created the World Wide Web with its myriad apps and services.

Admittedly, some service providers are apprehensive about whether

customers will be put off once they see how telling their digital traces are and worry it will create a public relations nightmare. But the data are already being captured for the most part, and in the long run consumers will know what is going on anyway. Perhaps transparency will lead to a more robust and sustainable basis for privacy. Assuming we overcome such disincentives, where are the positive incentives for commercial service providers to cooperate and make digital traces available to the individual? The economics of the market seem to be on our side. On the cost side, these digital traces are already recorded by the service providers so the added cost of providing small data to the customer can be quite low. In terms of benefits, if standard interfaces to personal digital traces spark a cottage industry of app makers who process small data and put it to work for subscribers, then implicitly they could increase the value of the consumers' engagement with the underlying digital services;

in the same way that mobiles apps greatly increased the value to consumers of smartphones. In other words, the business case for the service providers could be one of marketing and sustaining customer engagement, as well as in opening up new service offerings based on their own new, small-data and personal data repository, offerings.

Again, it is never as simple as just getting the data. We face intriguing technical and design challenges in making sense of that data for the users, and we have regulatory challenges in navigating and adapting FDA, HIPAA, and privacy policies; for example, whether to treat this data as medical data or something more akin to personal diaries. But I do not think any of these are showstoppers; if we start the flow of $n=me$ data, we can make the right things happen, and in the right way.

With my colleagues at Open mHealth (<http://openmhealth.org>) and Cornell Tech (<http://tech.cornell.edu>), we are building prototypes that demonstrate the power of small $n=me$ data, and we are developing standard interfaces that service providers, app creators, and science researchers can use to build the applications that will process, fuse, and filter your small data for you. We have created a website to let you tell service providers we want our digital traces formatted and made available to us: <http://smalldata.tech.cornell.edu>. You will be the customer for the data about you; I will be the customer for the data about me. Let's get our search engines, social networks, and mobile carriers, to start packaging our small data, for us. ■

Further Reading

Acquisti, A. and Heinz College. *The Economics of Personal Data and the Economics of Privacy. Background Report for The Economics of Personal Data and Privacy: 30 Years after the OECD Privacy Guidelines.*

Becker, R. et al. *Human mobility characterization from cellular network data. Commun. ACM* 56, 1 (Jan. 2013), 74–82

Campbell, A.T. et al. *The rise of people-centric sensing. IEEE Internet Computing* 12, 4 (Apr. 2008), 12–21.

Estrin, D. and Sim, I. *Open mhealth architecture: An engine for*

health care innovation. Science 330, 6005 (2010), 759–760.

Kang, J. et al. *Self-surveillance privacy. Iowa Law Review* 97 (2012), 809.

Kleinberg, J. *Bursty and hierarchical structure in streams. In Proceedings of the 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2002.*

Kumar, S. et al. *Mobile health: Revolutionizing healthcare through transdisciplinary research. IEEE Computer* 46, 1 (Jan. 2013), 28, 35.

Miller, G. *The smartphone psychology manifesto. Perspectives on Psychological Science* 7, 3 (2012), 221–237.

Morris, M. et al. *PIXEE: Pictures, interaction and emotional expression, ACM CHI 2013 Extended Abstracts (Apr. 27–May 2, 2013, Paris, France).*

The Organisation for Economic Cooperation and Development. *Exploring the Economics of Personal Data: A Survey of Methodologies for Measuring Monetary Value.*

Tucker, C. *The economics value of online customer data. Background report for The Economics of Personal Data and Privacy: 30 Years after the OECD Privacy Guidelines.*

<http://mobile.nytimes.com/2013/05/26/technology/for-consumers-an-open-data-society-is-a-misnomer.html>

<http://givememydata.com/>

<https://www.google.com/settings/activity>

<http://www.weforum.org/issues/rethinking-personal-data>

<http://blog.stephenwolfram.com/2012/03/the-personal-analytics-of-my-life/>

<http://wethedata.org>

<http://patientslikeme.org>

<http://quantifiedself.com>

<http://ginger.io>

<http://web.media.mit.edu/~sandy/>

<http://www.amia.org/amia2011/keynotes>

<http://research.microsoft.com/~horvitz/logdata.htm>

<http://pac.cs.cornell.edu/>

Deborah Estrin (destrin@cs.cornell.edu) is a professor of computer science at Cornell NYC Tech.

This Viewpoint is based on a talk presented at TEDMED 2013. The author thanks Gregory Abowd, Faisal Alquaddoomi, Marjory Blumenthal, Tanzeem Choudhury, Mark Hansen, Andy Hsieh, Lynnette Millett, Fred Schneider, Ben Shneiderman, Ida Sim, Marcus Webb, and anonymous reviewers.

Copyright held by Author/Owner(s).



ECSEE
European Conference
Software Engineering
Education 2014

27 November and
18 November 2014

Seon Monastery
Germany

Full Paper
Submission Deadline:
16 May 2014

Conference Website
www.ecsee.eu

Viewpoint

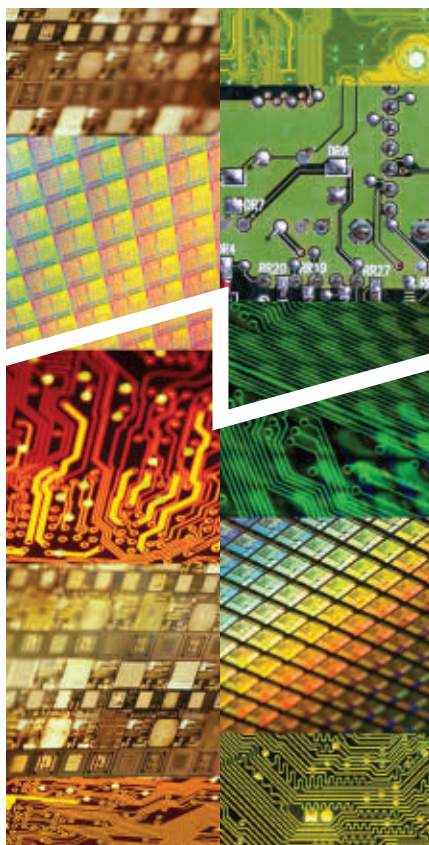
Is Multicore Hardware for General-Purpose Parallel Processing Broken?

The current generation of general-purpose multicore hardware must be fixed to support more application domains and to allow cost-effective parallel programming.

IN THE RECENT decade, most opportunities for performance growth in mainstream general-purpose computers have been tied to their exploitation of the increasing number of processor cores. Overall, there is no question that parallel computing has made big strides and is being used on an unprecedented scale within companies like Google and Facebook, for supercomputing applications, and in the form of GPUs. However, this Viewpoint is not about these wonderful accomplishments. A quest for future progress must begin with a critical look at some of the current shortcomings of parallel computing, which is the aim of this Viewpoint. This will hopefully stimulate a constructive discussion on how to best remedy the shortcomings toward what could ideally become a parallel computing golden age.

Current-day parallel architectures allow good speedups on regular programs, such as dense-matrix type programs. However, these architecture are mostly *handicapped on other programs, often called “irregular,” or when seeking “strong scaling.”* Strong scaling is the ability to translate an increase in the number of cores to faster runtime for problems of fixed input size. Good speedups over the fastest serial algorithm are often feasible only when an algorithm for the problem at hand

can be mapped to a highly parallel, rigidly structured program. But, even for regular parallel programming, cost-effective programming remains an issue. The programmer’s effort for achieving basic speedups is much higher than for basic serial programming, with some limited exceptions



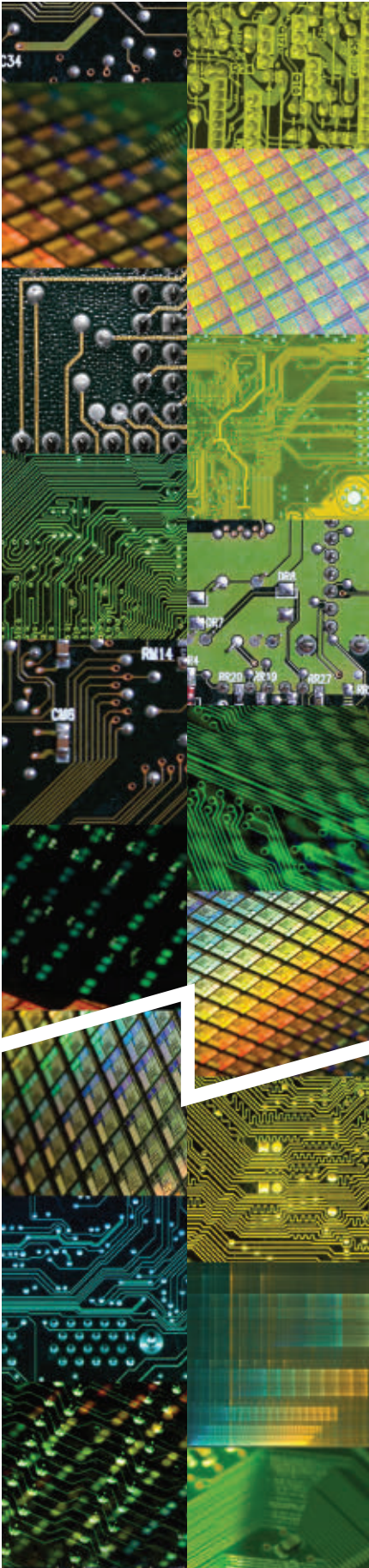
for domain-specific languages where this load is somewhat reduced. It is also worth noting that during the last decade innovation in high-end general-purpose desktop applications has been minimal, perhaps with the exception of computer graphics; this is especially conspicuous in comparison to mobile and Internet applications. Moreover, contrasting 2005 predictions by vendors such as Intel² that mainstream processors will have several hundred cores (“paradigm shift to a many-core era”) by 2013 with the reality of around a handful; perhaps the reason was that the diminished competition among general-purpose desktop vendors in that timeframe did not contribute to their motivation to take the risks that such a paradigm shift entails. But, does this mean the hardware of these computers is broken?

I believe the answer is yes. Many more application domains could significantly benefit from exploiting irregular parallelism for speedups. A partial list of such domains follows.

- ▶ *Bioinformatics.* Genomics involves many graph and other combinatorial problems that are rarely regular.

- ▶ *Computer vision.* Only a fraction of the OpenCV library is supported well on graphics processing units (GPUs). Other OpenCV primitives are typically irregular.

- ▶ *Scientific computing.* Sparse matri-



ces and problems that require runtime adaptation such as multiscale methods for solving linear equations, or for kinetic modeling of quantum systems.

- ▶ Data compression.
- ▶ Sparse sensing and recovery in signal processing.
- ▶ Electronic design automation (EDA) for both design and simulations.
- ▶ Data assimilation; and
- ▶ The numerous uses of graph models:
 - ▶ Data analytics.
 - ▶ Analysis of social networks.
 - ▶ Epidemiological networks.
 - ▶ Belief propagation.
- ▶ *Open-ended problems and programming.* This domain is characterized by the way problems may be posed and computer programs developed rather than by a particular application. Many problems whose initial understanding does not imply clear output, or sometimes even input, definitions lend themselves to irregular programming.

Today's general-purpose multicore hardware does not provide sufficient support for any of these domains. It must be fixed to support more of them, and to allow cost-effective parallel programming. A fix will require changes both to current hardware, and to the overall ecological system comprising them, including programming practice and compilers. The lead questions for such a system are: how should programmers express the parallelism in the algorithms they implement; what will the responsibility of compilers and hardware/software runtime systems be; and how will the hardware execute the eventual ready-to-run parallelism as efficiently as possible. I believe that for irregular problems: programmers are good at seeing parallelism (that is, understand which operations can be done concurrently) well beyond what current and near-future compiler technology can extract from serial code; much of this parallelism is fine-grained; and this parallelism can vary from very scarce (very few operations that can be executed concurrently) to plenty (many such operations), even among steps of the same parallel algorithm. Thus, system designers must “keep their eye on the ball” by viewing the parallelism that programmers provide as perhaps their *most precious resource*. Specifically: encour-

age programmers to express all the parallelism they see (of course, after facilitating such expression in programming languages), and optimize compiler, runtime systems, and hardware to derive the best performance possible from whatever amount of parallelism programmers provide. In particular, the overheads for translating any amount of parallelism coming from the program to performance must be minimized.

Examples for what could be done include shared (on-chip) caches, low-overhead control mechanisms, high bandwidth, low latency interconnection networks, and flexible data fetch and store mechanisms. Many would be justifiably puzzled by this list and ask: Aren't we already aware of these measures? Aren't they already being implemented? Closer scrutiny would suggest the incorporation of these measures has been greatly diluted by competing considerations (not keeping one's eye on the ball), leading to the current chasm between parallel architectures and irregular parallel algorithms. Several examples of competing objectives that hurt multicore performance on irregular parallel programs follow.

Competing objective: *Improve throughput.* Current multiprocessors consist of tightly coupled processors whose coordination and usage are controlled by a single operating system (OS). This has been a common approach when the objective of the architecture is to maximize the number of (possibly small) jobs to be completed in a given time. (Sometimes this objective is referred to as optimizing throughput.) Being a software system, the OS encounters unnecessarily high overhead for thread management, including: thread initiation, dynamic allocation of threads to hardware, and thread termination. However, had the main objective been low overhead support for irregular parallel programs, we would have seen migration of more of these functions to hardware, especially for *fine-grained programs as many irregular programs are*.

Competing objective: *Maximize peak performance.* Designs of GPUs seem to have been guided by fitting as many functional units as possible within a silicon budget in an attempt to maximize peak performance (for example,

FLOPs). But, effective support of irregular programs, strong scaling, and better sustained (rather than peak) performance is a different objective requiring different choices to be reflected both on the hardware side and on the programmer side. Examples for the hardware side: the data a functional unit needs to process cannot be generally assumed to be available near the functional unit, or when vector functional units are used, data needs to be provided to them both at a high rate and in a structured way, but this cannot be generally assumed, as well. Simple parallel programming is also not compatible with expecting the programmer to work around such data feeds. There are quite a few examples of how GPUs require data to be structured in a very rigid way by the programmer.

Competing objective: *Maximize locality.* The respective roles that caches have come to play in serial and parallel architectures help explain at least part of the problem.

Caches in serial architectures. Serial computing started with a successful general-purpose programming model. As improvement in memory latency started falling behind improvement in serial processor speed during the 1980s, caches emerged as the solution for continued support of the serial programming model. The observation that serial programs tend to reuse data (or nearby addresses of data recently used), also known as the “principle of locality,” meant caches could generally mitigate problems with continued support of that model. Thus, while locality has become a major theme for optimizing serial hardware, it was not allowed to interfere with the basic programming model of everyday programming; even when programming for locality was done, it was by relatively few “performance programmers.”

Local parallel memories. Parallel computing has never enjoyed a truly successful general-purpose programming model, so there was no sufficient motivation to invest in continued support of one. Parallel processing architectures have been driven since their early days by the coupling of each processor with a considerable local memory component. One key reason, already noted earlier, was the quest for higher peak performance counting FLOPs. This

A quest for future progress must begin with a critical look at some of the current shortcomings of parallel computing.

meant maximizing the number of functional units and their nearby memory within a given dollar budget, silicon-area budget, or power budget. I also noted that trading off some of these budgets for improved sustained performance, perhaps at the expense of peak performance, appears to have been a lower priority. Thus, mapping parallel tasks to these local memories has become an enormous liability for the programmer, and one of the main obstacles for extending the outreach of parallel processing beyond regular applications and for making parallel programming simpler.

Competing objective: *Prioritize highly parallel applications.* Current designs seem to expect a very high amount of parallelism to come from applications. I believe that, in general, this view is too optimistic. One lesson of serial computer architecture has been the need to put any general-purpose hardware platform through nontrivial benchmarking stress tests. Downscaling of parallelism on current parallel machines is often a problem for irregular algorithms and problems. For example, in breadth-first search on graphs, some problem instances may provide a large amount of parallelism (for example, random graphs) while other instances do not (for example, high-diameter graphs). Some algorithms operate such that the parallelism in different steps of the algorithm is drastically different. For example, standard max-flow algorithms provide a useful stress test. These max-flow algorithms iterate breadth-first search on graphs of increasing diameter, and therefore their parallelism decreases as the algorithm progresses, failing architec-

tures that can handle well only a high level of parallelism.

Competing objective: *Prioritize energy saving over programmer's productivity.* Approaching the end of the so-called Dennard scaling and the decreasing improvement in power consumption of computers it implies are important concerns.⁵ Power consumption is also easier to quantify than programmer's productivity and is closer to the comfort zone of hardware designers. This may explain the often heard sentiment that parallel programmers must take on themselves programming for reducing energy consumption, which found its way into some design decisions. I see two problems with this trend, one is rather concrete and the other is more principled. The concrete problem is that irregular problems make it much more difficult, if not impossible, for programmers to conform with these design decisions. The general problem is that the basic sentiment seems to “go against history.” Much of the progress attributed to the Industrial Revolution is due to using more power for reducing human effort. Can future progress in computing performance be based on reversing this trend?

The reader needs to be aware that this approach of questioning vendors' hardware presented here is far from unanimous. In fact, many authors have sought conformity with such hardware, modeling limitations for meeting them in algorithm design. Bulk-synchronous parallelism (BSP),⁶ and more recently quite a few communication-avoiding algorithms, such as Ballard et al.,¹ are notable examples for considerable accomplishments regarding regular algorithms. However, the state of the art remains that unless problem instances can be mapped to dense matrix structure, they cannot be solved efficiently, and after many years of parallel algorithms research I do not believe such a fundamental change in reality is feasible.

Interestingly, the chasm between this communication-avoiding school of thought and the position of this Viewpoint is not as large as it may appear. Before explaining why, I point out that historically government investment in parallel computing has been mostly tied to the bleeding edge of large high-performance computers, driven

advent of 3D-VLSI technology, along with its potential accommodation of greater heterogeneity in hardware, may allow vendors to add new components without removing support for current programming models.

A reviewer of an earlier version of this Viewpoint challenged its basic thrust with the following interesting argument. Since the sole point of parallel computing is performance, every parallel programmer is by definition a performance programmer. Thus, the effort of parallel programming should be judged by the standards of performance (serial) programming, which is also considerably more demanding than just standard serial programming. My answer is that getting good (though not the best) serial performance, as students enrolled in CS courses often manage to get and without the need to retune their code for new machine generations, should be the proper effort standard for getting significant (though not the best) parallel speedups. In other words:

► Performance in serial programming often comes through the optimizations made available by compilers, freeing the programmer from much of the burden of fine-tuning the code for performance. Such optimizations allow keeping more of the programmer's focus on getting performance by bettering the algorithm, which is not the case in parallel programming.

► The programmer can only do so much when the hardware stands in his or her way. Current computer architectures are geared toward multiple streams of serial codes (threads). For performance, a programmer is required to come up with threads that are large enough, which is something not readily available in irregular programs.

Still, this reviewer's comment and my answer suggest this Viewpoint must also demonstrate I am not dreaming, and a multicore system allowing good speedups on irregular problems and algorithms with limited effort is indeed feasible. For this reason the current version cites the XMT^a many-core computer platform described in Vish-

a XMT stands for explicit multithreading and should not be confused with the generation of the Tera computer project, which is called Cray XMT.

The world has yet to see a commercial parallel machine that can handle general-purpose programming and applications effectively.

kin,⁷ which provides a useful demonstration. For example:

► Students in the graduate parallel algorithms theory class I teach at the University of Maryland are required to complete five or six nontrivial parallel programming assignments, and nearly all manage to get significant speedups over their best serial version, in every assignment.

► Nearly 300 high school students, mostly from the Thomas Jefferson High School for Science and Technology in Alexandria, VA, have already programmed XMT achieving significant speedups.

► The XMT home page^b also cites success on par, or nearly on par, with serial computing with students in middle school, an inner-city high school, and college freshmen and other undergraduate students.

► As the max-flow problem was mentioned, the XMT home page also cites a publication demonstrating speedups of over 100X over the best serial algorithm counting cycles, while speedups on any commercial system do not exceed 2.5X.

► Publications demonstrating similar XMT speedups for some of the other advanced parallel algorithms in the literature are also cited.

► XMT also demonstrates there is no conflict with backward compatibility on serial code.

The world has yet to see a commercial parallel machine that can handle general-purpose programming and applications effectively. It is up to the research community to vertically develop an integrated computing stack, and

b The XMT home page is <http://www.umiacs.umd.edu/~vishkin/XMT/>.

prototype and validate it with significant applications and easier parallel programming, trailblazing the way for vendors to follow. Due to its high level of risk, prototype development fits best within the research community. On the other hand, exploiting parallelism for today's commercial systems is of direct interest to industry. If indeed application development for current commercial systems will be funded by industry, more of today's scarce research funding could shift toward prototype development where it is needed most.

Ludwik Fleck, a Polish-Israeli founder of the field of sociology of science, observed the discourse of research communities is not without problems, pointing out that even the most basic consensus of such a community (for example, what constitutes a fact) merits questioning.⁴ In particular, through feedback external to the community reaching consensus. This Viewpoint seeks to provide such feedback for general-purpose multicore parallelism. Its ideal impact would be driving the field toward seeking enablement of systemic advancement that will get the field out of its "rabbit hole" of limited-potential well-worn paths and ad hoc solutions, as significant and successful as these paths and solutions have been. ■

References

- Ballard, G. et al. Communication efficient Gaussian elimination with partial pivoting using a shape morphing data layout. In *Proceedings of the 25th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, (Montreal, Canada, 2013), 232–240.
- Borkar, S.Y. et al. Platform 2015: Intel processor and platform evolution for the next decade. White Paper, Intel Corporation, 2005.
- Edwards, J.A. and Vishkin, U. Parallel algorithms for Burrows-Wheeler compression and decompression. *Theoretical Computer Science*, to appear in 2014; see <http://dx.doi.org/10.1016/j.tcs.2013.10.009>.
- Fleck, L. *The Genesis and Development of a Scientific Fact*, (edited by T.J. Trenn and R.K. Merton, foreword by Thomas Kuhn). University of Chicago Press, 1979. English translation of *Entstehung und Entwicklung einer wissenschaftlichen Tatsache. Einführung in die Lehre vom Denkstil und Denkkollektiv* Schwabe und Co., Verlagsbuchhandlung, Basel, 1935.
- Fuller, S.H. and Millet, L.I., Eds. *The Future of Computing Performance: Game Over or Next Level*. National Research Council of the National Academies, The National Academies Press, 2011.
- Valiant, L.G. A bridging model for parallel computation. *Commun. ACM* 33, 8 (Aug. 1990), 103–111.
- Vishkin, U. Using simple abstraction to reinvent computing for parallelism. *Commun. ACM* 54, 1 (Jan. 2011), 75–85.

Uzi Vishkin (vishkin@umd.edu) is a professor in the Department of Electrical and Computer Engineering at The University of Maryland, College Park, and at the University of Maryland Institute for Advanced Computer Studies.

Partially supported by awards CNS-1161857 CCF-0811504 from the National Science Foundation.

Copyright held by Author/Owner(s).

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

The edge of the Internet is an unruly place.

BY PAUL VIXIE

Rate-Limiting State

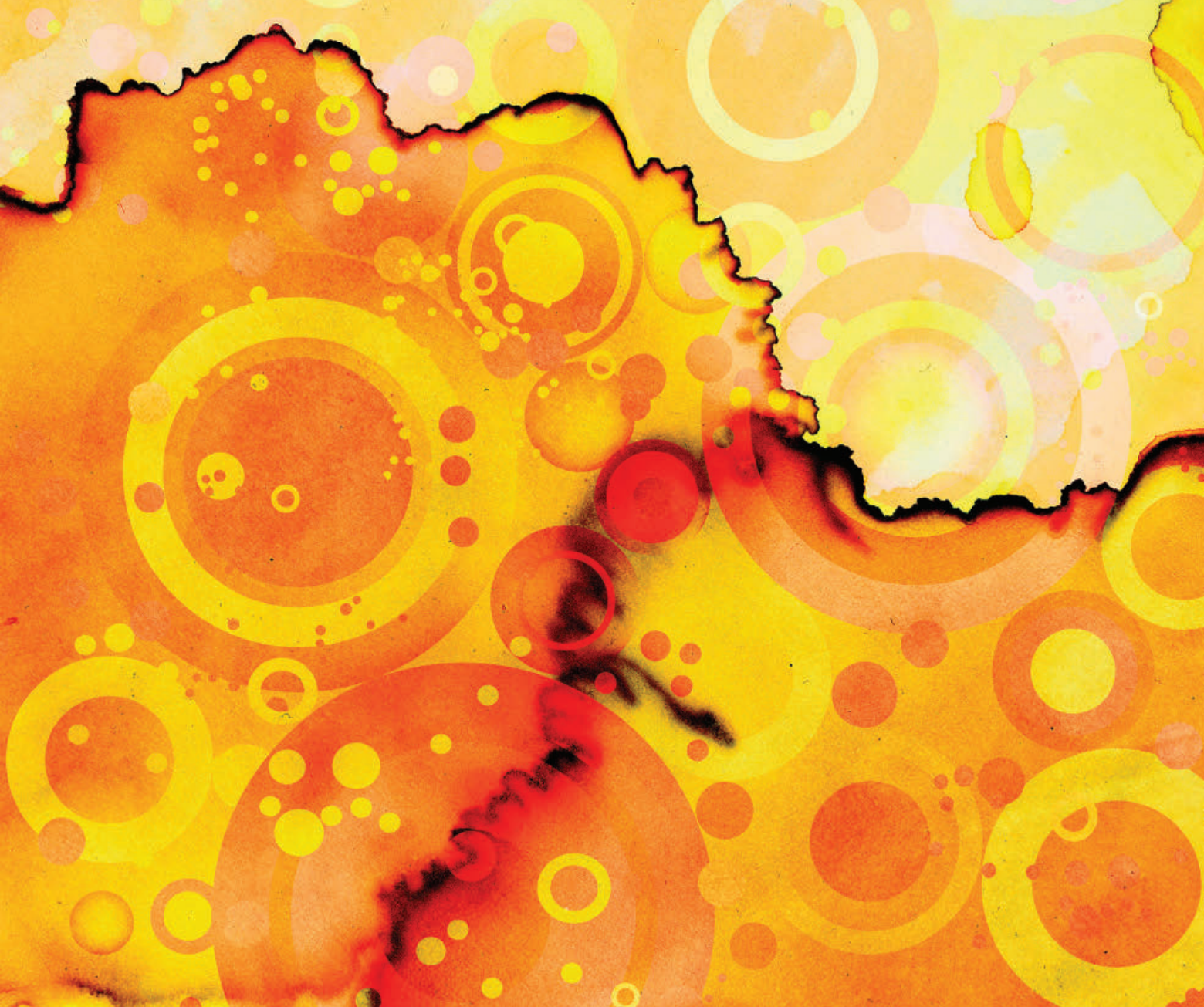
BY DESIGN, THE Internet *core* is stupid, and the *edge* is smart. This design decision has enabled the Internet's wildcat growth, since without complexity the core can grow at the speed of demand. On the downside, the decision to put all smartness at the edge means we are at the mercy of scale when it comes to the quality of the Internet's aggregate traffic load. Not all device and software builders have the skills—and the quality assurance budgets—that something the size of the Internet deserves. Furthermore, the resiliency of the Internet means a device or program that gets something importantly wrong about Internet communication stands a pretty good chance of working “well enough” in spite of its failings.

Witness the hundreds of millions of customer-premises equipment (CPE) boxes with literally too much memory for buffering packets. As Jim Gettys and Dave Taht have been demonstrating in recent years, more is not better when it comes to packet memory.¹ Wireless networks in homes and coffee shops and businesses all degrade shockingly when the traffic load increases. Rather than the “fair-share” scheduling we expect, where N network flows will each get roughly $1/N^{\text{th}}$ of the available bandwidth, network flows end up in quicksand where they each get $1/(N^2)$ of the available bandwidth. This is not because CPE designers are incompetent; rather, it is because the Internet is a big place with a lot of subtle interactions that depend on every device and software designer having the same—largely undocumented—assumptions.

Witness the endless stream of patches and vulnerability announcements from the vendor of literally every smartphone, laptop, or desktop operating system and application. Bad guys have the time, skills, and motivation to study edge devices for weaknesses, and they are finding as many weaknesses as they need to inject malicious code into our precious devices where they can then copy our data, modify our installed software, spy on us, and steal our identities—113 years of science fiction has not begun to prepare us for how vulnerable we and our livelihoods are, now that everyone is online. Since the adversaries of freedom and privacy now include nation-states, the extreme vulnerability of edge devices and their software is a fresh new universal human-rights problem for the whole world.

Source Address Validation

Nowhere in the basic architecture of the Internet is there a more hideous flaw than in the lack of most gateways to enforce simple source-address validation (SAV). Because the Internet works well enough even without SAV, and because the Internet's roots are in academia where there were



no untrusted users or devices, it is safe to say most gateway makers (for example, wireless routers, DSL modems, and other forms of CPE) will allow most edge devices to emit Internet packets claiming to be from just about anywhere. Worse still, providers of business-grade Internet connections, and operators of Internet hosting data centers and “clouds” are mostly not bothering to turn on SAV toward their customers. Reasons include higher cost of operation (since SAV burns some energy and requires extra training and monitoring), but the big reason why SAV is not the default is: SAV benefits only other people’s customers, not an operator’s own customers.

There is no way to audit a network from outside to determine if it prac-

tices SAV. Any kind of compliance testing for SAV must be done by a device inside the network whose compliance is in question. That means the same network operator who has no incentive in the first place to deploy SAV at all is the only party who can tell whether SAV is deployed. This does not bode well for a general improvement in SAV conditions, even if bolstered by law or treaty. It could become an insurance and audit requirement in countries where insurance and auditing are common, but as long as most of the world has no reason to care about SAV, it is safe to assume enough of the Internet’s edge will always permit packet-level source-address forgery, so we had better start learning how to live with it—for all eternity.

While there are some interesting problems in data poisoning made possible by the lack of SAV, by far the most dangerous thing about packet forgery is the way it facilitates DDoS (distributed denial of service).³ If anybody can emit a packet claiming to be from anybody else, then a modest stream of requests by an attacker, forged to appear to have come from the victim, directed at publicly reachable and massively powerful Internet servers, will cause that victim to drown in responses to requests they never made. Worse, the victim cannot trace the attack back to where it entered the network and has no recourse other than to wait for the attack to end, or hire a powerful network-security vendor to absorb the attack so the victim’s other services remain reachable during the attack.²

IMAGE FROM SHUTTERSTOCK.COM

Domain Name System Response Rate Limiting

During a wave of attacks several years ago where massively powerful public DNS (Domain Name System) servers were being used to reflect and amplify some very potent DDoS attacks, Internet researchers Paul Vixie and Vernon Schryver developed a system called Response Rate Limiting (DNS RRL) that allowed the operators of the DNS servers being used for these reflected amplified attacks to deliberately drop the subset of their input request flow that was statistically likely to be attack-related.⁴ DNS RRL is not a perfect solution, since it can cause slight delays in a minority of normal (non-attack) transactions during attack conditions. The DNS RRL trade-off, however, is obviously considered a positive since all modern DNS servers and even a few intrusion protection system/intrusion detection system (IPS/IDS) products now have some form of DNS RRL, and many top-level domain (TLD) DNS servers are running DNS RRL. Operators of powerful Internet servers must all learn and follow Stan Lee's law (as voiced by Spider-Man): "With great power comes great responsibility."

DNS RRL was a domain-specific solution, relying on detailed knowledge of DNS itself. For example, the reason DNS RRL is *response* rate limiting is the mere fact of a question's arrival does not tell the rate limiter enough to make a decision as to whether that request is or is not likely to be part of an attack. Given also a prospective response, though, it is possible with high confidence to detect spoofed-source questions and thereby reduce the utility of the DNS server as a reflecting DDoS amplifier, while still providing "good enough" service to non-attack traffic occurring at the same time—even if that non-attack traffic is very similar to the attack.

The economics of information warfare is no different from any other kind of warfare—one seeks to defend at a lower cost than the attacker, and to attack at a lower cost than the defender. DNS RRL did not have to be perfect; it merely had to tip the balance: to make a DNS server less attractive to an attacker than the attacker's alternatives. One important principle of DNS RRL's design is that it makes a DNS server

Bad guys have the time, skills, and motivation to study edge devices for weaknesses, and they are finding as many weaknesses as they need to inject malicious code into our precious devices.

into a DDoS *attenuator*—it causes not just lack of amplification, but also an actual reduction in traffic volume compared with what an attacker could achieve by sending the packets directly. Just as importantly, this attenuation is not only in the number of bits per second, but also in the number of packets per second. That is important in a world full of complex stateful firewalls where the bottleneck is often in the number of packets, not bits, and processing a small packet costs just as much in terms of firewall capacity as processing a larger packet.

Another important design criterion for DNS RRL is that its running costs are so low as to be not worth measuring. The amount of CPU capacity, memory bandwidth, and memory storage used by DNS RRL is such a small percentage of the overall load on a DNS server that there is no way an attacker can somehow "overflow" a DNS server's RRL capacity in order to make DNS RRL unattractive to that server's operator. Again, war is a form of applied economics, and the design of DNS RRL specifically limits the cost of defense to a fraction of *a fraction* of the attacker's costs. Whereas DNS achieves its magnificent performance and scalability by being stateless, DNS RRL adds the minimum amount of state to DNS required for preventing reflected amplified attacks, without diminishing DNS's performance.

Current State

To be stateless in the context of network protocols means simply that the responder does not have to remember anything about a requester in between requests. Every request is complete unto itself. For DNS this means a request comes in and a response goes out in one single round-trip from the requester to the responder and back. Optional responder state is not prohibited—for example, DNS RRL adds some modest state to help differentiate attack from non-attack packets. Requesters can also hold optional state such as RTT (round-trip time) of each candidate server, thus guiding future transactions toward the server that can respond most quickly. In DNS all such state is optional, however, and the protocol itself will work just fine even if nobody on either end retains any state at all.

DNS is an example of a User Datagram Protocol (UDP), and there are other such protocols. For example, Network Time Protocol (NTP) uses UDP, and each response is of equal or greater size than the request. A true NTP client holds some state, in order to keep track of what time the Internet thinks it is. An attacker, however, need not show an NTP responder any evidence of such state in order to solicit a response. Since NTP is often built into CPE gateways and other edge devices, there are many millions of responders available for DDoS attackers to use as reflectors or as amplifying reflectors.

Transmission Control Protocol (TCP) is, on the other hand, stateful. In current designs both the initiator and the responder must remember something about the other side; otherwise, communication is not possible. This statefulness is a mixed blessing. It is burdensome in that it takes several round-trips to establish enough connection state on both sides to make it possible to send a request and receive a response, and then another one-and-a-half round-trips to close down the connection and release all state on both sides. TCP has an initiation period when it is trying to create shared state between the endpoints, during which several SYN-ACK messages can be sent by the responder to the purported initiator of a single SYN message. This means TCP itself can be used as an amplifier of bits and packets, even though the SYN-ACK messages are not sent back to back. With hundreds of millions of TCP responders available, DDoS attackers can easily find all the reflecting amplifying TCP devices needed for any attack on any victim no matter how capacious or well-defended.

Internet Control Message Protocol (ICMP) is stateless, in that gateways and responders transmit messages back to initiators in asynchronous response to network conditions and initiator behavior. The popular “ping” and “traceroute” commands rely on the wide availability of ICMP; thus, it is uncommon for firewalls to block ICMP. Every Internet gateway and host supports ICMP in some form, so ICMP-based reflective DDoS attackers can find as many ICMP reflectors as they look for.

The running theme of these observations is that in the absence of SAV, statelessness is bad. Many other UDP-based protocols, including Server Message Block (SMB) and NFS, are stateful when used correctly, but, like TCP, are stateless during initial connection startup and can thus be used as DDoS reflectors or amplifying DDoS reflectors depending on the skill level of a DDoS attacker. While the ultimate cause of all this trouble is the permanent lack of universal SAV, the proximate cause is stateless protocols. Clearly, in order to live in a world without SAV, the Internet and every protocol and every system is going to need more state. That state will not come to the Internet core, which will be forever dumb. Rather, the state that must be added to the Internet system in order to cope without SAV has to be added at the Internet edge.

Conclusion

Every reflection-friendly protocol mentioned in this article is going to have to learn rate limiting. This includes the initial TCP three-way handshake, ICMP, and every UDP-based protocol. In rare instances it is possible to limit one’s participation in DDoS reflection and/or amplification with a firewall, but most firewalls are either stateless themselves, or their statefulness is so weak it can be attacked separately. The more common case will be like DNS RRL, where deep knowledge of the protocol is necessary for a correctly engineered rate-limiting solution applicable to the protocol. Engineering economics requires the cost in CPU, memory bandwidth, and memory storage of any new state added for rate limiting be insignificant compared with an attacker’s effort. Attenuation also has to be a first-order goal—we must make it more attractive for attackers to send their packets directly to their victims than to bounce them off a DDoS attenuator.

This effort will require massive investment and many years. It is far more expensive than SAV would be; yet SAV is completely impractical because of its asymmetric incentives. Universal protocol-aware rate limiting (in the style of DNS RRL, but meant for every other presently stateless interaction on

the Internet) has the singular advantage of an incentive model where the people who would have to do the work are actually motivated to do the work. This effort is the inevitable cost of the Internet’s “dumb core, smart edge” model and Postel’s Law (“be conservative in what you do, be liberal in what you accept from others”).

Reflective and amplified DDoS attacks have steadily risen as the size of the Internet population has grown. The incentives for DDoS improve every time more victims depend on the Internet in new ways, whereas the cost of launching a DDoS attack goes down every time more innovators add more smart devices to the edge of the Internet. There is no way to make SAV common enough to matter, nor is there any way to measure or audit compliance centrally if SAV somehow were miraculously to become an enforceable requirement.

DDoS will continue to increase until the Internet is so congested the benefit to an attacker of adding one more DDoS reaches the noise level, which means, until all of us including the attackers are drowning in noise. Alternatively, rate-limiting state can be added to every currently stateless protocol, service, and device on the Internet. ■

Related articles on queue.acm.org

DNS Complexity

Paul Vixie

<http://queue.acm.org/detail.cfm?id=1242499>

Broadcast Messaging: Messaging to the Masses

Frank Jania

<http://queue.acm.org/detail.cfm?id=966719>

Lessons from the Letter

George V. Neville-Neil

<http://queue.acm.org/detail.cfm?id=1837255>

References

1. Bufferbloat; <http://www.bufferbloat.net/>.
2. Defense.net; <http://defense.net/>.
3. Vixie, P. Securing the edge, 2002; <http://archive.icann.org/en/committees/security/sac004.txt>.
4. Vixie, P. and Schryver, V. Response rate limiting in the Domain Name System, 2012; <http://www.redbarn.org/dns/ratelimits>.

Paul Vixie is the CEO of Farsight Security. He previously served as president, chairman, and founder of Internet Systems Consortium (ISC); president of MAPS, PAIX, and MIBH; and CTO of Abovenet/MFN. Vixie is a founding member of ICANN RSSAC (Root Server System Advisory Committee) and ICANN SSAC (Security and Stability Advisory Committee).

© 2014 ACM 0001-0782/14/04 \$15.00

Article development led by **acmqueue**
queue.acm.org

Becoming better, faster, cheaper, and happier.

BY IVAR JACOBSON, PAN-WEI-NG,
IAN SPENCE, AND PAUL E. MCMAHON

Major-League SEMAT— Why Should an Executive Care?

IN TODAY'S EVER more competitive world, boards of directors and executives demand that CIOs and their teams deliver “more with less.” Studies show, without any real surprise, that there is no one-size-fits-all method to suit all software initiatives, and that a practice-based approach with some light but effective degree of order and governance is the goal of most software-development departments.

Software Engineering Method and Theory (SEMAT) is a collaborative initiative whose mission is to provide software engineering with a foundation, based on a kernel of practice-independent elements, supported by a language that allows best practices to be described

in as light or as detailed a way as needed. These can then be selected by teams for the context of their endeavor, thus ensuring the organization neither falls into a practice free-for-all nor constrains its business into a process straitjacket.

Executives have reason to care about SEMAT. The initiative supports the goals of “more with less” by delivering value at the team and organizational levels. Initiatives will always remain at a theoretical level unless and until proven through experimentation, and the case for SEMAT is strongly supported through real-life case studies.

According to Gartner's “Top 10 CIO Business and Technology Priorities in 2014,” CIO imperatives are split between business enablement and efficiency, reflecting the importance of the CIO's contribution in both these areas of the organization.¹ This is obviously a tough challenge—and the phrase “getting more with less” is often heard in conversations with IT executives and their leadership teams. What can be done to gain a real effect, and how can the software functions of major-league IT organizations react?

In support of this challenge, respondents to a survey conducted at the 2014 Gartner Application Architecture, Development, and Integration (AADI) Summit pointed out that trying to do more with less by attempting to standardize around a single development process was not the answer.⁷ The results reflect that no “silver bullet” universal method has been found (no surprise there!) and that a software endeavor needs to use the most effective practices based on its particular context. The respondents went on to suggest that at a foundational level, ensuring an ability to measure the effectiveness of endeavors end to end was a key requirement across all of applications development.

The survey acknowledged that while the agile movement has helped accelerate development, respondents and their CIO leadership are still looking for some degree of rigor, structure,



• MORE WITH LESS •

and governance in order to align better with business, behave more predictably, measure and address risk, and finally, deliver business value better, faster, and cheaper with happier stakeholders—the BFCH metrics.

Of course, “being agile” can be said to be a goal for any software endeavor, or any software organization (after all, who would not want to be agile?). It is not necessarily the case, however, that the same set of agile practices can be applied universally across all programs and teams. Indeed, in some endeavors, particular agile techniques or practices may not be possible. Take, for example, the case of a bank’s portfolio of programs. At any time there is huge diversity across many dimensions: pure development projects require a different approach from maintenance projects; highly regulated software initiatives such as financial clearing may require more rigorous requirements documentation than new consumer-driven, graphical front-end development; and simple stand-alone apps may require a radically faster cadence than architectural changes such as the replacement of the enterprise application server. This, therefore, gives rise to the need for multiple practice alternatives within a single organization.

The conclusions of the AADI Summit survey—that a standard one-size-fits-all process should not be a 2014 priority and that some order and governance is necessary to prevent what could become a practice free-for-all if left unchecked—certainly make sense when considering the reality of a large organization’s application-development landscape, as in the bank example.

The SEMAT initiative supports the findings of the AADI Summit survey, as well as the more general CIO goal of “more with less” through delivery of value at both the organizational and team levels.

Introducing SEMAT

Since 2010, experts from industry and academia have collaborated under the SEMAT banner in an explicit attempt to “refund software engineering based on a solid theory, proven principles, and best practices.” As a first step on this journey they have created Essence, a kernel and language for

the improvement of software-development methods, which is soon to be ratified as a standard by Object Management Group (OMG).⁵ It will be clear that the standard is an underpinning of the various practices available in the industry—agile or not.

The SEMAT initiative and the Essence kernel enable organizations to establish an environment where they can make use of the right practices for the right endeavors and right contexts. The practices are built on an independent, solid foundation (the kernel) that incorporates a lightweight yet appropriate level of order and measurement for the business. This approach represents a first of its kind in software engineering.

The Essence kernel includes “things we always have” and “things we always do” when developing software. It is universal to all software-development endeavors—agile or pre-agile. SEMAT has adopted several fundamentally new ideas. One of them is called the *alpha* concept, which allows teams to focus on the *doing* rather than on the *describing*. Teams can measure the progress and health of their endeavors in a practice- or method-independent way. At any moment the team can identify where it is now and where it is going next. It makes the team results-focused instead of document-driven or activity-centric. The alpha concept supports several other ideas—for example, the idea of a method being a composition of practices built on top of the universal kernel. Thanks to the alpha concept, SEMAT has been able to create a robust, extensible, intuitive kernel.

More complete descriptions of SEMAT are available in previous writings.^{2,3}

The Value of Semat for Large Organizations

It is a fact the software world has a huge number of methods—perhaps more than 100,000 have been developed, although many have not been used more than once. Some are known by brands or labels such as Rational Unified Process (RUP), Scrum, and Extreme Programming (XP), but most are homegrown with ideas picked up from textbooks or other resources, with zero opportunity for reuse outside of

the single context for which they were built. The development of SEMAT has demonstrated that underneath all of these methods is a common ground, or a *kernel*, of “things we always have” and “things we always do” when developing software. These things form the essence of software engineering and are now included in Essence.⁵

As discussed earlier, large organizations have a diverse range of projects carried out by a diverse range of people. Without an accepted kernel to share, coordinating development endeavors is unnecessarily complicated, because teams must spend scarce time and attention on deciding how to work at the start of every project. This results in significant wasted effort, project delays, frustrations among developers, and dissatisfied customers. All of these factors have the potential to compromise that all-important CIO priority of achieving “more with less.” The inefficiencies occur both at the project/team (endeavor) level and the organizational level. To quantify this, consider the value it could bring to an organization if the development team did not need to come up with a specific new way of working, but instead could start from a common ground and then add existing well-proven (possibly agile or lean) ideas published in a library of practices.

The Essence kernel has benefits at both the team and organizational levels. To make this discussion something readers can identify with, let’s introduce two characters: Smith, a project leader; and Dave, an executive in a large company that has been trying to improve the way it develops software. The company had invested in and achieved a Capability Maturity Model Integration (CMMI) rating, then adopted the Unified Process, and recently experimented with and, in some teams, successfully applied agile methods.

Grass-Roots Level—Ensuring the Team’s Success

At the team level, the objective is to ensure the success of the endeavor, which Smith is responsible for. Success means delivering what his customers really want in the given time frame and budget, and with the right quality. Suc-

cess also means his team is motivated and collaborating effectively—fundamentally, a happy team. This requires four skills: using the right method for the right job; measuring project progress and health; effective collaboration; and decision-making based on a middle ground.

Using the right method for the job at hand. The battle-seasoned Smith understands (as did the Gartner AADI survey respondents) there is no one-size-fits-all, future-proofed software-development method. As mentioned, Smith's organization had adopted CMMI, the Unified Process, and agile methods. Smith is not against any of these approaches and, in fact, thinks there is something good in each one of them. The problem is that, in the past, his organization at any point in time mandated just one way of running development.

Although Smith's colleague who runs small enhancements found Scrum to be useful, Smith found Scrum by itself to be inadequate for the large-scale enterprise development he is running. Smith was also mindful that his organization's previous CMMI- and Unified Process-based approaches were also inadequate. In each project, Smith still had to make significant adjustments to the mandated way of working to achieve a suitable approach. In effect, he was adding to the "100,000-methods problem" by creating his own variant. Making and establishing such adjustments was no small feat, as Smith had to communicate them to his teammates, all of whom had different backgrounds and opinions. Smith wanted a better approach for his teammates to come quickly to agreement on how they should work on each project.

The lightweight kernel and its support for systematically combining practices to create a variety of "just enough" methods, all based on the same common kernel, allows organizations to provide development teams with the right methods to undertake each job at hand, supporting each organization's specific culture, standards, and mandated practices.

Ability to measure project progress and health accurately. If Smith has learned anything, it is that software development is a complex endeavor,



The SEMAT initiative and the Essence kernel enable organizations to establish an environment where they can make use of the right practices for the right endeavors and right contexts.



and the progress and health of a project must be considered from multiple perspectives—for example, stakeholders' satisfaction; requirements (and requirements grow as the software system evolves); software system; and so on. The problem, however, is that Smith's company does not have governance procedures that holistically help each team evaluate progress and health. In fact, no commonly accepted framework has previously been able to do this.

Each alpha (that is, opportunity, stakeholders, requirements, software system, team, work, and way of working) covers a dimension, and the states defined for each alpha provide an indication of progress and health. For example, Essence identifies states of the software system as being architecture selected, demonstrable, usable, ready, operational, and retired. Each of these states provides an indication of how far the development of the software system has come, and what a team should do next to move the progress and health of the software system forward. Modern approaches exist that recommend a similar way of thinking but only in a single dimension (for example, Kanban focuses on the work alpha). Essence, on the other hand, is more comprehensive and holistic, since it is multidimensional (that is, it focuses on all the alphas), and thus gives a more accurate picture of progress and health. What's more, Essence does this in a lightweight manner, unlike previous attempts such as within CMMI and the Unified Process.


Ability to get team members to collaborate effectively. Each time Smith begins a project, he has to assemble a team with members from different parts of the organization, and possibly even from different contractor organizations. The team members need to agree on how they should collaborate. For example, they need to agree on which practices to use. Because they have different backgrounds and experiences, this often turns out to be nontrivial. Preparing team members can be as complex as a reengineering effort. As further evidence, in much of the literature published over the past 20 years on the causes of failed projects, poor communication

among team members frequently ranks very high.⁶


Individuals from different organizations have a common base from which to start. This helps the team communicate more effectively and therefore become productive more rapidly, even when their native or favorite ways of working are significantly different.

Decision-making based on a middle ground. The basis for decision-making should be quick health checks rather than inflexible processes—or nothing at all. An integral part of Smith's job is deciding with his teammates where to place their efforts to achieve progress and deal with project risks. Some requirements might be vague and need further exploration with customer representatives; some technical issues might need investigation; some defects might need attention. Previously, however, Smith had to work within prescribed governance procedures that required the production of specific documents at specific milestones within the project and specific activities to be conducted in a specific order. This was a heavyweight, ineffective approach that did not help Smith and his teammates make relevant decisions. With Essence, however, the focus is on the results generated and dealing with risks within the project's context.

Essence provides a framework that enables faster, more effective, more autonomous decision-making and can help organizations evolve their current governance practices to meet the needs of present-day challenges. One example of how Essence accomplishes this is through the alphas, alpha states, and checklists that keep the team focused on what is most important at any point in the project, based on the current project context. That provides an independent check of how well the team is doing at achieving the intended result. This check can be applied regardless of the method and practices the team is using. Today, we understand far better the importance of each project's context when it comes to managing cost and performance effectively. This is where Essence can be a great differentiator from previous approaches, because it helps ask the right question at the right time based on the current state and the target goal state.



Visualize a library of interchangeable practices, written using a language that can be as light or as formal as the situation requires, and built around the fact that all software endeavors are successful based on progression of certain key elements.



To summarize, the use of the SEMAT kernel provides many benefits at the team level. Visualize a library of interchangeable practices, written using a language that can be as light or as formal as the situation requires, and built around the immutable fact that all software endeavors are successful based on progression of certain key elements. The team can select from that library based on the needs of the endeavor and *compose* their way of working rather than *create* it. They can build the optimum way for their situation. They can measure the progress and health of their project using standard metrics that are independent of those practices and methods—again invention and distraction that the team can avoid. New members of the team need not learn everything from scratch. All of these factors contribute positively to improving time-to-effectiveness for the team. Reducing such wasted time and energy, in addition to helping the team achieve its goals, clearly has cumulative benefits for the organization at large, thus helping attain the CIO's goal of “more with less.”

Organizational Level—Growth and Competitiveness Mean “More with Less”

As discussed at the beginning of this article, at the organizational level the objective is business growth. In our scenario this is what Dave is responsible for. He knows his business will grow only if his customers are happy and keep coming back, providing more business opportunities for the organization. He knows speed to market and relevance are improved through having a highly motivated and driven team. He also knows he needs to run a tight ship. Therefore, Dave wants to introduce mechanisms to nurture a continuously learning and inspiring culture. Not only does Dave want his teams to learn from one another, he also wants them to learn from others in the industry. This requires some changes.

A common vocabulary and body of knowledge. One would have thought Dave's team would share a vocabulary because they work for the same company, but this is not the case. Many teams within his organization use different

terminology, and often the same word can carry different meanings. In order for one team to learn successful practices from another team, it has to cross this unnecessary language barrier.

Essence tears down this language barrier by enabling practices to be described using one single vocabulary—the kernel. In addition, Essence provides a structure to organize practices such that they can be discovered, assessed, and applied quickly.

Teams can also contribute what they have learned back to these practices, thus building a structured library of knowledge—namely, practices. This also provides a learning roadmap for individuals in the organization. The more practices from the library they can master, the more competent they become. The benefits of a common practice library reach beyond single projects. For example, it can substantially help the communications problems documented on many failed IT projects.⁶

Empowering teams to change their way of working safely. This must be done not just in minor steps, but also in major ways. Dave wants to encourage his teams to be innovative, but at the same time he worries that they may venture into chaos. As such, his approach has tended to err toward the conservative, and the flexibility given to team leaders like Smith was rather limited. With Essence and the layering of practices on top of the kernel, however, organizations have a proven framework from which they can change their way of working safely. This allows Dave to focus on mandating a minimum number of specific practices—those that he needs the teams to “practice” in a consistent manner. At the same time he can also empower the teams to innovate in the other areas of the method, adding and replacing practices as and when it is appropriate.

Teams get a controlled and intuitive way to replace an existing practice or add a new one. This empowers teams to right-size their methods safely and in major ways, if appropriate.

Continual performance improvement/true learning organization. As mentioned previously, Dave’s organization had adopted CMMI, the Unified Process, and, lately, agile practices, each providing a different emphasis. When embracing a new

approach, however, his organization unfortunately threw away many of the good practices they had learned in the past, along with the parts they wanted to free themselves from. This was a great loss for the organization. In fact, it led to significant and costly reinvention. For example, when they introduced agile practices they had to reinvent the way architectural work was accomplished. By adopting the Essence kernel and working with practices rather than monolithic methods, Dave’s organization became a true learning organization—continually growing and improving the set of practices they use and improving their development capability.

Teams can easily understand what is working and where they need to tune their practices—dropping or refining the unsuccessful ones and adding appropriate new ones to address the changing needs of their development organization.

The result is a software-development department that is not only allowed, but also encouraged to explore the possibilities presented by new techniques and practices in the industry. Successful practices (the good ideas) can be slotted into the lightweight Essence framework and incorporated quickly into the organization’s way of working. Such freedom is highly motivating in a knowledge-worker environment—and motivation is key to attracting and retaining the best employees in the industry, and therefore to heightened effectiveness in the software organization.

Organizations benefit from SEMAT, too, and not simply because of the roll-up benefits from teams. The practice-based approach to ways of working, or methods, means the organization invests at this level and has a greater chance of reuse, rather than creating bespoke descriptions of project-specific methods. Resource groups master practices rather than methods, arming them with the ability to move more easily among teams and giving the CIO greater flexibility in the workforce. This more efficient management of practices, combined with higher adaptability and effectiveness of people, supports the “more with less” requirement imposed by the business.

SEMAT Success Stories

Without doubt, Essence is a powerful tool, but it has to be used appropriately. The SEMAT approach has helped a number of large-scale organizations and development endeavors for many years, both in general and in specific scenarios.⁴

Offshore development. The ideas behind Essence have been applied in a major initiative involving a large, well-known Japanese consumer electronics company that had outsourced development to a vendor in China. This client had never outsourced, nor did it understand the impending risks and see the value of iterative development in a use case and test-driven manner. The Chinese outsourcer was accustomed to following methods that were dictated by its customers. These methods tended to be traditional waterfall approaches. Thus, getting started with good communication and a clear vision was a huge challenge. The solution was to start off using the Essence kernel alphas to agree on how teams should be organized and composed. The lightweight intuitive nature of the alpha-state cards was able to help both the client and the vendor team leaders visualize the way they could work together.

The endeavor started with eight client staff and gradually grew to 10 client staff/20 vendor staff, then to 30 client staff/50 vendor staff. At the end of the endeavor, there were 80 client staff and 200 vendor staff involved.

When members joined the development team, they were given a quick briefing of the kernel and the new practices (iterative development, use-case-driven development, test-driven development). Project managers monitored the state of development and the size of teams. When a team became too big, it was split, and the new team leader was trained in the kernel to understand the forces acting on the team. Clearly, the involvement of each team and individual was not static, even within this one major endeavor. Not only was Essence able to help the teams get started, it also helped them grow.

Collaborative global software development. Building on the foundation provided by the Essence kernel, a major global reinsurance company has established a family of collaboration

models that cover the whole spectrum of software- and application-development work. Four collaboration models have been built on the same kernel from the same set of 12 practices. The models are exploratory, standard, maintenance, and support. Each defines a lightweight governance process suitable for the kind of work being undertaken and provides enough practices for the teams to get started.

This has many organizational, as well as local, benefits, including:

- *The unification of the people working in their service organization.* The application-development group is organized into a number of independent services. Teams are formed by drawing on the members of the various services as they are needed. The use of the kernel provides the common ground that allows these teams to come together quickly and ensure all speak the same language.

- *The enabling of flexible sourcing policies.* Adopting a kernel-based approach also allows the various services to flex their capacity safely by drawing on external resources and suppliers. The use of the kernel allows everyone to get up to speed quickly with the way of working and easily understand his or her roles and responsibilities.

- **Increased agility and improved project performance.** By focusing on the practices that join up the services and the teamwork needed to develop good software, the organization has been able to empower the teams and the services they draw on to continuously inspect and adapt, and safely embrace agile and lean thinking in their day-to-day work.

- *Global collaboration.* The company's application-development teams exist all over the world. By providing a standard, lightweight way of working, greater global collaboration and in some cases global-sourcing models have been achieved.

Standard Application Life-Cycle Management Tooling

A number of companies have used the Essence approach to help establish practice-independent tooling and application life-cycle management frameworks, including:

- A major systems integrator assembled an open source, low-cost tooling

environment to support distributed teams working around the Essence kernel, as well as its support for practice composition and execution. The environment runs as an organizational service and has been successfully used on a variety of projects. The flexibility inherent in Essence has allowed the same tooling and practices to be used in support of both agile and waterfall ways of working.

- A major U.K. government department introduced a kernel-based agile tool set to enable disciplined agility and the tracking of project progress and health in a practice-independent fashion. This allowed teams to adopt the agile practices of their choice within a consistent, effective, lightweight governance framework. It also helped the teams stay on track and avoid some of the pitfalls that can occur when transitioning to an agile way of working.

In both cases the common ground provided by the kernel and the results focus of the alphas enabled the widespread adoption of the tooling without compromising the teams' autonomy and empowerment.


Becoming Better, Faster, Cheaper, and Happier

The effectiveness of a method should be measured in how much better, faster, cheaper software you develop and how much happier your customers and employees are. Better methods results in improvements in all these factors. Better software means increased competitiveness for your products. Faster is critical to getting products on the market. Cheaper often comes as a side effect of faster, but also from automation. And happier customers come from many sources, one being the creation of better user experiences.

Better, faster, cheaper, and happier are all elements of the CIO's priorities (that is, getting more with less). SEMAT provides the foundation that allows a large organization's application-development department to make improvements to all these elements, resulting in a more effective team of motivated professionals truly contributing to the competitiveness of the organization.

Organizations do not need to wait to benefit from the Essence kernel. Teams and departments can start ben-

efiting today. The primary value that Essence brings is preventing the costly reinvention and unnecessary relearning of what is already known.

Essence can help your organization get to where you know you need to go, and it will help you get there faster and cheaper, so you are ready for whatever the future brings. 

Related articles on queue.acm.org

The Essence of Software Engineering: The SEMAT Kernel

Ivar Jacobson, Pan-Wei Ng, Paul McMahon, Ian Spence, and Svante Lidman
<http://queue.acm.org/detail.cfm?id=2389616>

A Conversation with Steve Bourne, Eric Allman, and Bryan Cantrill

<http://queue.acm.org/detail.cfm?id=1454460>

Voyage in the Agile Memeplex

Philippe Kruchten
<http://queue.acm.org/detail.cfm?id=1281893>

References

1. Gartner Inc. Top 10 CIO business and technology priorities in 2014; www.gartnerinfo.com/sym23/evtm_219_CIOtop10%5B3%5D.pdf.
2. Jacobson, I., Ng, P.-W., McMahon, P., Spence, I., Lidman, S. 2012. The essence of software engineering: the SEMAT kernel. *ACM Queue* 10, 10 (2012); <http://queue.acm.org/detail.cfm?id=2389616>.
3. Jacobson, I., Ng, P.-W., McMahon, P. E., Spence, I., Lidman, S. *The Essence of Software Engineering: Applying the SEMAT Kernel*. Addison-Wesley, 2014.
4. Jacobson, I., Ng, P.-W., Spence, I. The essential unified process. *Dr. Dobbs's Journal* (Aug, 2006).
5. Object Management Group. Essence: the kernel and language for software engineering methods; http://semat.org/wp-content/uploads/2014/02/Essence_final_submission_18Feb13.pdf.
6. Rosencrance, L. Survey: Poor communication causes most IT project failures; *Computerworld*, 2007; http://www.computerworld.com/s/article/9012758/Survey_Poor_communication_causes_most_IT_project_failures.
7. Serena Software Inc. 2014 IT priorities survey: app dev gets down to business. Conducted at the Gartner Application Architecture, Development and Integration (AADI) Summit; <http://www.serena.com/index.php/en/solutions/app-dev-delivery/infographic-application-development-priorities-2014/>.

Ivar Jacobson, chair of Ivar Jacobson International, is a father of components and component architecture, use cases, the Unified Modeling Language, and the Rational Unified Process. He has contributed to modern business modeling and aspect-oriented software development.

Pan-Wei Ng coaches large-scale systems development involving many millions of lines of code and hundreds of people per release, helping them transition to a lean and agile way of working, not forgetting to improve their code and architecture and to test through use cases and aspects.

Ian Spence is CTO at Ivar Jacobson International and the team leader for the development of the SEMAT kernel. An experienced coach, he has introduced hundreds of projects to iterative and agile practices.

Paul E. McMahon (pemcmahon@acm.org) is an independent consultant focusing on coaching project managers, team leaders, and software professionals in the practical use of lean and agile techniques in constrained environments.

Decoupled from IP, TCP is at last able to support multihomed hosts.

BY CHRISTOPH PAASCH AND OLIVIER BONAVENTURE

Multipath TCP

THE INTERNET RELIES heavily on two protocols. In the network layer, IP provides an unreliable datagram service and ensures any host can exchange packets with any other host. Since its creation in the 1970s, IP has seen the addition of several features, including multicast, IPsec (IP security), and QoS.

The latest revision, IPv6, supports 16-byte addresses.

The second major protocol is Transmission Control Protocol (TCP), which operates in the transport layer and provides a reliable bytestream service on top of IP. TCP has evolved continuously since the first experiments in research networks.

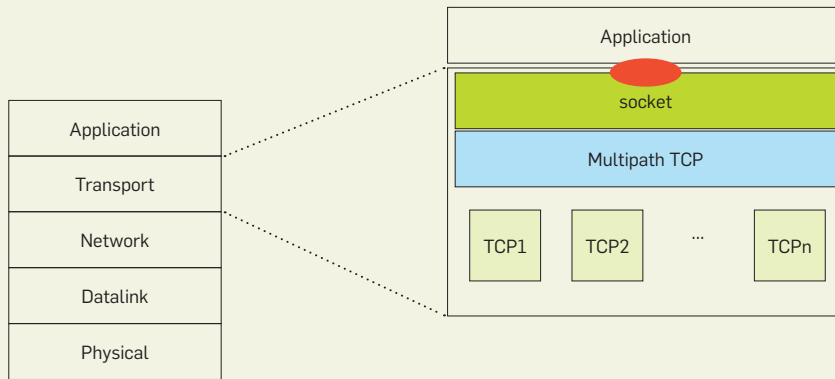
Still, one of the early design decisions of TCP continues to frustrate many users. TCP and IP are separate protocols, but the separation between the network and transport protocols is not complete. To differentiate the individual data streams among incoming packets, a receiving end host demultiplexes the packets based on the so-called 5-tuple, which includes the IP addresses, port numbers, and protocol identifiers. This implies a TCP connection is bound to the IP addresses used on the client and the server at connection-establishment time. Despite the growing

importance of mobile nodes such as smartphones and tablets, TCP connections cannot move from one IP address to another. When a laptop switches from Ethernet to Wi-Fi it obtains another IP address. All existing TCP connections must be torn down and new connections restarted.

Various researchers have proposed solutions to this problem over the past several years. The first approach was to solve the problem in the network layer. Examples of such solutions include Mobile IP, Host Identity Protocol (HIP), and Site Multihoming by IPv6 Intermediation (Shim6). A significant drawback of these network-layer solutions is that they hide all changes to the addresses and thus the paths to TCP's congestion-control scheme. This is inefficient since the congestion-control scheme sends data over paths that may change without notice.

Another possible solution is to expose multiple addresses to the transport

Figure 1. Multipath TCP in the stack.



layer. This is the approach chosen for Stream Control Transmission Protocol (SCTP),¹⁷ which is an alternative transport protocol capable of supporting several IP addresses per connection. The first versions of SCTP used multiple addresses in failover scenarios, but recent extensions have enabled it to support the simultaneous use of several paths.⁹

Unfortunately, besides niche applications such as signaling in telephony networks, SCTP has not been widely deployed. One reason is that many firewalls and Network Address Translation (NAT) boxes are unable to process SCTP packets and thus simply discard them. Another reason is that SCTP exposes a different API from the socket API to the

applications. These two factors lead to the classic chicken-and-egg problem. Network manufacturers do not support SCTP in their firewalls because no application is using this protocol; and application developers do not use SCTP because firewalls discard SCTP packets. There have been attempts to break this vicious circle by encapsulating SCTP on top of UDP (User Datagram Protocol) and exposing a socket interface to the application, but widespread usage of SCTP is still elusive.

Multipath TCP (MPTCP) is designed with these problems in mind. More specifically, the design goals for MPTCP are:¹⁵

- ▶ It should be capable of using multiple network paths for a single connection,
- ▶ It must be able to use the available network paths at least as well as regular TCP, but without starving TCP,
- ▶ It must be as usable as regular TCP for existing applications, and
- ▶ Enabling MPTCP must not prevent connectivity on a path where regular TCP works.

The following section describes the main architectural principles that underlie MPTCP. In an ideal network, these simple principles should have been sufficient. Unfortunately, this is not the case in today's Internet, given the prevalence of middleboxes, as explained later.

The Architectural Principles

Applications interact through the regular socket API, and MPTCP manages the underlying TCP connections (called subflows⁶) that are used to carry the actual data. From an architectural viewpoint, MPTCP acts as a shim layer between the socket interface and one or more TCP subflows, as shown in Figure 1. MPTCP requires additional signaling between the end hosts. It achieves this by using TCP options to achieve the following goals:

- ▶ Establish a new MPTCP connection.
- ▶ Add subflows to an MPTCP connection.
- ▶ Transmit data on the MPTCP connection.

An MPTCP connection is established by using the three-way handshake with TCP options to negotiate its usage. The MP_CAPABLE option in the SYN

Figure 2. Connection establishment.

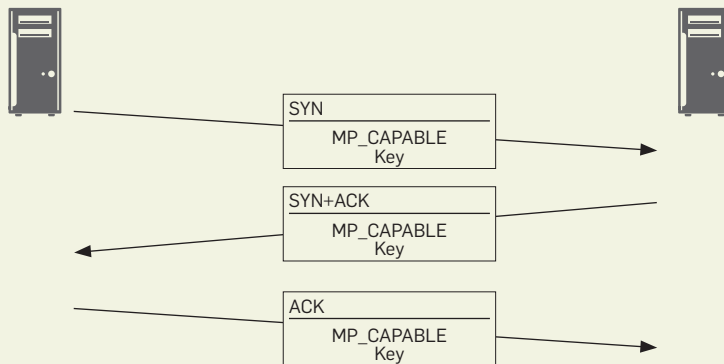
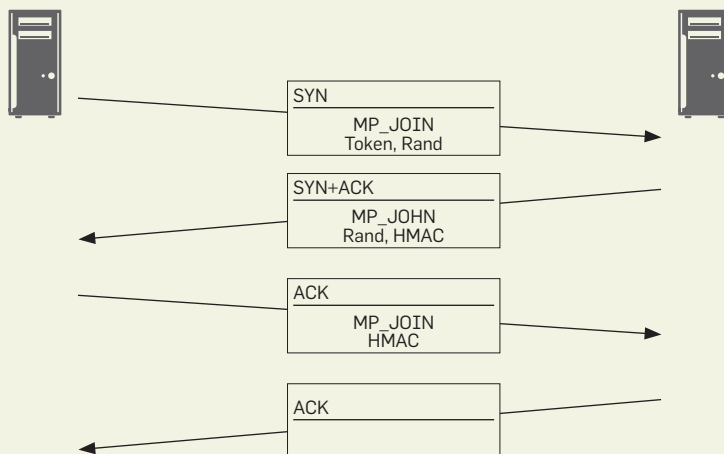


Figure 3. Establishment of an additional subflow.



segment indicates the client supports MPTCP. This option also contains a random key used for security purposes. If the server supports MPTCP, then it replies with a SYN+ACK segment that also contains the `MP_CAPABLE` option. This option contains a random key chosen by the server. The third ACK of the three-way handshake also includes the `MP_CAPABLE` option to confirm the utilization of MPTCP and the keys to enable stateless servers.

The three-way handshake shown in Figure 2 creates the first TCP subflow over one interface. To use another interface, MPTCP uses a three-way handshake to establish one subflow over this interface. Adding a subflow to an existing MPTCP connection requires the corresponding MPTCP connection to be uniquely identified on each end host. With regular TCP, a TCP connection is always identified by using the tuple $\langle \text{SourceIP}, \text{DestIP}, \text{SourcePort}, \text{DestPort} \rangle$.

Unfortunately, because NAT is present, the addresses and port numbers that are used on the client may not be the same as those exposed to the server. Although on each host the 4-tuple is a unique local identification of each TCP connection, this identification is not globally unique. MPTCP needs to be able to link each subflow to an existing MPTCP connection. For this, MPTCP assigns a locally unique token to each connection. When a new subflow is added to an existing MPTCP connection, the `MP_JOIN` option of the SYN segment contains the token of the associated MPTCP connection. This is illustrated in Figure 3.

The astute reader may have noticed that the `MP_CAPABLE` option does not contain a token. Still, the token is required to enable the establishment of subflows. To reduce the length of the `MP_CAPABLE` option and avoid using all the limited TCP options space (40 bytes) in the SYN segment, MPTCP derives the token as the result of a truncated hash of the key. The second function of the `MP_JOIN` option is to authenticate the addition of the subflow. For this, the client and the server exchange random nonces, and each host computes an hash-based message authentication code (HMAC) over the random nonce chosen by the other host and the keys exchanged during the initial handshake.

Now that the subflows have been established, MPTCP can use them to exchange data. Each host can send data over any of the established subflows. Furthermore, data transmitted over one subflow can be retransmitted on another to recover from losses. This is achieved by using two levels of sequence numbers.⁶ The regular TCP sequence number ensures data is received in order over each subflow and allows it to detect losses. MPTCP uses the data sequence number to reorder the data received over different subflows before passing it to the application.

From a congestion-control viewpoint, using several subflows for one connection leads to an interesting problem. With regular TCP, congestion occurs on one path between the sender and the receiver. MPTCP uses several paths, and two paths will typically experience different levels of congestion. A naive solution to the congestion problem in MPTCP would be to use the standard TCP congestion-control scheme on each subflow. This

can be easily implemented but leads to unfairness with regular TCP. In the network depicted in Figure 4, two clients share the same bottleneck link. If the MPTCP-enabled client uses two subflows, then it will obtain two-thirds of the shared bottleneck. This is unfair because if this client used regular TCP, it would obtain only half of the shared bottleneck.

Figure 5 shows the effect of using up to eight subflows across a shared bottleneck with the standard TCP congestion-control scheme (Reno). In this case, MPTCP would use up to 85% of the bottleneck's capacity, effectively starving regular TCP. Specific MPTCP congestion-control schemes have been designed to solve this problem.^{10,18} Briefly, they measure congestion on each subflow and try to move traffic away from those with the most congestion. To do so, they adjust the TCP congestion window on each subflow based on the level of congestion. Furthermore, the congestion windows on the different subflows are coupled

Figure 4. Coupled congestion control.

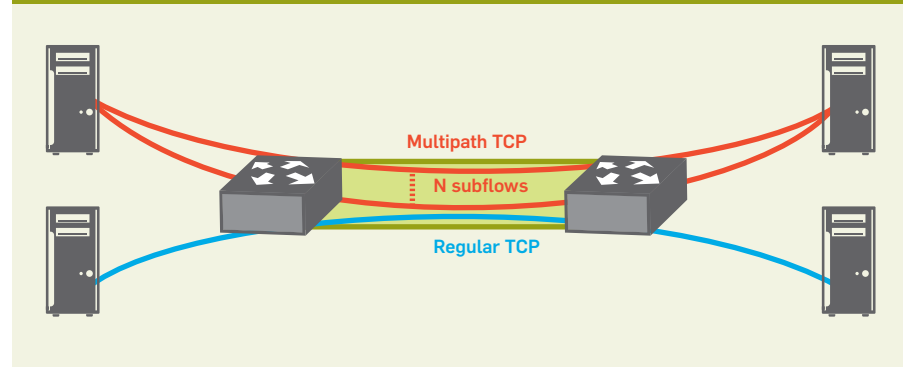
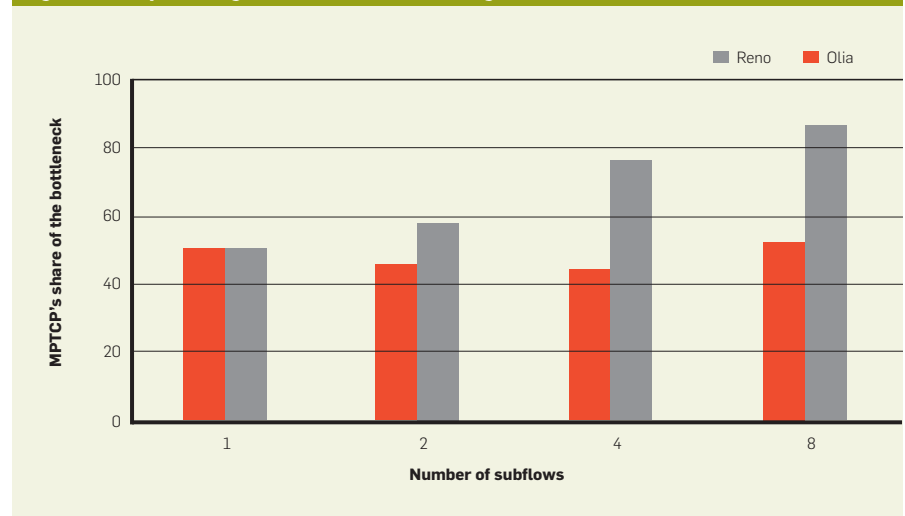


Figure 5. Coupled congestion control is fair to regular TCP across shared bottlenecks.




to ensure their aggregate does not grow faster than a single TCP connection. Figure 5 shows the OLIA congestion-control scheme¹⁰ preserves fairness with regular TCP across the shared bottleneck.


The Devil is in the Middleboxes

Today's Internet is completely different from the networks for which the original TCP was designed. These networks obeyed the end-to-end principle, which implies the network is composed of switches and routers that forward packets but never change their contents. Now the Internet contains various middleboxes in addition to these routers and switches. A recent survey revealed enterprise networks often contain more middleboxes than regular routers.¹⁶ These middleboxes include not only the classic firewalls and NATs, but also devices such as transparent proxies, virtual private network gateways, load balancers, deep-packet inspectors, intrusion-detection systems, and WAN accelerators. Many of these devices process and sometimes modify the TCP header and even the payload of passing TCP segments. Dealing with those middleboxes has been one of the most difficult challenges in designing the MPTCP protocol, as illustrated by the examples that follow.^{6,7,15}

Upon receiving data on a TCP subflow, the receiver must know the data-sequence number to reorder the data stream before passing it to the application. A simple approach would be to include the data-sequence number in a TCP option inside each segment. Unfortunately, this clean solution does not work. There are deployed middleboxes that split segments on the Internet. In particular, all modern network interface cards (NICs) act as segment-splitting middleboxes when performing TCP segmentation offloading (TSO). These NICs split a single segment into smaller pieces. In the case of TSO, CPU cycles are offloaded from the operating system to the NIC, as the operating system handles fewer, larger segments that are split down to maximum transmission unit (MTU)-sized segments by the NIC. The TCP options, including the MPTCP data-sequence number, of the large segment are copied in each smaller segment. As a result, the receiver will collect several



The data-sequence option thus accurately maps each byte from the subflow-sequence space to the data-sequence space, allowing the receiver to reconstruct the data stream.



segments with the same data-sequence numbers, unable to reconstruct the data stream correctly. The MPTCP designers solved this by placing a mapping in the data-sequence option, which defines the beginning (with respect to the subflow sequence number) and the end of the data-sequence number (indicating the length of the mapping). MPTCP can thus correctly work across segment-splitting middleboxes and can be used with NICs that use TCP segmentation offloading to improve performance.

Using the first MPTCP implementation in the Linux kernel to perform measurements revealed another type of middlebox. Since the implementation worked well in the lab, it was installed on remote servers. The first experiment was disappointing. An MPTCP connection was established but could not transfer any data. Still, the same kernel worked perfectly in the lab, but no one could understand why longer delays would prevent data transfer. The culprit turned out to be a local firewall that was changing the sequence numbers of all TCP segments. This feature was added to firewalls several years ago to prevent security issues with hosts that do not use random initial sequence numbers. In some sense, the firewall was fixing a security problem in older TCP stacks, but in trying to solve this problem, it created another problem. The mapping from subflow-sequence number to data-sequence number was wrong as the firewall modified the former. Since then, the mapping in the data-sequence option uses relative subflow-sequence numbers compared with the initial sequence number, instead of using absolute sequence numbers.⁶

The data-sequence option thus accurately maps each byte from the subflow-sequence space to the data-sequence space, allowing the receiver to reconstruct the data stream. Some middleboxes may still disturb this process: the application-level gateways that modify the payload of segments. The canonical example is active FTP. FTP uses several TCP connections that are signaled by exchanging ASCII-encoded IP addresses as command parameters on the control connection. To support active FTP, NAT boxes have to modify the private IP address is being sent in ASCII by the client host. This implies a modification of not only the content of the payload,

but also, sometimes, its length, since the public IP address of the NAT device may have a different length in ASCII representation. Such a change in the payload length will make the mapping from subflow-sequence space to data-sequence space incorrect. MPTCP can even handle such middleboxes thanks to a checksum that protects the payload of each mapping. If an application-level gateway modifies the payload, then the checksum will be corrupted; MPTCP will be able to detect the payload change and perform a seamless fallback to regular TCP to preserve the connectivity between the hosts.

Several researchers have analyzed the impact of these middleboxes in more detail. One study used active measurements over more than 100 Internet paths to detect the prevalence of various forms of middleboxes.⁸ The measurements revealed protocol designers can no longer assume the Internet is transparent when designing a TCP extension; and no TCP extension can assume a single field of the TCP header will reach the destination without any modification. TCP options are not safer. Some of these options can be modified on their way to the destination. Furthermore, some middleboxes remove or copy TCP options. Despite these difficulties, MPTCP is able to cope with most middleboxes.⁷ When faced with middlebox interference, MPTCP always tries to preserve connectivity. In some situations, this is achieved by falling back to regular TCP.^{6,7}

Use Cases

Two use cases that have already been analyzed in the literature will serve to illustrate possible applications of MPTCP. Other use cases will probably appear in the future.

MPTCP Smartphones are equipped with Wi-Fi and 3G/4G interfaces, but they typically use only one interface at a time. Still, users expect their TCP connections to survive when their smartphone switches from one wireless network to another. With regular TCP, switching networks implies changing the local IP address and leads to a termination of all established TCP connections.⁴ With MPTCP, the situation could change because it enables seamless handovers from Wi-Fi to 3G/4G and the opposite.¹³

Figure 6. 3G/WiFi handover with Multipath TCP.

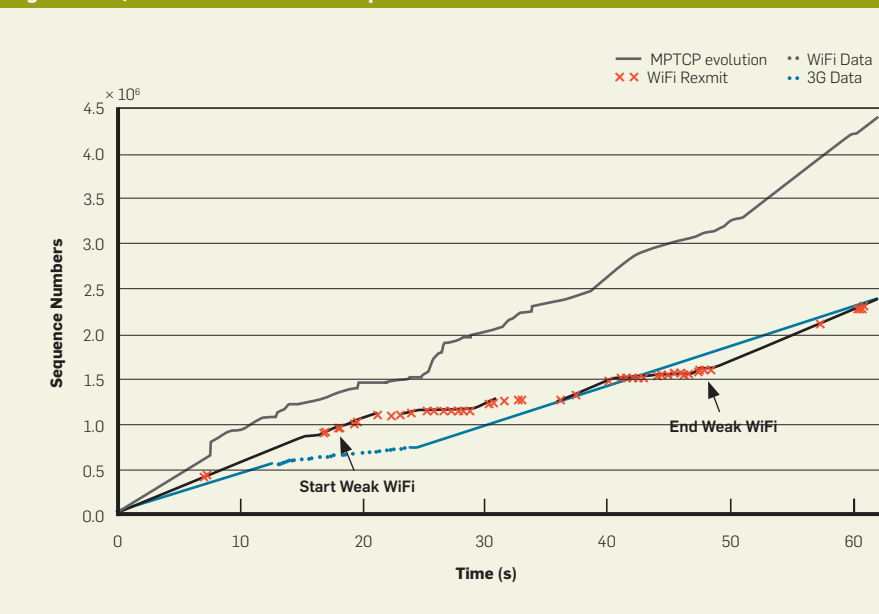
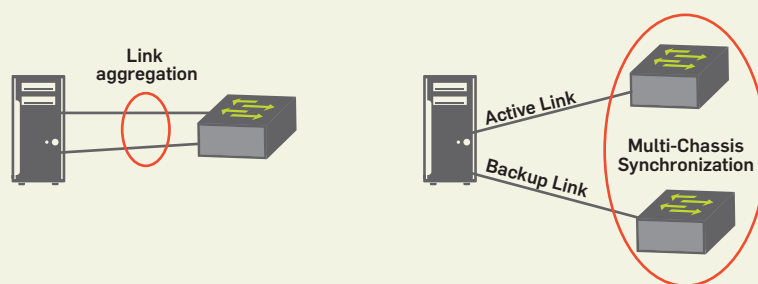


Figure 7. Link bonding for performance (left) and availability (right).



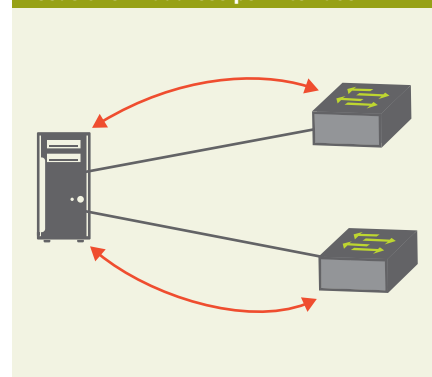
Different types of handovers are possible:

- First, the *make-before-break* handover can be used if the smartphone can predict that one interface will disappear soon (for example, because of a decreasing radio signal). In this case, a new subflow will be initiated on the second interface and the data will switch to this interface.

- With the *break-before-make* handover, MPTCP can react to the failure on one interface by enabling another interface and starting a subflow on it. Once the subflow has been created, the data that was lost as a result of the failure of the first interface can be retransmitted on the new subflow, and the connection continues without interruption.

- The third handover mode uses two or more interfaces simultaneously. With regular TCP, this would be a waste of energy. With MPTCP, data can

Figure 8. Link bonding with Multipath TCP needs one IP-address per interface.



be transmitted over both interfaces to speed up the data transfer. From an energy viewpoint, enabling two radio interfaces is more costly than enabling a single one; however, the display often consumes much more energy than the radio interfaces.² When the user looks at the screen (for example, while waiting for a Web page), increasing the

download speed by combining two interfaces could reduce the display usage and thus the energy consumption.

As an illustration, let's analyze the operation of MPTCP in real Wi-Fi and 3G networks. The scenario is simple but representative of many situations. A user starts a download inside a building over Wi-Fi and then moves outside. The Wi-Fi connectivity is lost as the user moves, but thanks to MPTCP the data transfer is not affected. Figure 6 shows a typical MPTCP trace collected by using both 3G and Wi-Fi. The *x*-axis shows the time since the start of the data transfer, while the *y*-axis displays the evolution of the sequence numbers of the outgoing packets.

For this transfer, MPTCP was configured to use both Wi-Fi and 3G simultaneously to maximize performance. The blue curve shows the evolution of the TCP sequence numbers on the TCP subflow over the 3G interface. The 3G subflow starts by transmitting at a regular rate. Between seconds 14 and 24, the 3G interface provides a lower throughput. This can be explained by higher congestion or a weaker radio signal since the smartphone moves inside the building during this period. After this short period, the 3G subflow continues to transmit at a constant rate. Note that TCP does not detect any loss over the 3G interface. The black curve shows the evolution of the TCP sequence number on the Wi-Fi subflow. The red crosses represent retransmitted packets.

A comparison of the 3G and Wi-Fi curves reveals Wi-Fi is usually faster than 3G, but packets are more frequently lost over the Wi-Fi interface. While the smartphone moves, there are short periods of time during which the Wi-Fi interface behaves like a black hole. The interface seems to be active, but no packet is transmitted correctly. These periods can last several seconds and severely impact the user experience when Wi-Fi is used alone. With MPTCP, the packets that are lost over the Wi-Fi interface are automatically retransmitted over the 3G subflow, and the MPTCP connection continues without interruption.

The upper gray curve shows the evolution of this MPTCP connection based on the returned acknowledgments. During the first seven seconds, MPTCP aggregates the Wi-Fi and 3G interfaces perfectly. The MPTCP throughput is the



With MPTCP, it is possible both to improve performance and to achieve high availability.



sum of the throughput of the two interfaces. Then a few packets are lost over the Wi-Fi interface. The first vertical bar on the gray curve corresponds to the reception of the first acknowledgment after the recovery of these losses. When the Wi-Fi interface is weak, the MPTCP throughput varies with the quality of the interface, but the data transfer continues. This ability to exploit the available interfaces quickly is a key benefit of MPTCP from a performance viewpoint.

Improving the data-transfer performance is not the only use case of MPTCP on smartphones.¹³ It is also possible to use only the 3G interface as a backup instead of enabling both the Wi-Fi and the 3G interfaces. When the Wi-Fi interface is active, all data is sent over it. If it becomes inactive, MPTCP automatically switches to the 3G interface to continue the data transfer and stops using it as soon as the Wi-Fi interface comes back.

Another possible use case is the 802.11 community network. Many cities are now covered by a large number of Wi-Fi access points accessible to many users.³ The density of these access points makes it possible for slow-moving users mainly to rely on Wi-Fi for Internet access and on MPTCP to maintain the connections when moving from one access point to another.

The smartphone use case has motivated Apple to implement MPTCP in iOS 7. As of this writing, MPTCP is not yet available through the socket API on iOS 7 and is used only to support the Siri voice-recognition application. It is also possible to use the MPTCP Linux kernel on some recent rooted Android smartphones (see <http://multipath-tcp.org/>).

MPTCP in the Data Center. Another important use case for MPTCP lies in data centers.¹⁴ Today, most servers are equipped with several high-speed interfaces, which can be combined to achieve either higher performance or better resilience to failures. When two or more interfaces are grouped together to improve performance, they are usually attached to the same switch, as illustrated on the left-hand side of Figure 7. To enable TCP to use these two interfaces efficiently, they usually appear as a single logical interface having one MAC address and one IP address. The bonding driver on

the server distributes the packets over the combined interfaces; several load-balancing algorithms exist.⁴ A round-robin system allows efficient spreading of the packets over the different interfaces. If the links have different delays, however, it causes reordering, which hurts TCP performance.

Most deployments opt for hash-based load-balancing techniques. Some fields of the Ethernet and IP and/or TCP headers are hashed to select the outgoing interface. Thanks to this hashing, the packets that belong to the same TCP connection are sent over the same interface to prevent reordering, and packets belonging to different connections are spread over the available interfaces. This solution works well if the traffic is composed of a large number of short TCP connections. Since all the packets belonging to one TCP connection are sent over the same interface, however, a single TCP connection cannot achieve a higher throughput than the link it uses. This is a major limitation, given that long TCP connections are frequent in data centers.¹

In some deployments, two or more interfaces are combined to improve availability. In this case, each interface is attached to a different switch, as illustrated on the right-hand side of Figure 7. Most deployments opt for using the same MAC and IP addresses on the two interfaces. One interface is used to carry all packets, and the other serves as a backup. When this mode is chosen, only one interface at a time is used to carry packets.

With MPTCP, it is possible both to improve performance and to achieve high availability. A typical MPTCP-aware server design would use two interfaces attached to different switches, as illustrated in Figure 8. Each interface has its own MAC and IP addresses. Once an MPTCP connection has been initiated over one interface, MPTCP will announce the other available IP addresses to the remote host by using the `ADD _ ADDR` option,⁶ and subflows will be established over these interfaces.

Once these subflows have been established, the MPTCP congestion-control scheme will dynamically spread the load over the available interfaces. If one interface or any intermediate switch fails, MPTCP automatically changes to the remaining paths.

Note that in contrast with existing load-balancing solutions, the interfaces used by MPTCP do not need to have the same bandwidth. This allows MPTCP to increase the throughput even for a single data stream by distributing the load over all interfaces. Indeed, the Linux kernel implementation of MPTCP described earlier¹¹ allowed memory-to-memory transfers between two high-end servers, each equipped with six 10Gbps interfaces. MPTCP was able to use the capacity of the six interfaces efficiently and reached 53Gbps for a single connection.¹²

On the server side, most experiments with MPTCP have been performed with the Linux MPTCP kernel (available from <http://multipath-tcp.org>). A FreeBSD implementation is being developed, and one commercial load balancer supports MPTCP.⁵

Conclusion

MPTCP is a major extension to TCP. By decoupling TCP from IP, TCP is at last able to support multihomed hosts. With the growing importance of wireless networks, multihoming is becoming the norm instead of the exception. Smartphones and data centers are the first use cases where MPTCP can provide benefits.

Acknowledgments

The development of MPTCP has been a collective work involving many researchers, including Sébastien Barré, Gregory Detal, Fabien Duchene, Phil Eardley, Alan Ford, Mark Handley, Benjamin Hesmans, and Costin Raiciu. This work has been supported by the European Commission under FP7 (the seventh Framework Programme for Research) projects Trilogy, CHANGE, and Trilogy 2; a gift from Google; and the Bestcom IAP. C

Related articles on queue.acm.org

Passively Measuring TCP Round-trip Times

Stephen D. Strowes

<http://queue.acm.org/detail.cfm?id=2539132>

You Don't Know Jack about Network Performance

Kevin Fall, Steve McCanne

<http://queue.acm.org/detail.cfm?id=1066069>

TCP Offload to the Rescue

Andy Currid

<http://queue.acm.org/detail.cfm?id=1005069>

References

- Benson, T., Akella, A. and Maltz, D.A. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, 2010.
- Carroll, A. and Heiser, G. An analysis of power consumption in a smartphone. In *Proceedings of the Usenix Annual Technical Conference*, 2010.
- Castignani, G., Blanc, A., Lampropoulos, A. and Montavont, N. Urban 802.11 community networks for mobile users: Current deployments and perspectives. *Mobile Networks and Applications* 17, 6 (2012), 796–807.
- Davis, T. et al. Linux Ethernet bonding driver how-to, 2011; <https://www.kernel.org/doc/Documentation/networking/bonding.txt>.
- Eardley, P. Survey of MPTCP implementations. Work in progress, 2014; <http://tools.ietf.org/html/draft-eardley-mptcp-implementations-survey-02>.
- Ford, A., Raiciu, C., Handley, M. and Bonaventure, O. TCP extensions for multipath operation with multiple addresses. RFC6824, 2014; <http://www.rfc-editor.org/rfc/rfc6824.txt>.
- Hesmans, B., Duchene, F., Paasch, C., Detal, G. and Bonaventure, O. Are TCP extensions middlebox-proof? In *Proceedings of the 2014 Workshop on Hot Topics in Middleboxes and Network Function Virtualization*.
- Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A., Handley, M. and Tokuda, H. Is it still possible to extend TCP? In *Proceedings of the 2011 ACM SIGCOMM Internet Measurement Conference*.
- Iyengar, J. R., Amer, P. D. and Stewart, R. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. *IEEE/ACM Transactions on Networking* 14, 5 (2006), 951–964.
- Khalili, R., Gast, N., Popovic, M., Upadhyay, U. and Le Boudec, J.-Y. MPTCP is not pareto-optimal: performance issues and a possible solution. In *Proceedings of the 8th ACM International Conference on Emerging Networking Experiments and Technologies*, 2012.
- Paasch, C., Barré, S., Detal, G. and Duchene, F. Linux kernel implementation of Multipath TCP, 2014; <http://multipath-tcp.org>.
- Paasch, C., Detal, G., Barré, S., Duchene, F. and Bonaventure, O. The fastest TCP connection with MultiPath TCP, 2014; <http://multipath-tcp.org/pmwiki.php?n=Main.50Gbps>.
- Paasch, C., Detal, G., Duchene, F., Raiciu, C. and Bonaventure, O. Exploring mobile/Wi-Fi handover with Multipath TCP. In *Proceedings of the ACM SIGCOMM workshop on Cellular Networks: Operations, Challenges, and Future Design*, 2012.
- Raiciu, C., Barré, S., Pluntke, C., Greenhalgh, A., Wischik, D. and Handley, M. Improving datacenter performance and robustness with Multipath TCP. *ACM SIGCOMM Computer Communication Review* 41, 4 (2011), 266–277.
- Raiciu, C., Paasch, C., Barré, S., Ford, A., Honda, M., Duchene, F., Bonaventure, O. and Handley, M. How hard can it be? Designing and implementing a deployable Multipath TCP. In *Proceedings of the 9th Symposium on Networked Systems Design and Implementation*, 2012.
- Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S. and Sekar, V. Making middleboxes someone else's problem: Network processing as a cloud service. *ACM SIGCOMM Computer Communication Review* 42, 4 (2012), 13–24.
- Stewart, R. and Xie, Q. *Stream Control Transmission Protocol: A Reference Guide*. Addison-Wesley, 2001.
- Wischik, D., Raiciu, C., Greenhalgh, A., and Handley, M. Design, implementation and evaluation of congestion control for Multipath TCP. In *Proceedings of the 8th Usenix Symposium on Networked Systems Design and Implementation*, 2011.

Christoph Paasch is a Ph.D. student at Université catholique de Louvain, Louvain-la-Neuve, Belgium, where he leads the development of Multipath TCP in the Linux kernel.

Olivier Bonaventure is a professor at Université catholique de Louvain, Louvain-la-Neuve, Belgium, where he leads the networking group. His research focuses on Internet protocols.

DOI:10.1145/2500876

Student-participation data from the inaugural MITx (now edX) course—6.002x: Circuits and Electronics—unpacks MOOC student behavior.

BY DANIEL T. SEATON, YOAV BERGNER, ISAAC CHUANG, PIOTR MITROS, AND DAVID E. PRITCHARD

Who Does What in a Massive Open Online Course?

MASSIVE OPEN ONLINE COURSES (MOOCs) collect valuable data on student learning behavior; essentially complete records of all student interactions in a self-contained learning environment, with the benefit of large sample sizes. Here, we offer an overview of how the 108,000 participants behaved in 6.002x - Circuits and Electronics, the first course in MITx (now edX) in the Spring 2012 semester. We divided participants into tranches based on the extent of their assessment activities, ranging from browsers (constituting ~76% of the participants but only 8% of the total time spent in the course) to certificate earners (7% of participants who accounted for 60% of total time). We examined

how the certificate earners allocated their time among the various course components and what fraction of each they accessed. We analyze transitions between course components, showing how student behavior differs when solving homework vs. exam problems. This work lays the foundation for future studies of how various course components, and transitions among them, influence learning in MOOCs.

Though free online courses are not new,⁸ they have reached an unprecedented scale since late 2011. Three organizations—Coursera, edX, and Udacity—have released MOOCs¹³ drawing more than 100,000 registrants per course. Numbers from these three initiatives have since grown to more than 100 courses and three million total registrants, resulting in 2012 being dubbed “The Year of the MOOC” by the *New York Times*.¹⁶ Though there has been much speculation regarding how these initiatives may reshape higher education,^{6,12,20} little analysis has been published to date describing student behavior or learning in them.

Our main objective here is to show how the huge amount of data available in MOOCs offers a unique research opportunity, a means to study detailed student behavior in a self-contained learning environment throughout an

» key insights

- **Data collected in MOOCs provides insight into student behavior, from weekly e-textbook reading habits to context-dependent use of learning resources when solving problems.**
- **In 6.002x, 76% of participants were browsers who collectively accounted for only 8% of time spent in the course, whereas, the 7% of certificate-earning participants averaged 100 hours each and collectively accounted for 60% of total time.**
- **Students spent the most time per week interacting with lecture videos and homework, followed by discussion forums and online laboratories; however, interactions with the videos and lecture questions were distinctly bimodal, with half the certificate earners accessing less than half of these resources.**




entire course. We thus studied the approximately 100GB of time-stamped log data describing student interactions with the inaugural MITx course 6.002x Circuits and Electronics in spring 2012, data at least two orders of magnitude larger than was analyzed in previous studies of online learning.^{10,21} We develop and exhibit several ways to study student interactions with course resources. We do not analyze demographic factors, but rather differentiate students by number of assessment items attempted and total time spent in the course. We studied all registrants with these metrics before turning to the more detailed time allocation and resource use of students earning a certificate of accomplishment. For certificate earners, we examined the use of course components (such as lecture videos, homework, and discussion forums) in terms of user time allocation and total fraction accessed. We also studied resource use during problem solving, revealing markedly different patterns of accesses and time allocation among different course components when students solve problems during homework vs. when taking exams.


6.002x, Procedures, Data Analysis

With some modification for online delivery, the 14 weeklong units of 6.002x largely mirrored a traditional on-campus course in both format and timing. The course sequence (see Figure 1, left navigation bar) involves lecture sequences consisting of lecture videos (annotated PowerPoint slides and actual MIT lectures) with embedded lecture questions, tutorial videos (recitation substitute), homework (three to four multi-part problems), and lab assignments (interactive circuit toolbox). Overall grades were determined by homework (15%), labs (15%), a midterm (30%), and a final (40%). Supplementary materials (see Figure 1, top navigation bar) included a course textbook (navigable page images), a staff and student-editable wiki and moderated student discussions. For further exploration of course structure and available resources, see the archived course at <https://6002x.mitx.mit.edu/>.

Parsing tracking logs. Analysis of tracking logs is an established means for understanding student behavior in blended and online courses.^{5,14} In the



The correlation of attrition with less time spent in early weeks begs the question of whether motivating students to invest more time would increase retention rates.



6.002x tracking logs, each interaction (click) contained relevant information, including username, resource ID, interaction details, and timestamp. Interaction details are context-dependent (such as correctness of a homework problem submission, body text of a discussion post, and page number for book navigation). The edX software is distributed through the cloud; meaning interaction data is logged on multiple servers. In total, approximately 230 million interactions were logged in 38,000 log files over the initial Spring 2012 semester.

We preprocessed the logs into separate time-series for each participant, then compiled participant-level descriptive statistics on resource usage, including number of unique resources accessed, total frequency of accesses per resource type, and total time spent per resource. We also parsed problem submissions, generating a response matrix including correctness and number of attempts. Where possible, we crosschecked our event-log assessment data against a MySQL database serving the 6.002x courseware. All log parsing was performed through standard modules in Python and R.

Estimation of time spent on resources. Time estimation for each participant involved measuring the durations between a student's initial interaction with a resource and the time the student would navigate away. We accumulated durations calculated from each participant's time series for each separate course component, including homework, book, and discussion forums. We found evidence that durations shorter than three seconds represent students navigating to desired resources; hence, we do not count these intervals as activity. In addition, we did not accumulate durations longer than one hour, assuming users have disengaged from their computers. Using alternate values of the high cutoff (20 minutes to one hour) can change overall time by 10%–20% but did not significantly alter relationships regarding time allocation among course components or total time spent by different participants.

An important point is that time accumulated is associated with the resource displayed at the moment; for example, if a student references the book while doing homework, this duration is accumulated with book time. In our case,

only direct interactions with the homework are logged with homework resources. There are clearly alternatives to this approach (such as considering all time between opening and answering a problem as problem-solving time²¹). Our time-accumulation algorithm is partially thwarted by users who open multiple browser windows or tabs; edX developers are considering ways to account for this in the future.

Results

The novelty and publicity surrounding MOOCs in early 2012 attracted a large number of registrants who were more curious than serious. We still take participation in assessment as an indication of serious intent. Of the 154,000 registrants in 6.002x in spring 2012, 46,000 never accessed the course, and the median time spent by all remaining participants was only one hour (see Figure 2a). We had expected a bimodal distribution of total time spent, with a large peak of “browsers” who spent only on the order of one hour and another peak from the certificate earners at somewhere more than 50 hours. There was, in fact, no minimum between

Figure 1. Screenshot of typical student view in 6.002x.

All course components are accessed from the interface shown below. The left sidebar defines the course sequence; weekly units include lecture sequences (videos and questions), homework, lab, and tutorials. The header navigation provides access to supplementary materials, including digital textbook, discussion forums, and wiki. The main frame represents the first lecture sequence; beige boxes below the header indicate lecture videos and questions.

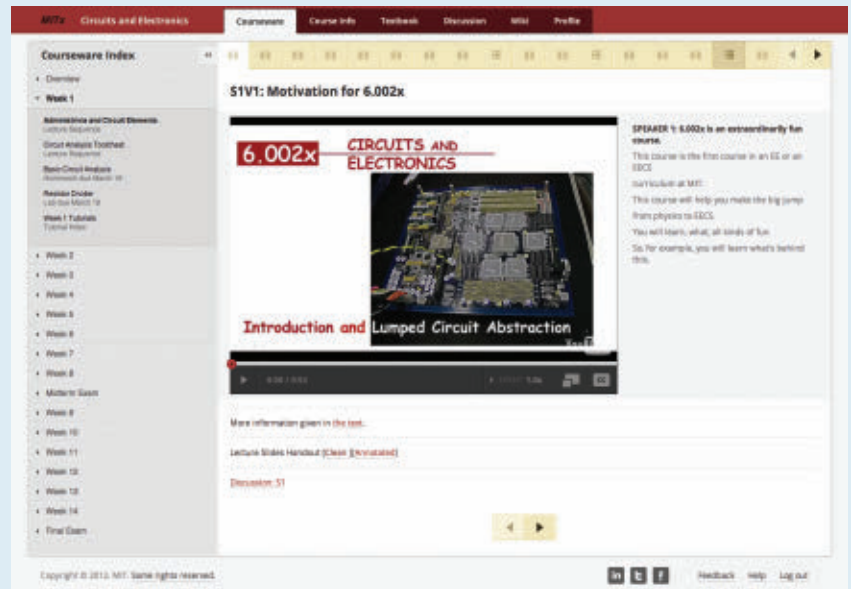


Figure 2. Tranches, total time, and attrition.

(a) Distribution of time spent by participants in 6.002x (time axis is log-transformed); we divided the noncertificate earners into tranches based on percentage of assessment activity they attempted (see also Table 1);

(b) percentage of total measured time spent by each tranche; and (c) average time a student invested per week. The shaded regions near Week 8 and Week 14 represent the time span for the midterm and final exams.

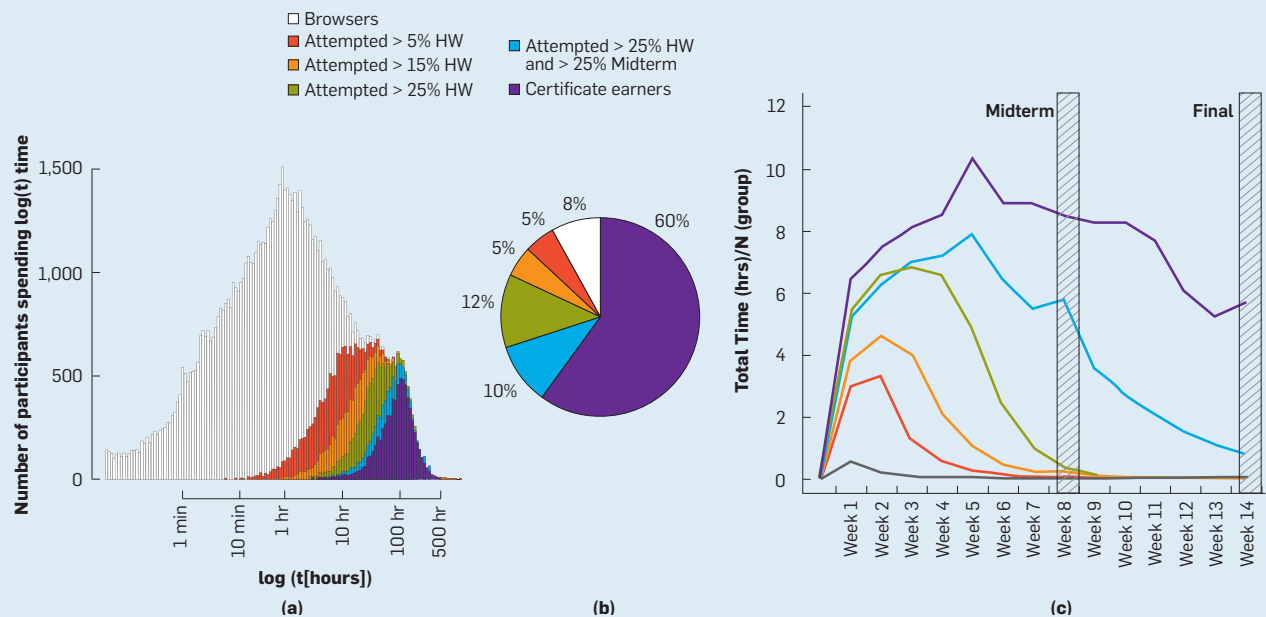
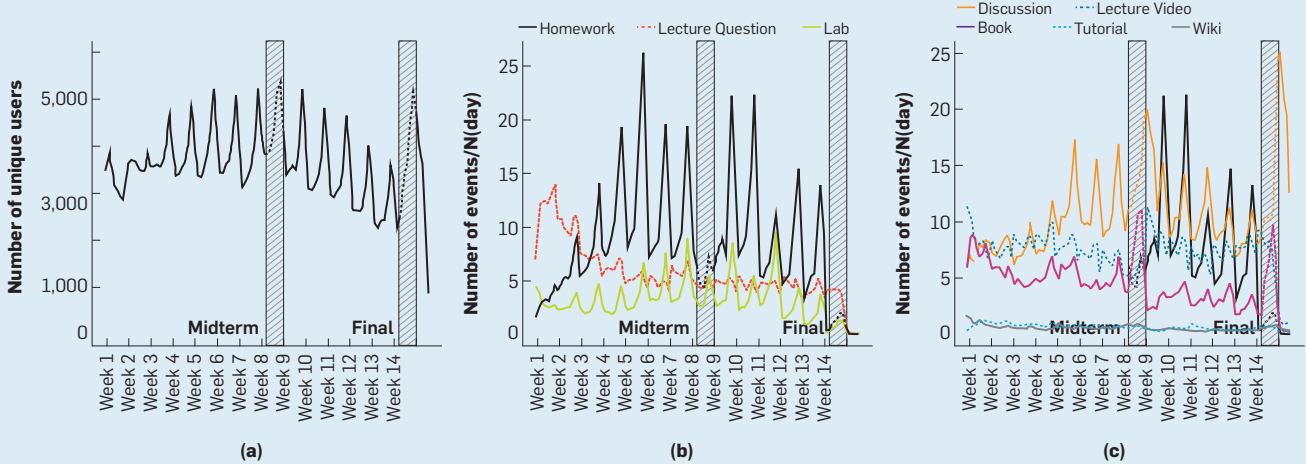


Figure 3. Frequency of accesses.

From left to right, number of unique certificate earners N active per day, their average number of accesses each day for assessment-based and learning-based course components. Plot (a) highlights the periodicity and trends of the certificate earners. Plot (b) is for assessment, including homework, lab, and lecture questions, showing number of accesses per active users that day. Learning-based components in plot (c) include lecture videos, textbook, discussion, tutorial, and wiki, showing discussion forums were used more heavily and with strong periodicity later in the term, similar to graded activities in plot (a), while other components lack periodicity and vary greatly in terms of frequency of accesses.

The shaded regions near Week 8 and Week 14 represent the time span for the midterm and final exams.



these extremes, only a noticeable shoulder (see Figure 2a). The intermediate durations are filled with attempters we divided into tranches (in colors) on the basis of how many assessment items they attempted on homework and exams: browsers (gray) attempted < 5% of homework; tranche 1 (red) 5%–15% of homework; tranche 2 (orange) 15%–25% of homework; tranche 3 (green) > 25% of homework; and tranche 4 (cyan) > 25% of homework and 25% of midterm exam. Certificate earners (purple) attempted most of the available homework, midterm, and final exams. The median total time spent in the course for each tranche was 0.4 hours, 6.4 hours, 13.1 hours, 30.0 hours, 53.0 hours, and 95.1 hours, respectively. In addition to these tranches, just over 150 certificate earners spent fewer than 10 hours in the course, possibly representing a highly skilled tranche seeking certification. Similarly, just over 250 test takers spent fewer than 10 hours in the course and completed more than 25% of both exams but did not earn a certificate.

The average time spent in hours per week for participants in each tranche is shown in Figure 2c. Tranches attempting fewer assessment items not only taper off earlier, as the majority

of participants effectively drop out, but also invested less time in the first few weeks than the certificate earners. The correlation of attrition with less time spent in early weeks begs the question of whether motivating students to invest more time would increase retention rates.

In the rest of this article, we restrict ourselves to certificate earners, as they accounted for the majority of resource consumption; we also wanted to study time and resource use over the whole semester.

Frequency of accesses. Figure 3a shows the number of active users per day for certificate earners, with large peaks on Sunday deadlines for graded homework and labs but not for lecture questions. There is a downward trend in the weeks between the midterm and the final exam (shaded regions). No homework or labs were assigned in the last two weeks before the final exam, though the peaks persist. We plotted activity in events (clicks subject to time cutoffs) per active student per day for assessment-based course components and learning-based components in Figure 3b and Figure 3c. Homework sets and the discussion forums account for the highest rate of activity per student,

with discussion activity increasing over the semester. Lecture question events decay early as homework activity increases. Textbook use peaks during exams, and there is a noticeable drop in textbook activity after the midterm, as is typical in traditional courses.¹⁸

Time on tasks. Time represents the principal cost function for students, so it is important to study how students allocate time among available course components.^{15,19} Figure 4 shows the most time is spent on lecture videos; since three to four hours per week is close to the total duration of the scheduled videos, students who rewind and reviewed the videos must compensate for those speeding up playback or omitting videos.

The most significant change over the first seven weeks was the apparent transfer of time from lecture questions to homework, as in Figure 4. Considering a performance-goal orientation (see Figure 5), it should be noted that homework counted toward the course grade, whereas lecture questions did not. But even on mastery-oriented grounds, students might have viewed completion of homework as sufficient evidence of understanding lecture content. The prominence of time spent in discussion

forums is especially noteworthy, as they were neither part of the course sequence nor did they count for credit. Students presumably spent time in discussion forums due to their utility, whether pedagogical or social or both. The small spike in textbook time at the midterm, a larger peak in the number of accesses, as in Figure 3, and the decrease in textbook use after the midterm are typical of textbook use when online resources are blended with traditional on-campus courses.¹⁸ Further studies comparing blended and online textbook use are also relevant.^{3,17}

Percentage use of course components. Along with student time allocation, the fractional use of the various course components continues to be an important metric for instructors deciding how to improve their courses and researchers studying the influence of course structure on student activity and learning. For fractional use, we plotted the percentage of certificate earners having accessed at least a certain percentage of resources in a course component (see Figure 5). Homework and labs (each 15% of overall grade) reflect high fractional use. The inflection in these curves near 80% might have been higher but for the course policy of dropping the two lowest-graded assignments. The low proportionate use of textbook and tutorials is similar to the distribution

observed for supplementary (not explicitly included in the course sequence) e-texts in large introductory physics courses,¹⁶ though the 6.002x textbook was assigned in the course syllabus. The course authors were disappointed with the limited use of tutorial videos, suspecting that placing tutorials after the homework and laboratory (they were meant to help) in the course sequence

was partly responsible. (The wiki and discussion forums had no defined number of resources so are excluded here.)

To better understand the middle curves representing lecture videos and lecture problems, it helps to recall that the negative slope of the curve is the density of students accessing that fraction of that course component (see Figure 5b and Figure 5c). Interestingly,

Figure 4. Time on tasks.

Certificate earners average time spent, in hours per week, on each course component; midterm and final exam weeks are shaded.

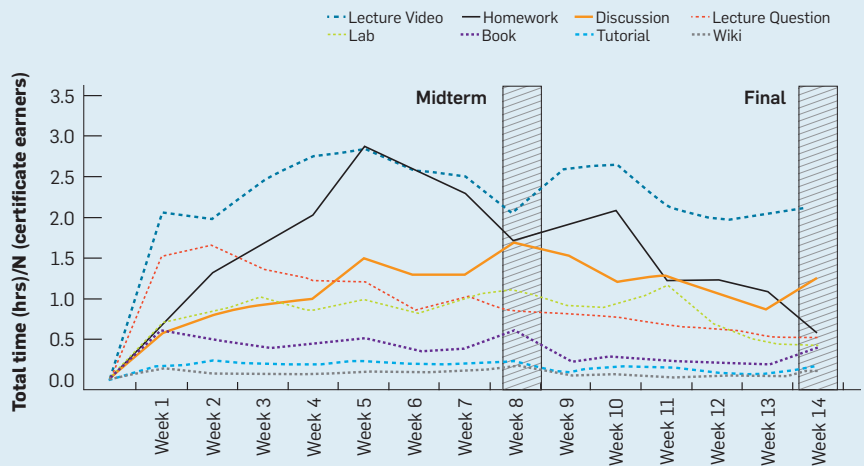
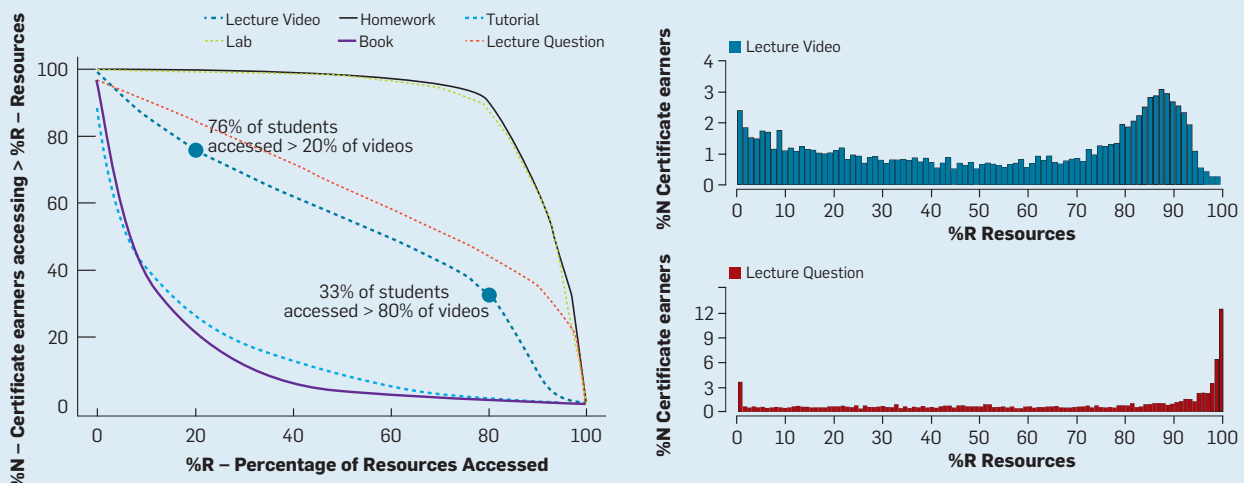


Figure 5. Fractional use of resources.

(a) Percentage of certificate earners who accessed greater than %R of that type of course resource. The density of users is the negative slope of the usage curve. Two points indicating bimodality of lecture video use are plotted: 76% of students accessed > 20% of lecture videos, and 33% of students accessed > 80% of lecture videos. (b) Bimodal distribution for videos accessed (as percentage). And (c) distribution of lecture questions accessed.



the distribution for the lecture videos is distinctly bimodal: 76% of students accessed over 20% of the videos (or 24% of students accessed less than 20%), and 33% accessed over 80% of the videos. This bimodality merits further study into learning preferences; for example, do some students learn from other resources exclusively? Or did they master the content prior to the course? The distribution of lecture-problem use is flat between 0% and 80%, then rises sharply, indicating that many students accessed nearly all of them. Along with the fact that the time on lecture questions drops steadily in the first half of the term (see Figure 2), this distribution suggests students not only allocated less time to them, some abandoned the lecture problems entirely.

Resources used when problem solving. Patterns in the sequential use of resources by students may hold clues to cognitive and even affective state.² We therefore explored the interplay between use of assessment and learning resources by transforming time-series data into transition matrices between resources. The transition matrix contains all individual resource-resource transitions we aggregated into transitions between major course components. The completeness of the 6.002x learning environment means students did not have to leave it to reference the

textbook, review earlier homework, or search the discussion forums. We thus had a unique opportunity to observe transitions to all course components accessed by students while working problems. In previous studies of on-line problem solving this information was simply missing.²¹

Figure 6 highlights student transitions from problems (while solving them) to other course components, treating homework sets, the midterm, and the final exam as separate assessment types of interest. Figure 6 shows the discussion forum is the most frequent destination during homework problem solving, though lecture videos consume the most time. During exams (midterm and final are similar), previously done homework is the primary destination, while the book consumes the most time. Student behavior on exam problems thus contrasts sharply with behavior on homework problems. Note that because homework was aggregated, we could not isolate “references to previous assignments” for students doing homework.

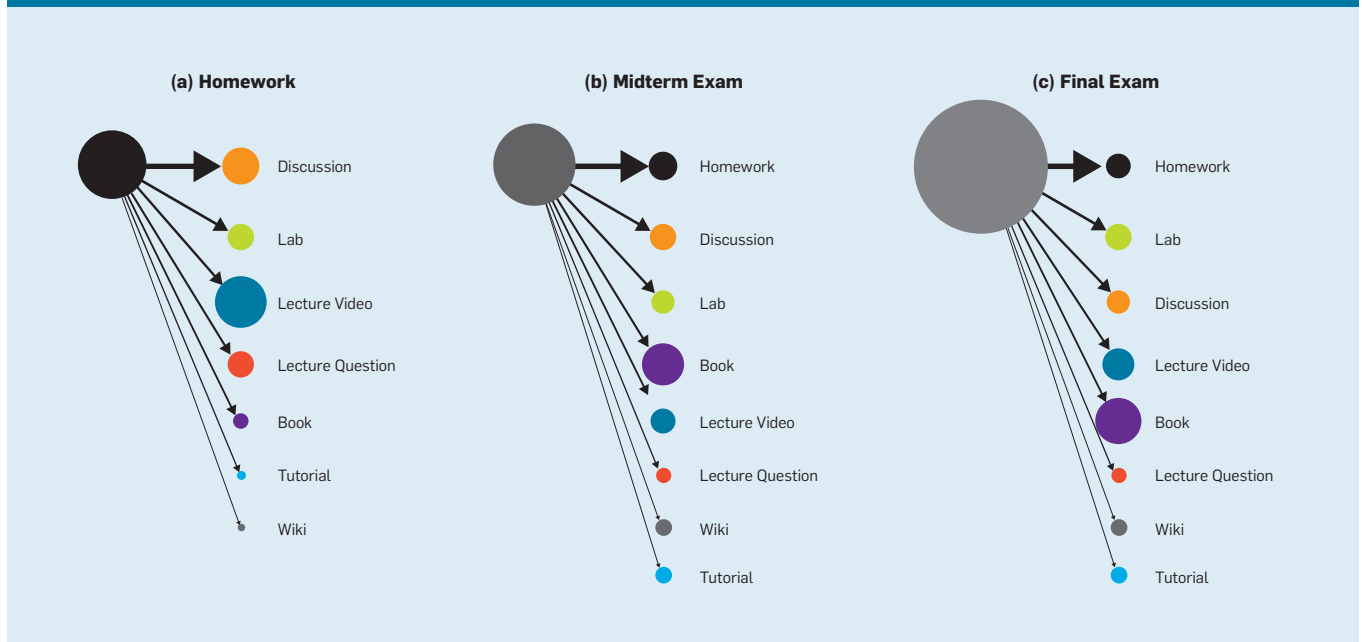
Conclusion

This article’s major contribution to course analysis is showing how MOOC data can be analyzed in qualitatively different ways to address important issues: attrition/retention, distribution of students’ time among resources,

fractional use of those resources, and use of resources during problem solving. Among the more significant findings is that participants who attempted over 5% of the homework represented only 25% of all participants but accounted for 92% of the total time spent in the course; indeed, 60% of the time was invested by the 6% who ultimately received certificates. Participants who left the course invested less effort than certificate earners, with those investing the least effort during the first two weeks tending to leave sooner. Most certificate earners invested the plurality of their time in lecture videos, though approximately 25% of the earners watched less than 20%. This suggests the need for a follow-up investigation into the correlations between resource use and learning. Finally, we highlight the significant popularity of the discussion forums in spite of being neither required nor included in the navigation sequence. If this social learning component played a significant role in the success of 6.002x, a totally asynchronous alternative might be less appealing, at least for a complex topic like circuits and electronics.

Some of these results echo effects seen in on-campus studies of how course structure affects resource use¹⁸ and performance outcomes^{4,11,19} in introductory (college) courses. This


Figure 6. Transitions to other components during problem solving on (a) homework, (b) midterm, and (c) final. Arrows are thicker in proportion to overall number of transitions, sorting components from top to bottom; node size represents total time spent on that component.



and future MOOC studies should further illuminate on-campus education generally. On the other hand, MOOCs could well take advantage of insights from existing research in on-campus education (such as frequent exams drive resource use and maximize learning outcomes¹¹).


Finally, we emphasize that MOOCs provide a unique view into the learning of a large, diverse population of students, allowing research based on detailed insight into all aspects of a course. In contrast to most previous studies of on-campus educational environments, we have time-stamped logs of essentially all student behavior and associated learning throughout the entirety of a course, all with solid statistics and the ability to study specific student cohorts (such as based on effort, learning habits, and demographics⁹). Combining time-on-task observations with measures of learning paves the way for measuring learning value—the amount learned per unit time spent on a given course component—possibly extending previous studies of online learning.^{7,15} This, in turn, will allow a process of cyclic improvement based on research development, experimentation, and measurement of learning outcomes, supporting improvement of educational content and delivery. Since many MOOCs largely mirror traditional on-campus courses in types of resources, format, and chronology, we anticipate insights into, and improvements of, learning in traditional on-campus courses as well.

Acknowledgments


This work is supported, but is not endorsed, by National Science Foundation grant DUE-1044294; additional support provided by a Google Faculty Award. We thank MITx for data access and J. deBoer and other members of the Teaching and Learning Laboratory and the Research in Learning, Assessing and Tutoring Effectively groups at MIT for their helpful suggestions and comments. 

References

1. Ames, C. and Archer, J. Achievement goals in the classroom: Students' learning strategies and motivation processes. *Journal of Educational Psychology* 80, 3 (1988), 260.
2. Baker, R.S., D'Mello, S.K., Rodrigo, M.M.T., and Graesser, A.C. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning



Textbook use peaks during exams, and there is a noticeable drop in textbook activity after the midterm, as is typical in traditional courses.



- environments. *International Journal of Human-Computer Studies* 68, 4 (2010).
3. Cummings, K., French, T., and Cooney, P.J. Student textbook use in introductory physics. In *Proceedings of the Physics Education Research Conference*, 2002.
 4. Freeman, S., Haak, D., and Wenderoth, M.P. Increased course structure improves performance in introductory biology. *CBE-Life Sciences Education* 10, 2 (2011).
 5. Guzdial, M. *Deriving Software Usage Patterns from Log Files*. Technical Report GIT-GVU-93-41, 1993.
 6. Hyman, P. In the year of disruptive education. *Commun. ACM*, 55, 12 (Dec. 2012).
 7. Jiang, L., Elen, J., and Clarebout, G. The relationships between learner variables, tool-usage behavior, and performance. *Computers in Human Behavior* 25, 2 (2009).
 8. Johnstone, S.M. Open educational resources serve the world. *Educause Quarterly* 28, 3 (2005).
 9. Kolowich, S. Who takes MOOCs? *Inside Higher Education* 5 (2012).
 10. Kortemeyer, G. Gender differences in the use of an online homework system in an introductory physics course. *Physical Review Special Topics: Physics Education Research* 5, 1 (2009).
 11. Laverty, J.T., Bauer, W., Kortemeyer, G., and Westfall, G. Want to reduce guessing and cheating while making students happier? Give more exams! *Physics Teacher* 50, 9 (2012).
 12. Martin, F.G. Will massive open online courses change how we teach? *Commun. ACM* 55, 8 (2012).
 13. McAuley, A., Stewart, B., Siemens, G., and Cormier, D. The MOOC model for digital practice. Social Sciences and Humanities Research Council, *Knowledge Synthesis Grant on the Digital Economy*, 2010.
 14. Minaei-Bidgoli, B., Kortemeyer, G., and Punch, W.F. Enhancing online learning performance: An application of data mining methods. *Immunohematology* 62, 150 (2004).
 15. Morote, E.S. and Pritchard, D.E. What course elements correlate with improvement on tests in introductory Newtonian mechanics? *American Journal of Physics* 77 (2009).
 16. Pappano, L. The year of the MOOC. *The New York Times* (Nov. 2, 2012).
 17. Podolefsky, N. and Finkelstein, N. The perceived value of college physics textbooks: Students and instructors may not see eye to eye. *The Physics Teacher* 44 (2006).
 18. Seaton, D.T., Bergner, Y., Kortemeyer, G., Rayyan, S., Chuang, I., and Pritchard, D.E. The impact of course structure on etext use in large-lecture introductory-physics courses. In *Proceedings of the Physics Education Research Conference*, 2013.
 19. Stewart, J., Stewart, G., and Taylor, J. Using time-on-task measurements to understand student performance in a physics class: A four-year study. *Physical Review Special Topics-Physics Education Research* 8, 1 (2012).
 20. Vardi, M.Y. Will MOOCs destroy academia? *Commun. ACM* 55, 11 (Nov. 2012).
 21. Warnakulasooriya, R., Palazzo, D.J., and Pritchard, D.E. Time to completion of Web-based physics problems with tutoring. *Journal of the Experimental Analysis of Behavior* 88, 1 (2007).

Daniel T. Seaton (dseaton@mit.edu) is a postdoctoral research fellow in the Office of Digital Learning at the Massachusetts Institute of Technology, Cambridge, MA.

Yoav Bergner (ybergner@ets.org) is a research scientist in the Center for Advanced Psychometrics at ETS, Princeton, NJ.

Isaac Chuang (ichuang@mit.edu) is a joint professor in the Department of Physics and the Department of Electrical Engineering and Computer Science and a member of the Research Laboratory of Electronics at the Massachusetts Institute of Technology, Cambridge, MA.

Piotr Mitros (piotr@mitros.org) is the chief scientist at edX and affiliated with the Center for Artificial Intelligence and Learning at the Massachusetts Institute of Technology, Cambridge, MA.

David E. Pritchard (dpritch@mit.edu) is the Cecil and Ida Green Professor of Physics and a member of the Center for Ultracold Atoms and the Research Laboratory for Electronics at the Massachusetts Institute of Technology, Cambridge, MA.

Copyright held by Author(s)/Owner(s)

DOI:10.1145/2591012

With the help of computational proof assistants, formal verification could become the new standard for rigor in mathematics.

BY JEREMY AVIGAD AND JOHN HARRISON

Formally Verified Mathematics

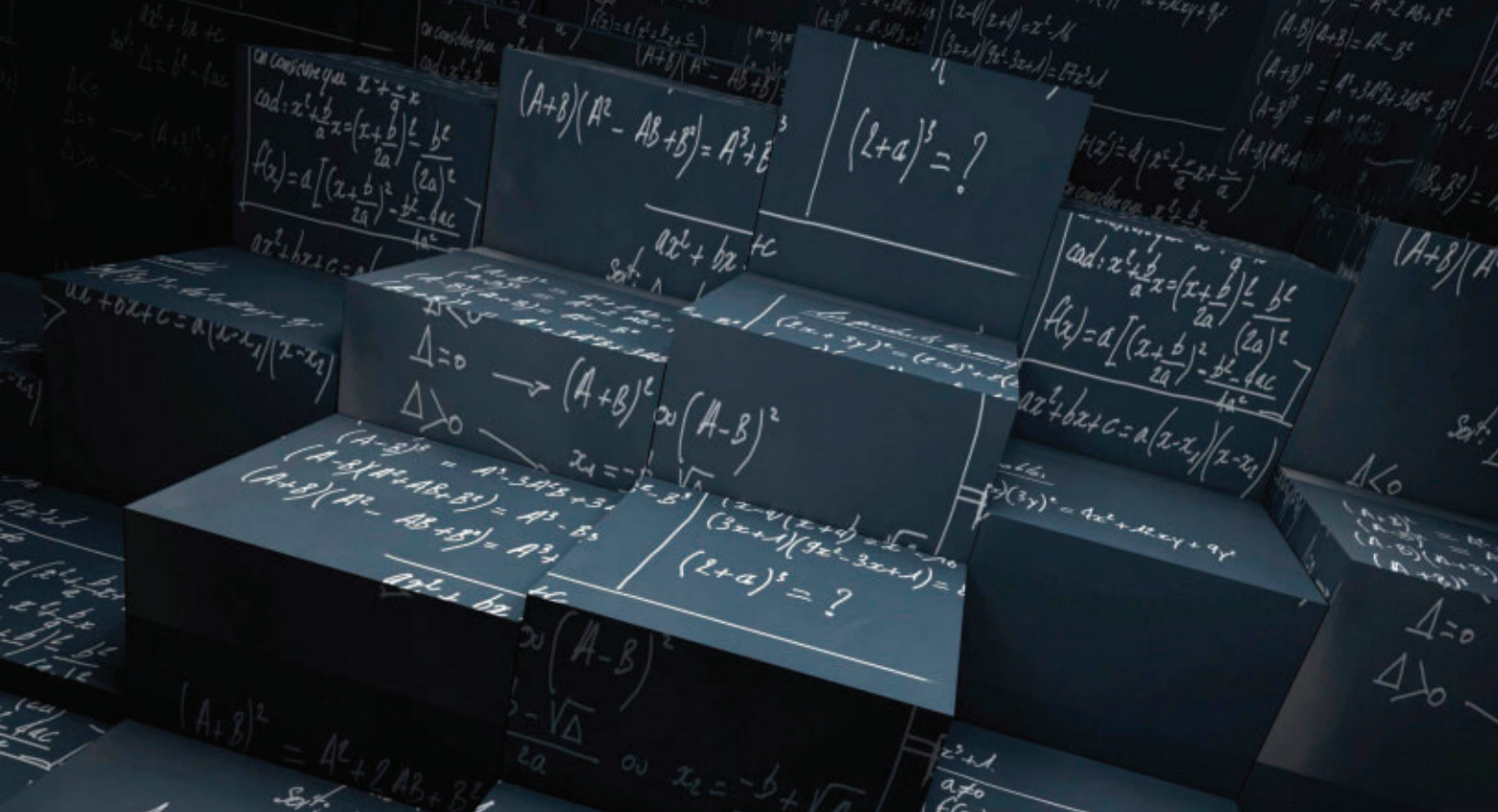
FROM THE POINT of view of the foundations of mathematics, one of the most significant advances in mathematical logic around the turn of the 20th century was the realization that ordinary mathematical arguments can be represented in formal axiomatic systems in such a way their correctness can be verified mechanically, at least in principle. Gottlob Frege presented such a formal system in the first volume of his *Grundgesetze der Arithmetik*, published in 1893, though in 1903 Bertrand Russell showed the system to be inconsistent. Subsequent foundational systems include the ramified type theory of Russell and Alfred North Whitehead's *Principia Mathematica*, published in three volumes from 1910 to 1913; Ernst Zermelo's axiomatic set theory of 1908, later extended by Abraham Fraenkel; and Alonzo Church's simple type theory of 1940. When Kurt Gödel presented his celebrated incompleteness theorems in 1931, he began with the following assessment:

“The development of mathematics toward greater precision has led, as is well known, to the formalization of large tracts of it, so one can prove any theorem using nothing but a few mechanical rules. The most comprehensive formal systems that have been set up hitherto are the system of *Principia Mathematica* on the one hand and the Zermelo-Fraenkel axiom system of set theory (further developed by J. von Neumann) on the other. These two systems are so comprehensive that in them all methods of proof used today in mathematics are formalized, that is, reduced to a few axioms and rules of inference. One might therefore conjecture that these axioms and rules of inference are sufficient to decide any mathematical question that can at all be formally expressed in these systems. It will be shown below that this is not the case...”⁴

Gödel was right to claim the mathematics of his day could generally be formalized in axiomatic set theory and type theory, and these have held up, to today, as remarkably robust foundations for mathematics. Indeed, set-theoretic language is now ubiquitous, and most mathematicians take the language and axioms of set theory to underwrite their arguments, in the sense that any ambiguities in a claim or proof could, in principle, be eliminated by spelling out the details in set-theoretic terms. In the mid-1930s, a group of mathematicians writing under the pen name Nicolas Bourbaki adopted set theory as the nominal foundation for a series of influential treatises aiming to provide a self-con-

» key insights

- Among the sciences, mathematics is distinguished by its precise language and clear rules of argumentation.
- This fact makes it possible to model mathematical proofs as formal axiomatic derivations.
- Computational proof assistants make it possible to check the correctness of these derivations, thereby increasing the reliability of mathematical claims.



tained, rigorous presentation of the core branches of mathematics:

“...the correctness of a mathematical text is verified by comparing it, more or less explicitly, with the rules of a formalized language.”³

From this standpoint, the contemporary informal practice of writing a mathematical proof is an approximation to that ideal, whereby the task of a referee is to exercise professional judgment as to whether the proof could be expressed, in principle, in a way that conforms to the rules. The mathematician Saunders Mac Lane put it as follows:

“A Mathematical proof is rigorous when it is (or could be) written out in the first-order predicate language $L(\in)$ as a sequence of inferences from the axioms ZFC, each inference made according to one of the stated rules... When a proof is in doubt, its repair is usually just a partial approximation to the fully formal version.”¹⁵

This point of view is to some extent anticipated by the 17th century philosopher Gottfried Leibniz, who called for development of a universal language (*characteristica universalis*) in which anything can be expressed and a calculus of reasoning (*calculus ratiocinator*) for deciding the truth of assertions expressed in the *characteristica*. Leibniz’s ambitions were not limited to mathematics; he dreamed

of a time when any disputants could translate their disagreement into the characteristic and say to each other “*calculemus*” (“let us calculate”). In the 20th century, however, even Bourbaki conceded complete formalization was an unattainable ideal:

“...the tiniest proof at the beginning of the Theory of Sets would already require several hundreds of signs for its complete formalization... formalized mathematics cannot in practice be written down in full... We shall therefore very quickly abandon formalized mathematics.”³

Due to developments in computer science over the past few decades, it is now possible to achieve complete formalization in practice. Working with “computational proof assistants,” users are able to verify substantial mathematical theorems, constructing formal axiomatic derivations of remarkable complexity. Our goal in this article is to describe the current technology and its motivations, survey the state of the art, highlight some recent advances, and discuss prospects for the future.

Mathematical Rigor

The notion of proof lies at the heart of mathematics. Although early records of measurement and numeric computation predate the ancient Greeks, mathematics proper is commonly seen as having begun with de-

velopment of the deductive method, as exemplified by Euclid’s *Elements of Geometry*. Starting from axioms and postulates, a mathematical proof proceeds by a chain of incontrovertible logical steps to its conclusion. Through the ages, the method of the *Elements* was held to represent the paradigm of rigorous argumentation, to mathematicians, scientists, and philosophers alike. Its appeal is eloquently conveyed in John Aubrey’s short biography¹ of the philosopher Thomas Hobbes (1588–1697), who made his first serious contact with mathematics at the age of 40:

“Being in a Gentleman’s Library, Euclid’s *Elements* lay open, and ’twas the 47 *El. libri* 1 [Pythagoras’s Theorem]. He read the proposition. By *G—*, said he (he would now and then swear an emphaticall Oath by way of emphasis) *this is impossible!* So he reads the Demonstration of it, which referred him back to such a Proposition; which proposition he read. That referred him back to another, which he also read. *Et sic deinceps* [and so on] that at last he was demonstratively convinced of that trueth. This made him in love with Geometry.”¹

The encounter turned Hobbes into an enthusiastic amateur geometer “wont to draw lines on his thigh and on the sheets, abed.” He became notorious in later years for bombarding

top mathematicians with his error-ridden ruler-and-compass geometric constructions, some of which are now known to be impossible in principle.

But what, exactly, constitutes a proof? Who is to say whether a proof is correct or not? Maintaining that a proof need convince only the person reading it gives the notion a subjective character. In practice, proofs tend to become generally accepted when they persuade not just one or two people but a broad or particularly influential group of mathematicians. Yet as Hobbes himself noted, this does not entirely avoid the subjective and fallible character of the judgment:

“But no one mans Reason, nor the Reason of any one number of men, makes the certaintie; no more than an account is therefore well cast up, because a great many men have unanimously approved it.”¹²

In the history of mathematics, there is no shortage of controversy over the validity of mathematical arguments. Berkeley’s extended critique of the methods of the calculus in *The Analyst* (1734) is one example. Another is the “vibrating string controversy” among Leonhard Euler, Jean d’Alembert, and Daniel Bernoulli, hinging on whether an “arbitrary” continuous function on a real interval could be represented by a trigonometric series. Carl Friedrich Gauss is usually credited with providing the first correct proof of the fundamental theorem of algebra, asserting that every nonconstant polynomial over the complex numbers has a root, in his doctoral dissertation of 1799; but the history of that theorem is especially knotty, since it was not initially clear what methods could legitimately be used to establish the existence of the roots in question. Similarly, when Gauss presented his proof of the law of quadratic reciprocity in his *Disquisitiones Arithmeticae* (1801), he began with the observation that Legendre’s alleged proof a few years prior contained a serious gap.

Mathematicians have always been reflectively conscious of their methods, and, as proofs grew more complex in the 19th century, mathematicians became more explicit in emphasizing the role of rigor. This is evident in, for example, Carl Jacobi’s praise of Johann Peter Gustav Lejeune Dirichlet:

“Dirichlet alone, not I, nor Cauchy, nor Gauss knows what a completely rigorous mathematical proof is. Rather we learn it first from him. When Gauss says that he has proved something, it is very clear; when Cauchy says it, one can wager as much pro as con; when Dirichlet says it, it is certain...” (quoted by Schubring²¹).

Mathematics has, at critical junctures, developed in more speculative ways. But these episodes are invariably followed by corresponding periods of retrenchment, analyzing foundations and increasingly adopting a strict deductive style, either to resolve apparent problems or just to make the material easier to teach convincingly.⁶

Reflecting on the history of mathematics, we can to some extent disentangle two related concerns. The first is whether the methods used in a given proof are valid, or appropriate to mathematics. This has to do with coming to consensus as to the appropriate rules of argumentation. For example, is it legitimate to refer to negative numbers, complex numbers, and infinitesimals, and, if so, what properties do they have? Is it legitimate to use the axiom of choice or to apply the law of the excluded middle to statements involving infinite structures? The second concern has to do with whether, given a background understanding of what is allowed, a particular proof meets those standards; that is, whether or not the proof is correct. The foundational debates of the early 20th century, and the set-theoretic language and formalism that emerged, were designed to address the question as to what methods are legitimate, by establishing an in-principle consensus as to the inferences that are allowed. Formal verification is not designed to help with that; just as human referees can strive only to establish correctness with respect to an implicit conception of what is acceptable, formal correctness can be assessed only modulo an underlying axiomatic framework. But, as will become clear in the next section, establishing correctness is a nontrivial concern, and is where formal methods come into play.

Correctness Concerns

Mathematical proofs are complex objects, and becoming more so. Human

referees, even those with the best of intentions, are fallible, and mistakes are inevitably made in the peer-review process. A book written by Lecat in 1935 included 130 pages of errors made by major mathematicians up to 1900, and even mathematicians of the stature of J.E. Littlewood have published faulty proofs:

“Professor Offord and I recently committed ourselves to an odd mistake (*Annals of Mathematics Annals of Mathematics* (2) 49, 923, 1.5). In formulating a proof a plus sign got omitted, becoming in effect a multiplication sign. The resulting false formula got accepted as a basis for the ensuing fallacious argument. (In defence, the final result was known to be true.)”¹⁴

Every working mathematician must routinely deal with inferential gaps, misstatements, missing hypotheses, unstated background assumptions, imprecise definitions, misapplied results, and the like. A 2013 article in the *Notices of the American Mathematical Society* (Grcar⁷) on errors in the mathematical literature laments the fact that corrections are not published as often as they should be. Some errors are not easily repaired. The first purported proof of the four-color theorem in 1879 stood for a decade before a flaw was pointed out. Referees reviewing Andrew Wiles’s first proof of Fermat’s Last Theorem found a mistake, and it took Wiles and a former student, Richard Taylor, close to a year to find a way to circumvent it. Daniel Gorenstein announced, in 1983, that the classification of finite simple groups had been completed, unaware there was a gap in the treatment of the class of “quasisim” groups. The gap was not filled until 2001, and doing so required a 1,221-page proof by Michael Aschbacher and Stephen Smith.

Even when an argument turns out to be correct, judging it to be so can take a long time. Grigori Perelman posted three papers on arXiv in 2002 and 2003 presenting a proof of Thurston’s geometrization conjecture. This result, in turn, implies the Poincaré conjecture, one of the Clay Mathematics Institute’s famed Millennium Prize challenges. The proof was scrutinized by dozens of researchers, but it was not until 2006 that three independent groups determined that


any gaps in Perelman's original proof were minor, and could be filled using the techniques he had developed.

The increased complexity is exacerbated by the fact that some proofs rely on extensive calculation. Kenneth Appel's and Wolfgang Haken's 1976 proof of the four-color theorem relied on an exhaustive computer enumeration of combinatorial configurations. Subsequent proofs, though more efficient, have this same character. Proofs that depend on explicit checking of cases are nothing new in themselves; for example, proofs of Bertrand's conjecture (for $n \geq 1$ there is a prime $n \leq p \leq 2n$) often begin with a comment like "Let us assume $n \geq 4,000$, since one can verify it explicitly for other cases." But this feature took on dramatic proportions with Thomas Hales's 1998 proof of the Kepler conjecture, stating that no packing of spheres in 3D space has higher density than the natural face-centered cubic packing commonly used to stack oranges, cannonballs, and such. Hales, working with Samuel Ferguson, arrived at a proof in 1998 consisting of 300 pages of mathematics and calculations performed by approximately 40,000 lines of computer code. As part of the peer-review process, a panel of 12 referees appointed by the *Annals of Mathematics* studied the proof for four full years, finally returning with the verdict that they were "99% certain" of the correctness, but in the words of the editor Robert MacPherson:


"The news from the referees is bad, from my perspective. They have not been able to certify the correctness of the proof, and will not be able to certify it in the future, because they have run out of energy to devote to the problem. This is not what I had hoped for..."

"Fejes Tóth thinks that this situation will occur more and more often in mathematics. He says it is similar to the situation in experimental science—other scientists acting as referees can't certify the correctness of an experiment, they can only subject the paper to consistency checks. He thinks that the mathematical community will have to get used to this state of affairs."

The level of confidence was such that the proof was indeed published in the *Annals*, and no significant error has been found in it. Nevertheless, the



Every working mathematician routinely has to deal with inferential gaps, misstatements, missing hypotheses, unstated background assumptions, imprecise definitions, misapplied results, and the like.



verdict is disappointingly lacking in clarity and finality. In fact, as a result of this experience, the journal changed its editorial policy on computer-assisted proof so it will no longer even try to check the correctness of computer code. Dissatisfied with this state of affairs, Hales turned to formal verification, as we will see.

In November 2005, the *Notices of the American Mathematical Society* published an article by Brian Davies called "Whither Mathematics?" that raised questions about the mounting complexity of mathematical proof and the role of computers in mathematics. In August 2008, the *Notices* published an opinion piece by Melvyn Nathanson that also raised concerns about the status of mathematical proof:

"...many great and important theorems don't actually have proofs. They have sketches of proofs, outlines of arguments, hints and intuitions that were obvious to the author (at least, at the time of writing) and that, hopefully, are understood and believed by some part of the mathematical community."¹⁸

He concluded:

"How do we recognize mathematical truth? If a theorem has a short complete proof, we can check it. But if the proof is deep, difficult, and already fills 100 journal pages, if no one has the time and energy to fill in the details, if a 'complete' proof would be 100,000 pages long, then we rely on the judgments of the bosses in the field. In mathematics, a theorem is true, or it's not a theorem. But even in mathematics, truth can be political."¹⁸

Nathanson's essay did not explicitly mention contemporary work in formal verification. But a few months later, in December 2008, the *Notices* devoted an entire issue to formal proof, focusing on methods intended to alleviate Nathanson's concerns.

Automating Mathematics

As noted, for suitable formal proof systems, there is a purely mechanical process for checking whether an alleged proof is in fact a correct proof of a certain proposition. So, given a proposition p , we could in principle run a search program that examines in some suitable sequence (in order of, say, length and then alphabetical order) every potential proof of p and termi-

nates with success if it finds one. That is, in the terminology of computability theory, the set of provable formulas is recursively enumerable.


But this naive program has the feature that if p is not provable, it will run fruitlessly forever, so it is not a yes/no decision procedure of the kind Leibniz imagined. A fundamental limitative result due to Church and Turing shows this cannot be avoided, because the set of provable formulas is not recursive (computable). This inherent limitation motivates the more modest goal of having the computer merely check the correctness of a proof provided (at least in outline form) by a person. Another reaction to limitative results like this, and related ones due to Gödel, Tarski, Post, and others, is to seek special cases (such as restricting the logical form of the problem) where a full decision procedure is possible.

All three possibilities have been well represented in the development of formal proof and automated reasoning:


- ▶ Complete but potentially nonterminating proof search;
- ▶ Decision procedures for special classes of problems; and
- ▶ Checking of proof hints or sketches given by a person.

In the category of general proof search, there were pioneering experiments in the late 1950s by Gilmore, Davis, Putnam, Prawitz, Wang, and others, followed by the more systematic development of practically effective proof procedures, including “tableaux” (Beth, Hintikka), “resolution” (Robinson, Maslov), and “model elimination” (Loveland), as well as more specialized techniques for “equational” reasoning, including “Knuth-Bendix completion.” Perhaps the most famous application of this kind of search is the proof of the Robbins conjecture, discovered by McCune¹⁷ using the automated theorem prover *EQP* in 1996 that settled a problem that had been open since the 1930s.

In the category of decision procedures, perhaps the first real “automated theorem prover” was Davis’s procedure for the special case of Presburger arithmetic, a generalization of integer programming. Implementations of many other decision procedures have followed, motivating further theoretical developments. Some procedures



Although automation is an exciting and ambitious goal, there is little realistic hope of having automated provers routinely prove assertions with real mathematical depth.



have been particularly effective in practice (such as Gröbner basis algorithms and Wu’s method, both applicable to theorem proving in geometry).

Although automation is an exciting and ambitious goal, there is little realistic hope of automated provers routinely proving assertions with real mathematical depth. Attention has thus focused on methods of verification making use of substantial interaction between mathematician and computer.

Interactive Theorem Proving

The idea behind interactive theorem proving is to allow users to work with a computational “proof assistant” to convey just enough information and guidance for the system to be able to confirm the existence of a formal axiomatic proof. Many systems in use today actually construct a formal proof object, a complex piece of data that can be verified by independent checkers.

One of the earliest proof systems was de Bruijn’s *Automath*, which appeared in the late 1960s. The computational assistance it rendered was minimal, since the project’s emphasis was on developing a compact, efficient notation for describing mathematical proof. An early milestone was Jutting’s 1977 Ph.D. thesis in which he presented a complete formalization of Landau’s book on a foundational construction of the real numbers as “Dedekind cuts,” deducing that the reals so constructed are a complete ordered field.

Proof checkers soon came to incorporate additional computer assistance. Andrzej Trybulec’s *Mizar* system, introduced in 1973 and still in use today, uses automated methods to check formal proofs written in a language designed to approximate informal mathematical vernacular. The Boyer-Moore *NQTHM* theorem prover (an ancestor of *ACL2*), also actively used today, was likewise introduced in the early 1970s as a fully automatic theorem prover; in 1974 the project’s efforts shifted to developing methods of allowing users to prove facts incrementally, then provide the facts as “hints” to the automated prover in subsequent proofs.

Influential systems introduced in the 1980s include Robert Constable’s *Nuprl*, Mike Gordon’s *HOL*, and the

Coq system, based on a logic developed by Thierry Coquand and Gerard Huet, and, in the 1990s, Lawrence Paulson's *Isabelle* and the *Prototype Verification System*, or PVS, developed by John Rushby, Natarjan Shankar, and Sam Owre.^a By 1994, William Thurston could write the following in an article in the *Bulletin of the American Mathematical Society*:

"There are people working hard on the project of actually formalizing parts of mathematics by computer, with actually formally correct formal deductions. I think this is a very big but very worthwhile project, and I am confident we will learn a lot from it."²³

Many of these systems are based on an architecture developed by Robin Milner with his 1972 proof LCF proof checker, which implemented Dana Scott's *Logic of Computable Functions*. An LCF-style prover is based on a small, trusted core of code used to construct theorems by applying basic rules of the axiomatic system. Such a system can then include more elaborate pieces of code built on top of the trusted core, to provide more complex proof procedures that internally decompose to (perhaps many) invocations of basic rules. Correctness is guaranteed by the fact that, ultimately, only the basic rules can change the proof state; everything the system does is mediated by the trusted core. (This restriction is often enforced by using a functional programming language like ML and OCaml and implementing the basic inference rules as the only constructors of an abstract data type.)

Many provers support a mode of working where the theorem to be proved is presented as a "goal" that is transformed by applying "tactics" in a backward fashion; for example, Figure 1 is a proof of the fact that every natural number other than 1 has a prime divisor in the *Isabelle* proof assistant.

The "lemma" command establishes the goal to be proved, and the first instruction invokes a form of complete induction. The next two statements split the proof into two cases, depending on whether or not $n = 0$. When $n = 0$,

setting $p = 2$ witnesses the conclusion; the system confirms this automatically (using "auto"). Otherwise, the proof splits into two cases, depending on whether or not n is prime. If n is prime, the result is immediate. The case where n is not prime is handled by appealing to a previously proved fact.

A user can step through such a "procedural" proof script within the proof assistant itself to see how the goal state changes in successive steps. But the script is difficult to read in isolation, since the reader must simulate, or guess, the results of applying each tactic. Ordinary mathematical proofs tend to emphasize, in contrast, the intermediate statements and goals, often leaving the justification implicit. The *Mizar* proof language was designed to model such a "declarative" proof system, and

such features have been incorporated into LCF-style provers. For example, Figure 2 is a proof of the same statement, again in *Isabelle*, but written in a more declarative style.

A number of important theorems have been verified in the systems described here, including the prime number theorem, the four-color theorem, the Jordan curve theorem, the Brouwer fixed-point theorem, Gödel's first incompleteness theorem, Dirichlet's theorem on primes in an arithmetic progression, the Cartan fixed-point theorems, and many more.^b In the next section we describe even more impressive milestones, though even more im-

b For a list of "100 great theorems," including those formalized in various systems, see <http://www.cs.ru.nl/~freek/100/>

Figure 1. An LCF tactic-style proof in *Isabelle*.

```
lemma prime_factor_nat: "n ~= (1::nat) ==> EX p. prime p & p dvd n"
  apply (induct n rule: nat_less_induct)
  apply (case_tac "n = 0")
  apply (rule_tac x = 2 in exI)
  apply auto
  apply (case_tac "prime n")
  apply auto
  apply (subgoal_tac "n > 1")
  apply (frule (1) not_prime_eq_prod_nat)
  apply (auto intro: dvd_mult dvd_mult2)
done
```

Figure 2. A declarative proof in *Isabelle*.

```
lemma prime_factor_nat: "n ~= (1::nat) ==> EX p. prime p & p dvd n"
proof (induct n rule: nat_less_induct)
  fix n :: nat
  assume "n ~= 1" and
  ih: "ALL m < n. m ~= 1 --> (EX p. prime p & p dvd m)"
  then show "EX p. prime p & p dvd n"
  proof -
    { assume "n = 0"
      hence "prime (2 :: nat) & 2 dvd n"
        by auto
      hence ?thesis by blast }
    moreover
    { assume "prime n"
      hence ?thesis by auto }
    moreover
    { assume "n ~= 0" and "~prime n"
      with `n ~= 1` have "n > 1" by auto
      with `~prime n` and not_prime_eq_prod_nat
      obtain m k where "n = m * k" and "1 < m" and "m < n" by blast
      with ih obtain p where "prime p" and "p dvd m" by blast
      with `n = m * k` have ?thesis by auto }
    ultimately show ?thesis by auto
  qed
qed
```

a For a more comprehensive list of provers, see <http://www.cs.ru.nl/~freek/digimath/index.html>; for an overview of 17 proof assistants in use today, see Wiedijk.²⁶

portant are the bodies of mathematical theory that have been formalized, often on the way to proving such “big name” theorems. Substantial libraries have been developed for elementary number theory, real and complex analysis, measure theory and measure-theoretic probability, linear algebra, finite group theory, and Galois theory. Formalizations are now routinely described in journals, including the *Journal of Automated Reasoning*, *Journal of Formalised Reasoning*, and *Journal of Formalized Mathematics*. The annual *Interactive Theorem Proving* conference includes reports on formalization efforts and advances in interactive theorem proving technology.

Also worth noting is that interactive proof systems are also commonly used to verify hardware and software sys-

tinct from those in the verification of ordinary mathematics, and the details would take us too far afield. So, here, in this article, we deliberately set aside hardware and software verification, referring the reader to Donald MacKenzie’s book *Mechanizing Proof: Computing, Risk and Trust*¹⁶ for a thoughtful exploration of the topic.

Contemporary Efforts

Interactive theorem proving reached a major landmark on September 20, 2012, when Georges Gonthier announced he and a group of researchers under his direction had completed a verification of the Feit-Thompson theorem. The project relied on the *Coq* interactive proof assistant and a proof language, *SSReflect*, Gonthier designed. The Feit-Thompson theo-

rem has approximately 150,000 lines of “code,” or formal proof scripts, including 4,000 definitions and 13,000 lemmas and theorems. As a basis for the formalization, Gonthier and his collaborators had to develop substantial libraries of facts about finite group theory, linear algebra, Galois theory, and representation theory. From there, they worked from a presentation of the Feit-Thompson theorem in two texts, one by Helmut Bender and George Glauberman describing the “local analysis,” the other by Thomas Peterfalvi describing a “character-theoretic” component. As one might expect, they had to cope with numerous errors and gaps, some not easy to fix, though none fatal.

Hales’s *Flyspeck* project is another ambitious formalization effort. In response to the outcome of the referee process at the *Annals*, Hales decided to formally verify a proof of the Kepler conjecture. (The name “*Flyspeck*” is a contraction of “Formal Proof of the Kepler Conjecture.”) He made it clear he viewed the project as a prototype:

“In truth, my motivations for the project are far more complex than a simple hope of removing residual doubt from the minds of few referees. Indeed, I see formal methods as fundamental to the long-term growth of mathematics.”⁹

The proof involves three essential uses of computation: enumerating a class of combinatorial structures called “tame hypermaps”; using linear-programming methods to establish bounds on a large number of systems of linear constraints; and using interval methods to verify approximately 1,000 nonlinear inequalities that arise in the proof. All this is in addition to the textual “paper” proof, which in and of itself is quite long and involves Euclidean measure theory, geometric properties of polyhedral, and various combinatorial structures. Partly as a result of the formalization effort, both the “paper” and “machine” parts of proof have been streamlined and reorganized.⁸ The combination of non-trivial paper proofs and substantial time-consuming computational components make it a particularly difficult formalization challenge, yet after a substantial effort by a large, geograph-



tems. In principle, this is no different from verifying mathematical claims; for the purposes of formal verification, hardware and software systems must be described in mathematical terms, and the statement that such a system meets a certain specification is a theorem to be proved. Most of the systems described here are thus designed to serve both goals. The connections run deeper; hardware and software specifications often make sense only against background mathematical theory of, say, the integers or real numbers; and, conversely, methods of verifying software apply to the verification of code that is supposed to carry out specifically mathematical computations. However, hardware and software verification raises concerns largely dis-

rem, sometimes called the odd-order theorem, says every finite group of odd order is solvable; equivalently, that the finite simple groups of odd order are exactly the cyclic groups of prime order. This theorem was an important first step in the classification of finite simple groups mentioned earlier. The original proof by Walter Feit and John Thompson, published in 1963, filled 255 journal pages. While a proof that long would not raise eyebrows today, it was unheard of at the time.

Gonthier launched the project in 2006 with support from the Microsoft Research - Inria Joint Centre in Orsay, France. Because *Coq* is based on a constructive logic, Gonthier had to reorganize the proof in such a way every theorem has a direct computa-

ically distributed team, the project is nearing completion.

Another significant formal verification effort is the *Univalent Foundations* project introduced by Fields Medalist Vladimir Voevodsky.²⁴ Around 2005, Voevodsky, and independently Steve Awodey and Michael Warren, realized that constructive dependent type theory, the axiomatic basis for *Coq*, has an unexpected homotopy-theoretic interpretation. Algebraic topologists routinely study abstract spaces and paths between elements of those spaces; continuous deformations, or “higher-order” paths, between the paths; and deformations between the paths; and so on. What Voevodsky, Awodey, and Warren realized is that one can view dependent type theory as a calculus of topological spaces and continuous maps between them, wherein the assertion $x = y$ is interpreted as the existence of a path between x and y .

More specifically, Voevodsky showed there is a model of constructive type theory in the category of “simplicial sets,” a mathematical structure well known to algebraic topologists. Moreover, this interpretation validates a surprising fact Voevodsky called the “univalence axiom,” asserting, roughly, that any two types that are isomorphic are identical. The axiom jibes with informal mathematical practice, wherein two structures that are isomorphic are viewed as being essentially the same. However, it is notably false in the universe of sets; for example, there is a bijection between the sets $\{1, 2\}$ and $\{3, 4\}$, but the first set contains the element 1 while the second does not. Voevodsky has suggested that dependent type theory with the univalence axiom can provide a new foundation for mathematics that validates structural intuitions. There is also hope among computer scientists that univalence can provide a new foundation for computation where code can be designed to work uniformly on “isomorphic” data structures (such as lists and arrays) implemented in different ways.

In an excerpt from a grant proposal posted on his Web page, Voevodsky described his motivations for the project as follows:

“While working on the completion of the proof of the Block-Kato conjecture I have thought a lot about what to

do next. Eventually I became convinced that the most interesting and important directions in current mathematics are the ones related to the transition into a new era which will be characterized by the widespread use of automated tools for proof construction and verification.”

During the 2012–2013 academic year, the Institute for Advanced Study in Princeton, NJ, held a program on Univalent Foundations, drawing an interdisciplinary gathering of mathematicians, logicians, and computer scientists from all over the world.

Rigor and Understanding

We have distinguished between two types of concern that can attend a mathematical proof: whether the methods it uses are appropriate to mathematics and whether the proof itself represents a correct use of those methods. However, there is yet a third concern that is often raised—whether a proof delivers an appropriate understanding of the mathematics in question. It is in this respect that formal methods are often taken to task; a formal, symbolic proof is for the most part humanly incomprehensible and so does nothing to augment our understanding. Thurston’s article²³ mentioned earlier added the following caveat:

“...we should recognize that the humanly understandable and humanly checkable proofs that we actually do are what is most important to us, and that they are quite different from formal proofs.”

The article was in fact a reply to an article by Jaffe and Quinn¹³ in the *Bulletin of the American Mathematical Society* that proposed a distinction between “theoretical” mathematics, which has a speculative, exploratory character, and fully rigorous mathematics. The Jaffe-Quinn article drew passionate, heated responses from some of the most notable figures in mathematics, some rising to defend the role of rigor in mathematics, some choosing to emphasize the importance of a broad conceptual understanding. Given that both are clearly important to mathematics, the Jaffe-Quinn debate may come across as much ado about nothing, but the episode makes clear that many mathematicians are wary that excessive concern for rigor can displace mathematical understanding.

However, few researchers working in formal verification would claim that checking every last detail of a mathematical proof is the most interesting or important part of mathematics. Formal verification is not supposed to replace human understanding or the development of powerful mathematical theories and concepts. Nor are formal proof scripts meant to replace ordinary mathematical exposition. Rather, they are intended to supplement the mathematics we do with precise formulations of our definitions and theorems and assurances that our theorems are correct. One need only recognize, as we say here, that verifying correctness is an important part of mathematics. The mathematical community today invests a good deal of time and resources in the refereeing process in order to gain such assurances, and surely any computational tools that can help in that regard should be valued.

The Quest for Certainty


In discussions of formally verified mathematics, the following question often arises: Proof assistants are complex pieces of software, and software invariably has bugs, so why should we trust such a program when it certifies a proof to be correct?

Proof assistants are typically designed with an eye toward minimizing such concerns, relying on a small, trusted core to construct and verify a proof. This design approach focuses concern on the trusted core, which consists of, for example, approximately 400 lines in Harrison’s *HOL light* system. Users can obtain a higher level of confidence by asking the system to output a description of the axiomatic proof that can be checked by independent verifiers; even if each particular verifier is buggy, the odds that a faulty inference in a formal proof can make it past multiple verifiers shrinks dramatically. It is even possible to use formal methods to verify the trusted core itself. There have been experiments in “self-verifications” of simplified versions of *Coq* by Bruno Barras and *HOL light* by Harrison,¹⁰ as well as Jared Davis’s work on *Milawa*, a kind of bootstrapping sequence of increasingly powerful approximations to the *ACL2* prover.


To researchers in formal verification, however, these concerns seem

misplaced. When it comes to informal proof, mistakes arise from gaps in the reasoning, appeal to faulty intuitions, imprecise definitions, misapplied background facts, and fiddly special cases or side conditions the author failed to check. When verifying a theorem interactively, users cannot get away with any of this; the proof checker keeps the formalizer honest, requiring every step to be spelled out in complete detail. The very process of rendering a proof suitable for machine verification requires strong discipline; even if there are lingering doubts about the trustworthiness of the proof checker, formal verification delivers a very high degree of confidence—much higher than any human referee can offer without machine assistance.

Mathematical results obtained through extensive computation pose additional challenges. There are at least three strategies that can be used to verify such results. First, a user can rewrite the code that carries out the calculations so it simultaneously uses the trusted core to chain together the axioms and rules that justify the results of the computation. This approach provides, perhaps, the highest form of verification, since it produces formal axiomatic proofs of each result obtained by calculation. This is the method being used to verify the linear and nonlinear inequalities in the *Flyspeck* project.²² The second strategy is to describe the algorithm in mathematical terms, prove the algorithm correct, then rely on the trusted core to carry out the steps of the computation. This was the method used by Gonthier in the verification⁵ of the four-color theorem in *Coq*; because it is based on a constructive logic, *Coq*'s “trusted computing base” is able to normalize terms and thereby carry out a computation. The third strategy is to describe the algorithm within the language of the proof checker, then extract the code and run it independently. This method was used by Tobias Nipkow, Gertrud Bauer, and Paula Schultz¹⁹ to carry out the enumeration of finite tame hypermaps in the *Flyspeck* project; ML code was extracted from formal definitions automatically and compiled. This approach to verifying the results of computation invokes additional layers of trust, that, for example, the extracted code is faith-



Many mathematicians are wary that excessive concern for rigor can displace mathematical understanding.



ful to the function described in axiomatic terms and the compiler respects the appropriate semantics. However, compared to using unverified code, this method provides a high degree of confidence as well.

Similar considerations bear on the use of automated methods and search procedures to support the formalization process; for example, one can redesign conventional automated reasoning procedures so they generate formal proofs as they proceed or invoke an “off-the-shelf” reasoning tool and try to reconstruct a formal proof from the output. With respect to both reasoning and computation, an observation that often proves useful is that many proof procedures can naturally be decomposed into two steps: a “search” for some kind of certificate and a “checking” phase where this certificate is verified.² When implementing these steps in a foundational theorem prover, the “finding” (often the difficult part) can be done in any way at all, even through an external tool (such as a computer algebra system¹¹), provided the checking part is done in terms of the logical kernel; for example, linear programming methods can provide easily checked certificates that witness the fact that a linear bound is optimal. Similarly, semi-definite programming packages can be used to obtain certificates that can be used to verify nonlinear inequalities.²⁰ Along these lines, the *Flyspeck* project uses optimized, unverified code to find informative certificates witnessing linear and nonlinear bounds, then uses the certificates to construct fully formal justifications.²² Such practices raise interesting theoretical questions about which symbolic procedures can in principle provide efficiently checkable certificates, as well as the pragmatic question of how detailed the certificates should be to allow convenient verification without adversely affecting the process of finding them.

Prospects

We have not touched on many important uses of computers in mathematics (such as in the discovery of new theorems, exploration of mathematical phenomena, and search for relevant information in databases of mathematical facts). Correctness is only one important part of mathematics, as we have

emphasized, and the process of verification should interact continuously with other uses of formal methods. But even with this restriction, the issues we have considered touch on important aspects of artificial intelligence, knowledge representation, symbolic computation, hardware and software verification, and programming-language design. Mathematical verification raises its own challenges, but mathematics is a quintessentially important type of knowledge, and understanding how to manage it is central to understanding computational systems and what they can do.

The developments we have discussed make it clear that it is pragmatically possible to obtain fully verified axiomatic proofs of substantial mathematical theorems. Despite recent advances, however, the technology is not quite ready for prime time. There is a steep learning curve to the use of formal methods, and verifying even straightforward and intuitively clear inferences can be time consuming and difficult. It is also difficult to quantify the effort involved. For example, 15 researchers contributed to the formalization of the Feit-Thompson theorem over a six-year period, and Hales suspects the *Flyspeck* project has already exceeded his initial estimate of 20 person-years to completion. However, it is ultimately difficult to distinguish time spent verifying a theorem from time spent developing the interactive proof system and its libraries, time spent learning to use the system, and time spent working on the mathematics proper.

One way to quantify the difficulty of a formalization is to compare its length to the length of the original proof, a ratio known as the “de Bruijn factor.” Freek Wiedijk carried out a short study²⁵ and, in the three examples considered, the ratio hovers around a factor of four, whether comparing the number of symbols in plain-text presentations or applying a compression algorithm first to obtain a better measure of the true information content.

A better measure of difficulty is the amount of time it takes to formalize a page of mathematics, though that can vary dramatically depending on the skill and expertise of the person carrying out the formalization, density of the material, quality and depth of the supporting library, and formal-

izer’s familiarity with that library. In the most ideal circumstances, an expert can handle approximately a half page to a page of a substantial mathematical text in a long, uninterrupted day of formalizing. But most circumstances are far less than ideal; an inauspicious choice of definitions can lead to hours of fruitless struggle with a proof assistant, and a formalizer often finds elementary gaps in the supporting libraries that require extra time and effort to fill.

We should not expect interactive theorem proving to be attractive to mathematicians until the time it takes to verify a mathematical result formally is roughly commensurate with the time it takes to write it up for publication or the time it takes a referee to check it carefully by hand. Within the next few years, the technology is likely to be most useful for verifying proofs involving long and delicate calculations, whether initially carried out by hand or with computer assistance. But the technology is improving, and the work of researchers like Hales and Voevodsky makes it clear that at least some mathematicians are interested in using the new methods to further mathematical knowledge.

In the long run, formal verification efforts need better libraries of background mathematics, better means of sharing knowledge between the various proof systems, better automated support, better means of incorporating and verifying computation, better means of storing and searching for background facts, and better interfaces, allowing users to describe mathematical objects and their intended uses, and otherwise convey their mathematical expertise. But verification need not be an all-or-nothing proposition, and it may not be all that long before mathematicians routinely find it useful to apply an interactive proof system to verify a key lemma or certify a computation that is central to a proof. We expect within a couple of decades seeing important pieces of mathematics verified will be commonplace and by the middle of the century may even become the new standard for rigorous mathematics. □

References

1. Aubrey, J. *Brief Lives*. A. Clark, Ed. Clarendon Press, Oxford, U.K., 1898.
2. Blum, M. Program result checking: A new approach

- to making programs more reliable. In *Proceedings of the 20th International Colloquium on Automata, Languages and Programming* (Lund, Sweden, July 5–9), A. Lingas, R. Karlsson, and S. Carlsson, Eds. Springer, Berlin, 1993, 1–14.
3. Bourbaki, N. *Elements of Mathematics: Theory of Sets*. Addison-Wesley, Reading, MA, 1968; translated from the French *Théorie des ensembles* in the series *Éléments de mathématique*, revised version, Hermann, Paris, 1970.
4. Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik* 38, 1 (1931), 173–198; reprinted with English translation in Kurt Gödel: *Collected Works, Volume 1*, S. Feferman et al., Eds. Oxford University Press, Oxford, U.K., 1986, 144–195.
5. Gonthier, G. Formal proof—The four-color theorem. *Notices of the American Mathematical Society* 55, 11 (Dec. 2008), 1382–1393.
6. Grabner, J.V. Is mathematical truth time-dependent? In *New Directions in the Philosophy of Mathematics: An Anthology*, T. Tymoczek, Ed., Birkhäuser, Boston, 1986, 201–213.
7. Gracar, J.F. Errors and corrections in the mathematical literature. *Notices of the American Mathematical Society* 60, 4 (Apr. 2013), 418–432.
8. Hales, T. *Dense Sphere Packings: A Blueprint for Formal Proofs*. Cambridge University Press, Cambridge, U.K., 2012.
9. Hales, T. *The Kepler Conjecture* (unpublished manuscript), 1998; <http://arxiv.org/pdf/math/9811078.pdf>
10. Harrison, J. Towards self-verification of HOL Light. In *Proceedings of the Third International Joint Conference in Automated Reasoning* (Seattle, Aug. 17–20), U. Furbach and N. Shankar, Eds. Springer, Berlin, 2006, 177–191.
11. Harrison, J. and Théry, L. A sceptic’s approach to combining HOL and Maple. *Journal of Automated Reasoning* 21, 3 (1998), 279–294.
12. Hobbes, T. *Leviathan*. Andrew Crooke, London, 1651.
13. Jaffe, A. and Quinn, F. “Theoretical mathematics”: Toward a cultural synthesis of mathematics and theoretical physics. *Bulletin of the American Mathematical Society (New Series)* 29, 1 (1993), 1–13.
14. Littlewood, J.E. *Littlewood’s Miscellany*. Cambridge University Press, Cambridge, U.K., 1986.
15. Mac Lane, S. *Mathematics: Form and Function*. Springer, New York, 1986.
16. MacKenzie, D. *Mechanizing Proof: Computing, Risk and Trust*. MIT Press, Cambridge, MA, 2001.
17. McCune, W. Solution of the Robbins problem. *Journal of Automated Reasoning* 19, 3 (1997), 263–276.
18. Nathanson, M. Desperately seeking mathematical truth. *Notices of the American Mathematical Society* 55, 7 (Aug. 2008), 773.
19. Nipkow, T., Bauer, G., and Schultz, P. Flyspeck I: Tame graphs. In *Proceedings of the Third International Joint Conference in Automated Reasoning* (Seattle, Aug. 17–20). Springer, Berlin, 2006, 21–35.
20. Parrilo, P.A. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming* 96, 2 (2003), 293–320.
21. Schüring, G. Zur Modernisierung des Studiums der Mathematik in Berlin, 1820–1840. In *Amphora: Festschrift für Hans Wussing Zu Seinem 65. Geburtstag*, S.S. Demidov et al., Eds. Birkhäuser, Basel, 1992, 649–675.
22. Solovyev, A. *Formal Computation and Methods*. Ph.D. thesis, University of Pittsburgh, Pittsburgh, PA, 2012.
23. Thurston, W.P. On proof and progress in mathematics. *Bulletin of the American Mathematical Society (New Series)* 30, 2 (1994), 161–177.
24. Univalent Foundations Program, Institute for Advanced Study. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Princeton, NJ, 2013; <https://github.com/HoTT/book>
25. Wiedijk, F. *The de Bruijn Factor* (unpublished manuscript), 2000; <http://www.cs.ru.nl/~freek/factor/>
26. Wiedijk, F. *The Seventeen Provers of the World*. Springer, Berlin, 2006.

Jeremy Avigad (avigad@cmu.edu) is a professor in the Department of Philosophy and the Department of Mathematical Sciences at Carnegie Mellon University, Pittsburgh, PA.

John Harrison (johnh@chips.intel.com) is a principal engineer in Intel Corporation, Hillsboro, OR.

DOI:10.1145/2591013

Scala unifies traditionally disparate programming-language philosophies to develop new components and component systems.

BY MARTIN ODERSKY AND TIARK ROMPF

Unifying Functional and Object-Oriented Programming with Scala

THOUGH IT ORIGINATED as an academic research project, Scala has seen rapid dissemination in industry and open source software development. Here, we give a high-level introduction to Scala and look to explain what makes it appealing for developers. The conceptual development of Scala began in 2001 at École polytechnique fédérale de Lausanne (EPFL) in Switzerland. The first internal version of the language appeared in 2003 when it was also taught in an undergraduate course on functional programming. The first public release was in 2004, and the 2.x series

in 2006, with slightly redesigned language and a new compiler, written completely in Scala itself. Shortly thereafter, an ecosystem of open-source software began to form around it, with the Lift Web framework as an early crystallization point. Scala also began to be used in industry. A well-known adoption was Twitter, which aimed (in 2008) to rewrite its own message queue implementation in Scala. Since then, much of its core software has been written in Scala. Twitter has contributed back to open source in more than 30 released projects²⁴ and teaching materials.²⁵ Many other companies have followed suit, including LinkedIn, where Scala drives the social graph service, Klout, which uses a complete Scala stack, including the Akka distributed middleware and Play Web framework, and Foursquare, which uses Scala as the universal implementation language for its server-side systems. Large enterprises (such as Intel, Juniper Networks, and Morgan Stanley) have also adopted the language for some of their core software projects.

Gaining broad adoption quickly is rare for any programming language, especially one starting in academic research. One can argue that at least some of it could be due to circumstantial factors, but it would still be interesting to ponder what properties of the language programmers find so attractive. There are two main ingredients: First, Scala is a pragmatic language. Its main focus is to make developers more productive. Productivity needs access to a large set of libraries and tools and is why Scala was designed from the start to interoperate well with Java and run efficiently on the JVM. Almost all

» key insights

- **Scala shows that functional and object-oriented programming fit well together.**
- **This combination allows a smooth transition from modeling to efficient code.**
- **Scala also offers an impressive toolbox for expressing concurrency and parallelism.**

Java libraries are accessible from Scala without needing wrappers or other glue code. Designing Scala libraries so they can be accessed from Java code is also relatively straightforward. Moreover, Scala is a statically typed language that compiles to the same bytecodes as Java and runs at comparable speed.⁶

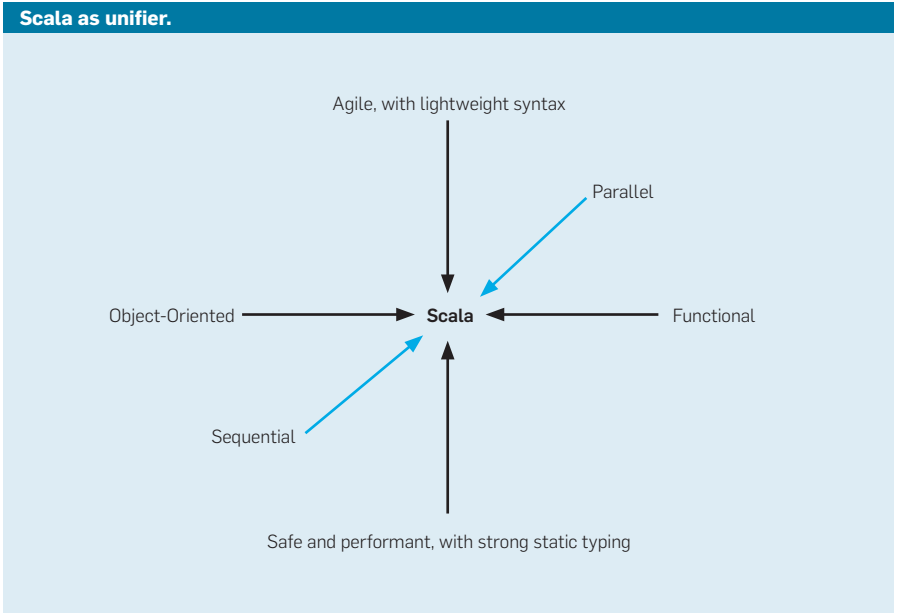
Several necessary compromises followed from the interoperability goal; for instance, Scala adopts Java's method overloading scheme, even though one could say multi-methods in the style of Fortress¹ would have been cleaner. Another is that Scala allows null pointers, called by their originator, Tony Hoare, the "billion-dollar mistake." This is again important for interoperability with Java. However, in pure Scala code null pointers are usually avoided in favor of the standard Option type.

The second ingredient for Scala's impressive adoption history is that it rides, and to a certain degree drives, the emerging trend of combining functional and object-oriented programming. Functional programming has emerged since the mid-2000s as an attractive basis for software construction. One reason is the increasing importance of parallelism and distribution in computing. In the world of software development where updates require logs or replication for consistency, it is often more efficient to use replayable operations on immutable data instead; prominent examples are parallel toolkits like Akka¹⁸ and Spark.²

Scala unifies areas of computing that were traditionally disparate (see the figure here). Its integration of functional and object-oriented concepts leads to a scalable language, in the sense that the same concepts work well for very small, as well as very large, programs. Scala was voted the most popular scripting language on the JVM at the JavaOne conference 2012. This was surprising, as scripting languages are usually dynamically typed, whereas Scala has an expressive, precise static type system, relying on local type inference^{14,16} to obviate



Scala stairs at École polytechnique fédérale de Lausanne inspired the Scala logo (<http://www.scala-lang.org>) designed by Gilles Dubochet.



the need for most annoying type annotations. The type system and good performance characteristics of its implementation make it suitable for large mission-critical back-end applications, particularly those involving parallelism or concurrency.

Every piece of data in Scala is conceptually an object and every operation a method call. This is in contrast to functional languages in the ML family that are stratified into a core language and a module system. Scala's approach leads to an economy of features, keeping the language reasonably small in spite of its multi-paradigm nature. The functional object system also enables construction of high-level, flexible libraries that are easy for programmers to use; for instance, its hierarchy of collection classes provides a uniform framework¹² for sequences, sets, and maps that systematically cover multiple dimensions, from immutable to mutable and sequential to parallel¹⁷ and (for sequences) from strict to lazy evaluation.

Here, we demonstrate Scala through a series of examples, starting with simple program fragments for how it identifies features from functional and object-oriented programming, then how its object model absorbs common concepts from module systems to achieve modularity and abstraction. This also demonstrates how Scala programmers can progress from a simple, high-level model to efficient sequential and parallel implementations of the model.

Combining Features

Scala combines features from the object-oriented and functional paradigms in new and interesting ways. As an example of the object-oriented style, consider the following definition of a simple class of `Persons`:

```
class Person(val name:
String, val age: Int) {
  override def toString =
    s"$name ($age)"
}
```

An instance of this class include fields `name` and `age` and provides an overridden `toString` implementation that returns a `String` describing the object. The syntax `s"..."` denotes an interpolated string that can contain computed expressions following the escape character `$`. As an example of the functional style, here is a way to split a list of persons according to their age:

```
val persons: List[Person] = ...
val (minors, adults) = persons.partition(_ .age < 18)
```

The `partition` method takes a sequence and a predicate and returns a pair of sequences, one consisting of elements that satisfy the predicate, the other of elements that do not. The notation `_ .age < 18` is shorthand for the anonymous function `x => x.age < 18`.

This is just one of many ways one can act on collections of objects through powerful purely functional

combinators. Compared to equivalent Java code, the Scala version is much more succinct. While brevity is not universally good, the difference in cognitive overhead in this example is striking.

Instead of objects and inheritance, traditional functional languages often have algebraic data types that can be decomposed with pattern matching. An algebraic data type consists of a fixed number of alternatives, each with a different constructor. Scala allows pattern matches over class instances, obviating the need for another fundamental type constructor besides classes while keeping the language simpler and more uniform.

As an example, consider the task of writing a type named `Try` that contains either a returned value or an exception. A standard usage of a type like `Try` is to communicate exceptions across thread or machine boundaries. Consider a small use case: Assume we have a function `checkAge` that checks whether a person is of legal age and throws an exception if not. We may want to use `Try` like this:

```
Try {
  checkAge(person)
  fetchRestrictedContent()
}
```

At a glance, this code looks similar to familiar `try/catch` exception handling. However, the `catch` block is missing, and the result of the whole expression is either a `Success` instance carrying the result of `fetchRestrictedContent` or a `Failure` carrying an exception. While `try/catch` decouples the handling of error conditions from the location of the error in the program, `Try` also decouples the handling from the time the error occurred and its physical location; if failures happen on, say, a Web server in response to a client request, we can collect all failures relating to the request, batch them up, and send them back to the client.

Here is a possible definition of type `Try` followed by an explanation of the `Try { ... }` syntax:

```
trait Try[T] {
  def get: T
}
```



```

case class Success[T](value: T)
extends Try[T] {
  def get = value
}
case class Failure[T](ex: Ex-
ception) extends Try[T] {
  def get = throw ex
}

```

A “trait” is a generalization of Java’s interface that can contain both abstract and concrete methods. The trait `Try` in this code defines a single abstract method `get`; we add more methods to it later. The trait is parameterized with a type parameter `T` meant to indicate the type of any returned value.

Two subclasses extend `Try`: `Success` and `Failure`. The case modifier in the class definitions enables pattern matching and also adds convenience methods to these classes. The `Success` class takes as a parameter the returned value, and the `Failure` class takes as parameter the thrown exception.

Consider the following client code that processes a `Try` value:

```

val x: Try[Int] = ...
x match {
  case Success(v) =>
    println(s"OK: $v")
  case Failure(ex: IOException)
=>
  println(s"I/O error")
  case Failure(ex) =>
    println(s"Other error $ex")
}

```

Pattern matching in Scala is done in `match` expressions, which are conceptually generalizations of the `switch` statement in C and Java. A selector value (the `x` to the left of `match`) is matched against a number of cases, each consisting of a pattern followed by an arrow `=>` and an expression that defines the result value in case the pattern matches. The expression here includes three patterns: The first matches any value of form `Success(x)` where `x` is arbitrary; the second matches any value of form `Success(ex)` where `ex` is of type `IOException`; and the third matches all other `Failures`. In all three, the result expression is a `println` invocation. The return type of `println` is `Unit`, which is inhabited by the single atomic value `()`.

Scala’s integration of functional and object-oriented concepts leads to a scalable language, in the sense that the same concepts work well for very small, as well as very large, programs.

The pattern-matching syntax and semantics is standard for a functional language. New is that pattern matching applies to object types, not algebraic data types; for instance, the second pattern matches at the same time on a `Failure` alternative with an embedded `IOException` value. Dealing with open types (such as exceptions) has been a headache for standard functional pattern matching, which applies only to closed sums with a fixed number of alternatives. Scala sidesteps this issue by matching on object hierarchies instead. Having alternatives like `Success` and `Failure` be first-class types (as opposed to only `Try`) gives the dual benefits of uniformity and expressiveness.

A possible criticism against this scheme is that the open-world assumption means one cannot conclusively verify that a pattern match is exhaustive, or contains a pattern for every possible selector value. Scala caters to this case by allowing sealed classes. We thus could have declared `Try` like this:

```

sealed trait Try[T] ...

```

In this case all immediate subclasses of `Try` need to be defined together with it, and exhaustiveness checking for `Try` expressions would be enabled; that is, a sealed trait with some extending case classes behaves like an algebraic data type.

Another criticism levied against Scala’s scheme is that a set of case class definitions tends to be more verbose than the definition of an algebraic data type. This is true but must be weighed against the fact that in a typical real-world system the amount of type definitions should be small compared to the amount of code that operates on the types.

We have seen the definition of type `Try` but still lack a convenient way of creating `Try` values from expressions that can throw exceptions, as in, say:

```

Try { readFile(file) }

```

We can achieve this through the following definition:

```

object Try {
  def apply[T](expr: => T) =

```

Code section 1: Graph signature.

```

trait Graphs {
  type Node
  type Edge
  def pred(e: Edge): Node
  def succ(e: Edge): Node
  type Graph <: GraphSig
  trait GraphSig {
    def nodes: Set[Node]
    def edges: Set[Edge]
    def outgoing(n: Node): Set[Edge]
    def incoming(n: Node): Set[Edge]
    def sources: Set[Node]
    def topSort: Seq[Node]
    def subGraph(nodes: Set[Node]): Graph =
      newGraph(nodes, edges filter (e =>
        (nodes contains pred(e)) &&
        (nodes contains succ(e))))
  }
  def newGraph(nodes: Set[Node], edges: Set[Edge]): Graph
}

```

```

try Success(expr)
catch {
  case ex: Throwable =>
    Failure(ex)
}

```

It includes several noteworthy points, and what gets defined is an object named `Try`. An object is a singleton instance that implements a set of definitions; in the example, `Try` implements a method `apply`. Object definitions are Scala's replacement of the concept of static members, which, despite being a common feature, are decidedly non-object-oriented. Instead of defining static values in a class, we define a singleton object containing the same values. A common pattern in Scala combines an object definition and a class or trait definition with the same name in a single source file. In this case, the object members are treated like the static members of a Java class.

The `Try` object defines a method `apply` that takes a type parameter `T` and a value parameter `expr` of type `=> T` that describes a computation of type `T`. The type syntax `=> T` describes "by-name" parameters. As indicated by the syntax, such parameters can be thought of as functions without an argument list. Expressions passed to them will not be evaluated at the call site; instead, they will be evaluated each time the parameter is dereferenced in the called function. Here is an example:

```
Try.apply(x / y)
```

In this case, the expression `x / y` will be passed unevaluated into the `apply` method. The `apply` method will then evaluate it at the point where the `expr` parameter is referenced as part of its `Success` result. If `y` is zero, this evaluation yields a division-by-zero error that will be caught in the enclosing `try/catch` expression and a `Failure` value will be returned.

The `Try` expression can be written even shorter, like this:

```
Try(x / y)
```

Or, if blocks of multiple statements are used, like this:

```
Try { ... }
```

This code makes it look as if `Try` is a function that can be applied to an argument and that yields a value of type `Try`. It also makes use of another cornerstone of Scala: Every object with an `apply` method can be used as a function value. This is Scala's way of making functions first class; they are simply interpreted as objects with `apply` methods.

Scala includes function types, written in double-arrow notation (such as `Int => String` is the type of functions from `Int` to `String`). But this type is simply a syntactic abbreviation for the object type `Function1[Int, String]`. `Function1` is defined as a trait along the following lines:

```

trait Function1[S, T] {
  def apply(x: S): T
}

```

One characteristic of good functional programming style is the definition of combinator libraries that take a data type and compose it in interesting ways. What are useful combinators for `Try` values? One obvious choice is the combinator `onSuccess`, which takes a `Try` value and runs another `Try`, returning function on its result, if the value was a success. A failure, on the other hand, would be returned directly.

The classic object-oriented way of implementing `apply` would rely on dynamic dispatch: Define an abstract method in trait `Try` and one implementing method in each subclass. Alternatively, we can use pattern matching, defining the `onSuccess` operator as a single method on trait `Try`:

```

sealed trait Try[T] {
  def onSuccess[U](f: T => Try[U]): Try[U] = this match
  {
    case Success(x) => f(x)
    case failure => failure
  }
  ...
}

```

The `onSuccess` combinator method can be used like this:

```
Try(x/y).onSuccess(z => Try(1/z))
```

Here, the argument to the `onSuccess` call is an anonymous function that takes its parameter `z` to the left of the double arrow `=>` and returns the result of evaluating the expression to the right of the arrow. Evaluation of the whole expression first divides `x` by `y` and, if successful, returns a `Success` value containing the inverse of the result `z`. Any arithmetic exceptions lead to `Failure` results. The code is thus equivalent to:

```
Try { val z = x/y; 1/z } === Try(1/(x/y))
```

More generally, `x.onSuccess(z=>Try(f(z)))` is equivalent to `Try(f(x.get))`. Both notational vari-

ants are useful for various purposes. The `onSuccess` combinator is most useful when we want to combine several operations to feed into each other, without assigning explicit names to the intermediate results; `onSuccess` corresponds to the pipe operator in Unix-like shells. We can define an `exec` command to launch processes:

```
def exec(cmd: String): (in: Buffer) => Try[Buffer] = ...
```

And use `onSuccess` to connect inputs and outputs, given some fixed input data `stdin`:

```
stdin.onSuccess(exec("ls"))
  .onSuccess(exec("grep ^.*\.scala"))
  .onSuccess(exec("xargs scalac"))
```

The simple all-or-nothing failure model is different from actual OS pipes; here, each command either succeeds after producing its output or fails without producing output. There is no failure after producing partial output, and output cannot be read by the next process until the previous process has terminated with success or failure. However, these features are easy to add with a level of indirection. Those familiar with the Haskell school of functional programming⁸ will notice that `Try` implements the exception monad and `onSuccess` “bind” operation.

Syntactically, most Scala programmers would write the expression a bit differently, like this:

```
(stdin onSuccess exec("ls")
  onSuccess exec("grep
^.*\.scala")
  onSuccess exec("xargs
scalac"))
```

This style makes use of another pervasive Scala principle: Any binary infix operator is expanded to a method call on one of its arguments. Most operators translate to a method call with the left operand as receiver; the only exception are operators ending in `:`, which are resolved to the right. An example is the list `cons` operator `::`, which prepends an element to the left of a list, as in `1::xs`.

The main benefit of this Scala convention is its regularity. There are no special operators in the Scala syntax.

Even operators (such as `+` and `-`) are conceptually method calls. So we could redefine `onSuccess` using the pipe char `|` as follows:

```
stdin | exec("ls") |
exec("grep ^.*\.scala") |
exec("xargs scalac")
```

Scaling Up

The previous section explored the combination of functional and object-oriented programming in a limited example where we defined one concrete type and several operations. Here, we show how Scala also applies to larger program structures, including interfaces, high-level models for specifications, and lower-level implementations.

The task is to find a general model for graphs. Abstractly, a graph is a structure consisting of nodes and edges. In practice, there are many kinds of graphs that differ in the types of information that are attached to the nodes and edges; for example, nodes might be cities and edges roads, or nodes might be persons and edges relationships.

We would like to capture what graphs have in common using one abstract structure that can then be augmented with models and algorithms in a modular way.

Code section 1 shows a trait for graphs. Simple as it is, the definition of this trait is conceptually similar to how the social graph is modeled in Scala at LinkedIn.

The trait has two abstract type members: `Node` and `Edge`. Scala is one of the few languages where objects can

have not only fields and methods but also types as members. The type declarations postulate that concrete subclasses of the trait `Graphs` will contain some definitions of types `Node` and `Edge`. The definitions themselves are arbitrary, as long as implementations of the other abstract members of `Graphs` exist for them.

Next come two abstract method definitions specifying that every `Graph` will define methods `pred` and `succ` that take an `Edge` to its predecessor and successor `Node`. The definition of the methods is deferred to concrete subclasses.

The type of `Graph` itself is then specified. This is an abstract type that has as upper bound the trait `GraphSig`; that is, we require that any concrete implementation of type `Graph` conforms to the trait. `GraphSig` defines a set of fields and methods that apply to every graph; the two essential ones are the set of nodes and the set of edges. Furthermore, here are three convenience methods:

- ▶ For each node the set of outgoing and incoming edges;
- ▶ The set of sources, or nodes that do not have incoming edges; and
- ▶ A method `subGraph` that takes a set of nodes and returns a new graph consisting of these nodes and any edges of the original graph that connect them.

To show some more interesting algorithmic treatment of graphs, we also include a method `topSort` for the topological sorting of an acyclic graph. The method returns in a list a total

Code section 2: Graph model.

```
abstract class GraphsModel extends Graphs {
  class Graph(val nodes: Set[Node], val edges: Set[Edge])
    extends GraphSig {
    def outgoing(n: Node) = edges filter (pred(_) == n)
    def incoming(n: Node) = edges filter (succ(_) == n)
    lazy val sources = nodes filter (incoming(_).isEmpty)
    def topSort: Seq[Node] =
      if (nodes.isEmpty) List()
      else {
        require (sources.nonEmpty)
        sources.toList ++
          subGraph(nodes -- sources).topSort
      }
  }
  def newGraph(nodes: Set[Node], edges: Set[Edge]) =
    new Graph(nodes, edges)
}
```

Code section 3: Graph implementation.

```

abstract class GraphsImpl extends Graphs {
  class Graph(val nodes: Set[Node],
              val edges: Set[Edge]) extends GraphSig {
    private val outEdges, inEdges =
      new mutable.HashMap[Node, Set[Edge]] {
        override def default(key: Node) = Set()
      }
    for (e <- edges) {
      inEdges(succ(e)) += e
      outEdges(pred(e)) += e
    }
    def outgoing(n: Node) = outEdges(n)
    def incoming(n: Node) = inEdges(n)
    def topSort: Seq[Node] = {
      val indegree = new mutable.HashMap[Node, Int]
      val sorted = new mutable.ArrayBuffer[Node]
      for (x <- nodes) {
        indegree(x) = inEdges(x).size
        if (indegree(x) == 0) sorted += x
      }
      var frontier = 0
      while (frontier < sorted.length) {
        for (e <- outEdges(sorted(frontier))) {
          val x = succ(e)
          indegree(x) -= 1
          if (indegree(x) == 0) sorted += x
        }
        frontier += 1
      }
      sorted
    }
  }
  def newGraph(nodes: Set[Node], edges: Set[Edge]) =
    new Graph(nodes, edges)
}

```

ordering of nodes in the graph consistent with the partial ordering implied by the graph edges; that is, nodes are ordered in the list in such a way the predecessor of every graph edge appears before its successor.

Unlike Java interfaces, Scala traits can have abstract, as well as concrete, members; to illustrate, trait `GraphSig` defines `subGraph` as a concrete method, whereas the other methods and fields are left abstract.

Finally, `Graphs` declares an abstract factory method `newGraph` that constructs an instance of type `Graph` from a set of nodes and a set of edges. Note because no concrete definition of `Graph` is given, `newGraph` cannot be defined as a concrete method at the level of `Graphs` because one does not know at this level how to construct a `Graph`. The method must be abstract.

How can `Graphs` be implemented? A wide range of solutions is possible. We start with a high-level model given in code section 2.

`GraphsModel` defines two concrete members: the `Graph` class and the `newGraph` factory method.

The `Graph` class takes as parameters the sets of nodes and edges that make up the graph. Note both parameters are prefixed with `val`. This syntax turns each parameter into a class field that serves as concrete definition of the corresponding parameter in `GraphSig`. With nodes and edges given, the incoming and outgoing methods can be defined as simple filter operations on sets. As the name implies, a filter operation on a collection forms a new collection that retains all elements of the original collection that satisfy a certain predicate. Note the shorthand method syntax used in these predicates where the underscore marks a parameter position; for instance `(pred(_)) == n` is shorthand for the anonymous function `x => pred(x) == n`.

Here are two other commonly used collection transformers:

- ▶ `xs map f`, which applies a function `f` to each element of a collection `xs` and forms a collection of the results; and

- ▶ `xs flatMap f`, which applies the collection-valued function `f` to each element of a collection `xs` and combines its results in a new collection using a `concat` or `union` operation.

The next definition defines `sources` as a lazy `val`, meaning the defining expression for `sources` will not be executed at object initialization but the first time somebody accesses the value of `sources` (possibly never). However, after the first access, the value will be cached, so repeated evaluations of `sources` are avoided. Lazy evaluation is a powerful technique for saving work in purely functional computations. The Haskell programming language makes laziness the default everywhere. Since Scala can be both imperative and functional, it has instead opted for an explicit lazy modifier.

The `topSort` method is a bit longer but still straightforward. If the graph is empty, the result is the empty list; otherwise, the result consists of a list containing all sources of the graph, followed by the topological sort of the subgraph formed by all other nodes. The `++` operation concatenates two collections. The implementation in code section 2 follows this specification to the word and is therefore obviously correct.

One case that still needs consideration is what to return if the graph contains cycles. No topological sorting can exist in it. We model this case through a `require` clause in the `topSort` method, specifying that every non-empty graph to be sorted must have a non-empty set of sources. If this requirement does not hold at any stage of the recursive algorithm, a graph with only cycles is left, and `require` will throw an exception. For implementations in the following sections, a restriction to acyclic graphs is no longer explicitly checked. Rather, we assume it as part of the implicit contract of `topSort`.

The final method to define is `newGraph`, trivial once the definition of `Graph` is fixed.

At the level of `GraphsModel`, there is still no definition of the `Node` and `Edge` types; they can be arbitrary. Here is a concrete object `myGraphModel`

that inherits from `GraphsModel`, defines the `Node` type to be a `Person`, and defines the `Edge` type to be a pair of persons.

```
object myGraphModel extends
GraphsModel {
  type Node = Person
  type Edge = (Person, Person)
  def succ(e: Edge) = { val (s,
p) = e; s }
  def pred(e: Edge) = { val (s,
p) = e; p }
}
```

The situation where edges are pairs of nodes, for whatever the definition of node is, appears quite common. To avoid repetitive code, we can factor out this concept into a separate trait like this:

```
trait EdgesAsPairs extends
Graphs {
  type Edge = (Node, Node)
  def succ(e: Edge) = { val (s,
p) = e; s }
  def pred(e: Edge) = { val (s,
p) = e; p }
}
```

Trait `EdgesAsPairs` extends `Graphs` with a definition of `Edge` as a pair of `Node` and the corresponding definitions of `pred` and `succ`. Using this trait we can now shorten the definition of `myGraphModel` as follows:

```
object myGraphModel extends
GraphsModel with EdgesAsPairs {
  type Node = Person
}
```

The combination of two traits with the `with` connective is called “mixin composition.” An important aspect of mixin composition in Scala is that it is multi-way; that is, any trait taking part in the composition can define the abstract members of any other trait, independent of the order in which the traits appear in the composition. For instance, the `myGraphModel` object in the composition defines the type `Node`, referred to in `EdgesAsPairs`, whereas the latter trait defines `Edge`, referred to in `GraphsModel`. In this sense, mixin composition is symmetric in Scala. Order of traits still matters for determining initialization order, resolving

super calls, and overriding concrete definitions. As usual, they are defined through a linearization of the mixin composition graph.¹¹

More efficient implementation.

The `GraphsModel` class is concise and correct by construction but would win no speed record. Code section 3 presents a much faster implementation of `Graphs`, satisfying the same purely functional specification as `GraphsModel` yet relies on mutable state internally. This illustrates an important aspect of the Scala “philosophy”: State and mutation are generally considered acceptable as long as one keeps the state local. If no one can observe state changes, it is as if they did not exist.

The idea to speed up graph operations in `GraphsImpl` is to do some preprocessing. Incoming and outgoing edges of a node are kept in two maps—`inEdges` and `outEdges`—initialized when the graph is created. The type of these maps is a mutable `HashMap` from `Node` to `Set[Edge]`, defining a default value for keys that were not entered explicitly; these keys are assumed to map to the empty set of nodes.

The first statements in the body of class `Graph` in code section 3 define and populate the `inEdges` and `outEdges` maps. Class initialization statements in Scala can be written directly in the class body; no separate constructor is necessary. This has the advantage that immutable values can be defined directly using the usual `val x = expr` syntax; no initializing assignments from within a constructor are necessary.

The final interesting definition in code section 3 is the one for `topSort` implemented as an imperative algorithm using a `while` loop that main-

tains at each step the in-degree of all nodes not yet processed. The algorithm starts by initializing the in-degree map and the output buffer sorted. The output list initially contains the `Graphs`’s sources, or the nodes with in-degree zero. When an element is added to the output list, the stored in-degree value of all its successors is reduced.

Whenever the in-degree of a node reaches zero, the node is appended to the output list. This is achieved through a single `while` loop that traverses the (growing) sorted buffer from the left, with variable `frontier` indicating the extent of nodes already processed, while elements are simultaneously added at the right. Intuitively, the `while` loop progresses the same way as the recursive calls in code section 2.

Going parallel. The `GraphsImpl` implementation from code section 3 is efficient on a single processor core. While this may be good enough for small- and mid-size problems, once input data exceeds a certain size we want to use all available processing power and parallelize the algorithm to run on multiple CPU cores.

Parallel programming has a reputation for being difficult and error-prone. Here, we consider a few alternatives of parallelizing the `topSort` algorithm and see how Scala’s high-level abstractions allow us to restructure the algorithm to make good use of the available resources.

Scala provides a set of parallel collection classes. Each of them (such as `Seq`, `Set`, and `Map`) has a parallel counterpart (such as `ParSeq`, `ParSet`, and `ParMap`), respectively. The contract is that operations on a parallel collec-

Code section 4: Parallel topSort using AtomicInteger.

```
import java.util.concurrent.atomic.AtomicInteger
def topSort: Seq[Node] = {
  val indegree = nodes.map(n =>
    (n, new AtomicInteger(inEdges(n).size))).toMap
  def sort(frontier: ParSet[Node]): ParSeq[ParSet[Node]] =
    if (frontier.isEmpty)
      ParSeq()
    else
      frontier +=: sort (
        frontier.flatMap(x =>
          outgoing(x).par.map(succ).filter(y =>
            indegree(y).decrementAndGet == 0))
        sort(sources.par).flatten.seq
      )
}
```

tion class (such as `ParSeq`) may be executed in parallel, and transformer operations (such as `map` and `filter`) will again return a parallel collection.

This way, a chain of operations `myseq.map(f).map(g)` on a `ParSeq` object `myseq` will synchronize after each step, but `f` and `g` on their own may be applied to elements of the collection in parallel. Methods `.par` and `.seq` can be used to convert a sequential into a parallel collection and vice versa.

A first cut at parallelizing `topSort` could start with the imperative code from code section 3 and add coarse-grain locks to protect the two shared data structures—the `indegree` map and the `result` buffer. Unfortunately, however, this approach does not scale, as every access to one of these objects will need to acquire the corresponding lock. The second attempt would be to use locks on a finer-grain level. Rather than protect the `indegree` map with one global lock, we could use a `ConcurrentHashMap` or roll our own by storing `AtomicInteger` objects instead of `Ints` in the map; the corresponding code is shown in code section 4. The `AtomicInteger`s add a layer of indirection, each of which can be changed individually through an `atomicDecrementAnd-`

`Get` operation without interfering with the other nodes.

The second point of contention is the result buffer. In code section 4 we switch back to a more functional style. The `topSort` method now takes the frontier of the currently processed graph as argument and returns a sequence of sets of node. Each set represents a cut through the graph of nodes at the same distance from one of the original sources. The `+` operation is the generalization of the list `cons` operator to arbitrary sequences; it forms a new sequence from a leading element and a trailing sequence. The sequence of sets is flattened (in parallel) at the end of the method.

While this implementation is a nice improvement over coarse-grain locking, it would not exhibit very good performance for many real-world inputs. The problem is that, empirically, most graph problems are scale-free; the degree distribution often follows a power law, meaning the graph has a low diameter (longest distance between two nodes), and most nodes have only a few connections, but a few nodes have an extremely large number of connections. On Twitter, for example, most people have close to zero followers, even though a few

celebrities have tens of millions; for example, in February 2014, Charlie Sheen had 10.7 million followers, Barack Obama 41.4 million, and Justin Bieber 49.6 million.

If we were to run an algorithm over cyclic graphs but implemented it in a style similar to the one in code section 4 on Twitter’s graph, then for every one of Justin Bieber’s almost 50 million followers, we would have to execute compare-and-swap operations on the very `AtomicInteger` used to hold Bieber’s indegree. Individual hubs in the graph thus become bottlenecks and impede scalability.

This means we need to try a different strategy. The key is to restructure the access patterns so no two threads ever write to the same memory location concurrently. If we achieve that, we can remove the synchronization and thus the bottlenecks.

The new code is in code section 5. Like the previous version in code section 4, we use a recursive method but instead of a `Map` that holds `AtomicInteger` objects use a separate `Counters` data structure to hold the indegrees. Instead of directly decrementing the indegree of each target node we encounter, we first group the edges by their successor field, like this:

```
val m = frontier.
flatMap(outgoing).groupBy(succ)
```

The `flatMap` and `groupBy` operations are executed as two data-parallel steps since `frontier` is a `ParIterable`. We can use this more general type here instead of `ParSet` in code section 5 because the `groupBy` operation already ensures unique keys. The result `m` of the operation is a `ParMap` that maps each node in the next frontier to the set of its incoming edges. The algorithm proceeds by performing another parallel iteration over its elements that adjusts the indegree counters.

The key innovation is that all parallel operations iterate over subsets of nodes, and all writes from within parallel loops go to the `indegree` map at the loop index. Since loop indices are guaranteed to be disjoint, there can be no write conflicts if the `Counters` implementation is designed to

Code section 5: Parallel topSort using groupBy.

```
def topSort = {
  val indegree = new Counters(nodes.par)(inEdges(_).size)
  def sort(frontier: ParIterable[Node]): ParSeq[ParIterable[Node]] =
    if (frontier.isEmpty)
      ParSeq()
    else
      frontier +: sort {
        val m = frontier.flatMap(outgoing).groupBy(succ)
        for ((s, es) <- m if indegree.decr(s, es.size) == 0) yield s
      }
    sort(sources.par).flatten.seq
}
```

Performance evaluation.

# Nodes	Listing 2	Listing 3	Listing 4	Listing 5
2,000	27.5056	0.0082	0.0454	0.1006
20,000	—	0.1150	0.1714	0.1686
200,000	—	1.9078	1.3472	1.0096

Running time in seconds for code sections 2–5 on graphs of various sizes. Graphs have 10x as many edges as nodes. The optimized implementations are orders of magnitude faster than the straightforward model. Parallelization adds overhead for small graphs but yields speedup up to 1.9x for large graphs.

allow concurrent writes to disjoint elements. In this case, no fine-grain synchronization is necessary because synchronization happens at the level of bulk operations.

The `Counters` class is a general abstraction implemented like this:


```
class Counters[T](base:
  ParSet[T])(init: T => Int) {
  private val index = base.
  zipWithIndex.toMap
  private val elems = new
  Array[Int](index.size)
  for (x <- base)
  elems(index(x)) = init(x)
  def decr(x: T, delta: Int):
  Int = {
    val idx = index(x)
    elems(idx) -= delta
    elems(idx)
  }
}
```

The constructor takes a parallel set `base` and an initializer `init` and uses `zipWithIndex` to assign a unique integer index to each element in `base`. The following `toMap` call turns the result set of `(T, Int)` pairs into a parallel map `index` with type `ParMap[T, Int]`. The actual counters are stored in an integer array `elems`. The method `decr`, which decrements the counter associated with object `x`, first looks up the index of `x`, then modifies the corresponding slot in `elems`. Counters for different elements can thus be written to concurrently without interference.


The result buffer handling in code section 5 is similar to the previous implementation in code section 4. Nodes are added en bloc, and the result buffer is flattened (in parallel) at the end of method `topSort`.

We have thus explored how Scala's concepts apply to larger program structures and how the language and the standard library support a scalable development style, enabling programmers to start with a high-level, "obviously correct" implementation that can be refined gradually into more sophisticated versions.

A quick performance evaluation is outlined in the table here. For a small graph with just 2,000 nodes and 20,000 edges, the naïve implementation takes 27 seconds on an eight-core Intel X5550 CPU at 2.67 GHz,



The key is to restructure the access patterns so no two threads ever write to the same memory location concurrently.



whereas the fast sequential version (code section 3) runs in under 0.01 seconds. Compared to the optimized sequential version, parallelization actually results in up to 10x slower performance for small graphs but yields speedups of up to 1.9x for a graph with 200,000 nodes and two million edges. All benchmarks were run 10 times; the numbers reported in the table are averages of the last five runs. Input graphs were created using the R-Mat algorithm,⁵ reversing conflicting edges to make the graphs acyclic. The overall achievable performance and parallel speedups depend a lot on the structure of the input data; for example, picking a sparser input graph with two million nodes and only 20,000 edges yields speedups of up to 3.5x compared to the optimized sequential version.

Finally, how can we convince ourselves that the efficient implementations actually conform to the high-level model? Since all versions are executable code with the same interface, it is easy to implement automated test suites using one of the available testing frameworks.

Conclusion

We have offered a high-level introduction to Scala, explaining what makes it appealing to developers, especially its focus on pragmatic choices that unify traditionally disparate programming-language philosophies (such as object-oriented and functional programming). The key lesson is these philosophies need not be contradictory in practice. Regarding functional and object-oriented programming, one fundamental choice is where to define pieces of functionality; for example, we defined `pred` and `succ` on the level of `Graphs` so they are functions from edges to nodes. A more object-oriented approach would be to put them in a bound of the edge type itself; that is, every edge would have parameterless `pred` and `succ` methods. One thing to keep in mind is that this approach would have prevented defining `type Edge = (Node, Node)` because tuples do not have these methods defined. Neither of the variants is necessarily better than the other, and Scala gives programmers the choice. Choice also involves

responsibility, and in many cases novice Scala programmers need guidance to develop an intuitive sense of how to structure programs effectively. Premature abstraction is a common pitfall. Ultimately though, every piece of data is conceptually an object and every operation is a method call. All functionality is thus a member of some object. Research branches of the language¹⁹ go even further, defining control structures (such as conditionals, loops, and pattern matching) as method calls.

The focus on objects and modularity makes Scala a library-centric language; since everything is an object, everything is a library module. Consequently, Scala makes it easy for programmers to define high-level and efficient libraries and frameworks—important for scaling programs from small scripts to large software systems. Its syntactic flexibility, paired with an expressive type system, makes Scala a popular choice for embedding domain-specific languages (DSLs). The main language constructs for component composition are based on traits that can contain other types, including abstract ones, as members.¹³ Scala's traits occupy some middle ground between mixins³ and Schärli's traits.²² As in the latter, they support symmetric composition so mutual dependencies between traits are allowed, but, as with traditional mixins, Scala traits also allow stackable modifications that are resolved through a linearization scheme. Another important abstraction mechanism in Scala is implicit parameters that let one emulate the essential capabilities of Haskell's type classes.¹⁵

Performance scalability is another important dimension. We have seen how we can optimize and parallelize programs using libraries included in the standard Scala distribution. Clients of the graph abstraction did not need to be changed when the internal implementation was replaced with a parallel one. For many real-world applications this level of performance is sufficient. However, we cannot expect to squeeze every last drop of performance out of modern hardware platforms, as with dedicated graph-processing languages (such as GraphLab¹⁰ and Green Marl⁷). With

a bit more effort, though, programmers can achieve even these levels of performance by adding runtime compilation and code generation to their programs. Lightweight modular staging (LMS)²⁰ and Delite^{4,9} are a set of techniques and frameworks that enable embedded DSLs and “active” libraries that generate code from high-level Scala expressions at runtime, even for heterogeneous low-level target languages (such as C, CUDA, and OpenCL). DSLs developed through Delite have been shown to perform competitively with hand-optimized C code. For graph processing, the Opti-Graph DSL²³ (embedded in Scala) performs on par with the standalone language Green Marl. Many Scala features are crucial for LMS and Delite to implement compiler optimizations in a modular and extensible way.²¹

Scala's blend of traditionally disparate philosophies provides benefits greater than the sum of all these parts. ■

References

1. Allen, E.E., Hallett, J.J., Luchangoo, V., Ryu, S., and Steele, G.L., Jr. Modular multiple dispatch with multiple inheritance. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, Y. Cho, R.L. Wainwright, H. Haddad, S.Y. Shin, and Y.W. Koo, Eds. (Seoul, Mar. 11–15). ACM Press, New York, 2007, 1117–1121.
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., and Zaharia, M. A view of cloud computing. *Commun. ACM* 53, 4 (Apr. 2010), 50–58.
3. Bracha, G. and Cook, W.R. Mixin-based inheritance. In *Proceedings of OOPSLA/ECCOP*, A. Yonezawa, Ed. (Ottawa, Oct. 21–25). ACM Press, New York, 1990, 303–311.
4. Brown, K.J., Sujeeth, A.K., Lee, H., Rompf, T., Chafi, H., Odersky, M., and Olukotun, K. A heterogeneous parallel framework for domain-specific languages. In *Proceedings of the 20th International Conference on Parallel Architectures and Compilation Techniques* (Galveston Island, TX, Oct. 10–14). IEEE Computer Society Press, 2011, 89–100.
5. Chakrabarti, D., Zhan, Y., and Faloutsos, C. R-MAT: A recursive model for graph mining. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, M.W. Berry, U. Dayal, C. Kamath, and D.B. Skillicorn, Eds. (Lake Buena Vista, FL, Apr. 22–24). SIAM, 2004, 442–446.
6. Fulgham, B. *The Computer Language Benchmark Game*, 2013; <http://benchmarksgame.alioth.debian.org/>
7. Hong, S., Chafi, H., Sedlar, E., and Olukotun, K. Greenmart: A DSL for easy and efficient graph analysis. In *Proceedings of the 17th International Conference on Architectural Support for Programming Languages and Operating Systems* (London, Mar. 3–7). ACM Press, New York, 2012, 349–362.
8. Hudak, P. *The Haskell School of Expression: Learning Functional Programming Through Multimedia*. Cambridge University Press, Cambridge, U.K., 2000.
9. Lee, H., Brown, K.J., Sujeeth, A.K., Chafi, H., Rompf, T., Odersky, M., and Olukotun, K. Implementing domain-specific languages for heterogeneous parallel computing. *IEEE Micro* 31, 1 (Jan.-Feb. 2011), 42–53.
10. Low, Y., Gonzalez, J., Kyrola, A., Bickson, D., Guestrin, C., and Hellerstein, J.M. Graphlab: A new framework for parallel machine learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, P. Grünwald and P. Spirtes, Eds. (Catalina Island, CA, July 8–11). AUAI Press, 2010, 340–349.

11. Odersky, M. *The Scala Language Specification, Version 2.9*. EPFL, Lausanne, Switzerland, Feb. 2011; <http://www.scala-lang.org/docu/manuals.html>
12. Odersky, M. and Moors, A. Fighting bit rot with types (experience report: Scala collections). In *Proceedings of the Annual Conference on Foundations of Software Technology and Theoretical Computer Science, Vol. 4 of LIPICs Leibniz International Proceedings in Informatics*, R. Kannan and K.N. Kumar, Eds. (Kanpur, India, Dec. 15–17). Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2009, 427–451.
13. Odersky, M. and Zenger, M. Scalable component abstractions. In *Proceedings of the 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, R.E. Johnson and R.P. Gabriel, Eds. (San Diego, Oct. 16–20). ACM Press, New York, 2005, 41–57.
14. Odersky, M., Zenger, M., and Zenger, C. Colored local type inference. In *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, C. Hankin and D. Schmidt, Eds. (London, Jan. 17–19). ACM Press, New York, 2001, 41–53.
15. Oliveira, B.C.d.S., Moors, A., and Odersky, M. Type classes as objects and implicits. In *Proceedings of the 25th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, W.R. Cook, S. Clarke, and M.C. Rinard, Eds. (Reno, NV, Oct. 17–21). ACM Press, New York, 2010, 341–360.
16. Pierce, B.C. and Turner, D.N. Local type inference. In *Proceedings of the 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, D.B. McQueen and L. Cardelli, Eds. (San Diego, Jan. 19–21). ACM Press, New York, 1998, 252–265.
17. Prokopec, A., Bagwell, P., Rompf, T., and Odersky, M. A generic parallel collection framework. In *Proceedings of the 17th International Conference on Parallel Processing, Vol. 6853 of Lecture Notes in Computer Science*, E. Jeannot, R. Namyst, and J. Roman, Eds. (Bordeaux, France, Aug. 29–Sept. 2). Springer, New York, 2011, 136–147.
18. Roestenburg, R. and Bakker, R. *Akka in Action*. Manning Publications Co., Shelter Island, NY, 2013.
19. Rompf, T., Amin, N., Moors, A., Haller, P., and Odersky, M. Scala-virtualized: Linguistic reuse for deep embeddings. *Higher-Order and Symbolic Computation* (Sept. 2013), 1–43.
20. Rompf, T. and Odersky, M. Lightweight modular staging: A pragmatic approach to runtime code generation and compiled DSLs. *Commun. ACM* 55, 6 (June 2012), 121–130.
21. Rompf, T., Sujeeth, A.K., Amin, N., Brown, K., Jovanovic, V., Lee, H., Jonnalagedda, M., Olukotun, K., and Odersky, M. Optimizing data structures in high-level programs. In *Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, R. Giacobazzi and R. Cousot, Eds. (Rome, Italy, Jan. 23–25). ACM Press, New York, 2013, 497–510.
22. Schärli, S., Ducasse, Nierstrasz, O., and Black, A.P. Traits: Composable units of behaviour. In *Proceedings of the 17th European Conference on Object-Oriented Programming*, Vol. 2743 of *Lecture Notes in Computer Science*, L. Cardelli, Ed. (Darmstadt, Germany, July 21–25). Springer, New York, 2003, 248–274.
23. Sujeeth, A.K., Rompf, T., Brown, K.J., Lee, H., Chafi, H., Popic, V., Wu, M., Prokopec, A., Jovanovic, V., Odersky, M., and Olukotun, K. Composition and reuse with compiled domain-specific languages. In *Proceedings of the 27th European Conference on Object-Oriented Programming*, Vol. 7920 of *Lecture Notes in Computer Science*, G. Castagna, Ed. (Montpellier, France, July 1–5). Springer, New York, 2013, 52–78.
24. Twitter. Open source projects; <http://twitter.github.com>
25. Twitter. Scala-school!; <http://twitter.github.com/scala.school>

Martin Odersky (martin.odersky@epfl.ch) is a professor of computer science at EPFL in Lausanne, Switzerland, co-founder of Typesafe, creator of the Scala language, and a fellow of the ACM.

Tiark Rompf (tiark.rompf@epfl.ch) is a researcher at Oracle Labs and EPFL in Lausanne, Switzerland.



Distinguished Speakers Program

talks by and with technology leaders and innovators

Chapters • Colleges and Universities • Corporations • Agencies • Event Planners

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

The Association for Computing Machinery (ACM), the world's largest educational and scientific computing society, now provides colleges and universities, corporations, event and conference planners, and agencies – in addition to ACM local Chapters – with direct access to top technology leaders and innovators from nearly every sector of the computing industry.

Book the speaker for your next event through the ACM Distinguished Speakers Program (DSP) and deliver compelling and insightful content to your audience. **ACM will cover the cost of transportation for the speaker to travel to your event.** Our program features renowned thought leaders in academia, industry and government speaking about the most important topics in the computing and IT world today. Our booking process is simple and convenient. Please visit us at: www.dsp.acm.org. If you have questions, please send them to acmdsp@acm.org.

The ACM Distinguished Speakers Program is an excellent solution for:

Corporations Educate your technical staff, ramp up the knowledge of your team, and give your employees the opportunity to have their questions answered by experts in their field.

Event and Conference Planners Use the ACM DSP to help find compelling speakers for your next conference and reduce your costs in the process.

Colleges and Universities Expand the knowledge base of your students with exciting lectures and the chance to engage with a computing professional in their desired field of expertise.

ACM Local Chapters Boost attendance at your meetings with live talks by DSP speakers and keep your chapter members informed of the latest industry findings.

Captivating Speakers from Exceptional Companies, Colleges and Universities

DSP speakers represent a broad range of companies, colleges and universities, including:

- | | | | |
|------------------|---------------------|----------------------------|--|
| IBM | Sony Pictures | Georgia Tech | University of British Columbia |
| Microsoft | McGill University | Carnegie Mellon University | Siemens Information Systems Bangalore |
| BBN Technologies | Tsinghua University | Stanford University | Lawrence Livermore National Laboratory |
| Raytheon | UCLA | University of Pennsylvania | National Institute of Standards and Technology |

Topics for Every Interest

Over 400 lectures are available from 120 different speakers with topics covering:

- | | | | |
|----------------------------|------------------|-------------------------|---|
| Software | Web Topics | Career-Related Topics | Computer Graphics, Visualization and Interactive Techniques |
| Cloud and Delivery Methods | Computer Systems | Science and Computing | High Performance Computing |
| Emerging Technologies | Open Source | Artificial Intelligence | Human Computer Interaction |
| Engineering | Game Development | Mobile Computing | |

Exceptional Quality Is Our Standard

The same ACM you know from our world-class Digital Library, magazines and journals is now putting the affordable and flexible Distinguished Speaker Program within reach of the computing community.

Microsoft
Research
The DSP is sponsored
in part by Microsoft Europe



**Association for
Computing Machinery**

Advancing Computing as a Science & Profession

AR systems pose potential security concerns that should be addressed before the systems become widespread.

BY FRANZISKA ROESNER, TADAYOSHI KOHNO, AND DAVID MOLNAR

Security and Privacy for Augmented Reality Systems

AUGMENTED REALITY (AR) technologies promise to enhance our perception of and interaction with the real world. Unlike virtual reality systems, which replace the real world with a simulated one, AR systems sense properties of the physical world and overlay computer-generated visual, audio, and haptic signals onto real-world feedback in real time. In this article, we consider the security and privacy concerns associated with AR systems themselves as well as those that arise from the supporting technologies.

Researchers have explored the idea of AR since the 1960s, when Ivan Sutherland described a transparent head-mounted display showing three-dimensional

information.³³ Since the 1990s, AR as a research area has focused on overcoming challenges with display technology, tracking, and registration to properly align virtual and real objects, user interfaces and human factors, auxiliary sensing devices, and the design of novel AR applications.^{1,2,6,22,36,41}

However, it is only recently that early-generation AR technologies have begun shipping commercially. For example, Google recently released a limited number of its Google Glass heads-up glasses for AR applications. Many other early-generation AR applications are enabled by the ubiquity of smartphones and other mobile devices. Examples include the Word Lens iPhone application—an application that overlays translated text on the camera's view of foreign text—and Layar, a geolocation-based AR platform that allows developers to create AR layers for the world (for example, for game playing); see Figure 1. The recent advent of 1GHz processors, location sensors, and high resolution, autofocusing cameras in mobile phones has made these applications possible.

In this article, we take a broad view of the AR space, considering both direct applications of AR as well the technologies necessary to support these applications. Beyond the mobile phone, devices are becoming available that enhance sensing, display, and data sharing, which will enable more com-

» key insights

- **Augmented reality technologies, which are rapidly evolving and becoming commercially available, will create new challenges and opportunities for security and privacy. These challenges can be characterized along two axes: system scope and functionality.**
- **Security and privacy challenges with AR technologies include conflicts among applications sharing input and output devices, as well as more complex access control for sensor data. While some issues can be addressed by adapting existing solutions for smartphones, others will require novel approaches.**
- **AR technologies provide an opportunity to address existing security and privacy challenges in new ways.**



Keychain
Password:
ke1F367c22



Figure 1. Phone-based augmented reality. On the left, a picture of Word Lens, an iPhone application that provides seamless “in-picture” translation (source: <http://www.flickr.com/photos/neven/5269418871/>). Here the app translates the word “craft” from English to Spanish and then back again. On the right, a picture of Layar, an “augmented reality browser” shipping on Android phones (source: <http://site.layar.com/company/blog/make-your-ownlayar-screen-shot-with-the-dreamcatcher/>).



Figure 2. Wearable input and output. On the left, a Looxcie body-worn camera worn by a ranger in Kenya (source: <http://looxcie.com/index.php/image-gallery>). On the right, a Google Glass prototype in June 2012 (source: <http://www.flickr.com/photos/azugal-dia/7457645618>).



plex AR systems. For example, Looxcie—an over-the-ear, always-on video camera—includes a feature enabling wearers to share their live video feed with anyone else in the world. Microsoft’s SDK for Kinect,²⁰ which provides accurate motion sensing by combining an RGB camera, a depth camera, and a multi-array microphone, has enabled numerous prototype AR applications. In addition to Google Glass, transparent, wearable displays are now available for research purposes from several companies, such as Vuzix, Lumus, and Meta SpaceGlasses. Figure 2 shows examples of such input and output devices. (Table 1 offers a summary of AR-enabling technologies; many of these technologies are shipping today, while others are still experimental.)

These technologies will enable commercial AR applications and are at the cusp of significant innovation, which will bring significant benefits to many users. However, these technologies

may also bring unforeseen computer security and privacy risks. Previous research in the AR space has rarely considered these issues. Rather than wait for these technologies to fully mature and then retroactively try to develop security and privacy safeguards, we argue that now is the time to consider security and privacy issues, while the technologies are still young and malleable. To guide this process, we ask the following questions: What new security and privacy research challenges arise with AR systems and the technologies that support them? What novel opportunities do AR technologies create for improving security and privacy?

We find that AR technologies form an important, new, and fertile playground for computer security and privacy research and industry. Of course, these technologies should leverage standard security best practices, such as on-device and network encryption. Nevertheless, we find unique obsta-

cles—such as handling conflicts between multiple applications sharing an AR system’s output—that are simultaneously intellectually challenging yet surmountable. Other challenges, such as access control for data, are well known in other arenas but become even more important for AR technologies with their always-on, always-sensing inputs. Given the future importance of AR technologies, researchers already tackling these issues in other domains can find value in refocusing their attention on AR applications.

In addition to presenting new challenges, AR systems present opportunities for new applications that improve security and privacy. For example, these technologies can provide personal digital views of content on personal displays. Imagine a password manager that superimposes visual indicators over the correct keys for a complex password when a user looks at a keyboard, or an application that alerts the user when someone is lying.

In this article, we explore new security and privacy challenges presented by AR technologies, defensive directions, and new applications of AR systems to known security and privacy issues.

Challenges

The AR applications and technologies we consider may have any or all of the following characteristics, in addition to the traditional definition of aligning real and virtual objects in real time:

- ▶ A complex set of input devices and sensors that are always on (for example, camera, GPS, microphone).
- ▶ Multiple output devices (for example, display, earpiece).
- ▶ A platform that can run multiple applications simultaneously.
- ▶ The ability to communicate wirelessly with other AR systems.

Here, we present a set of security and privacy challenges that come with these novel technologies and their applications, as summarized in Table 2. We organize these challenges along two axes: *system scope* and *functionality*. On one axis, we consider AR systems of increasing scope: single applications, multiple applications within a single AR platform, and multiple communicating AR systems. The challenges in each category first ap-

pear at that level of system complexity. For each scope, we further categorize challenges as related to input, output, or data access. We encourage future designers of AR technologies to consider security and privacy challenges along both axes.

Readers familiar with smartphone security may observe some overlap between those challenges and the set we present here. We note that some techniques from smartphone security may be applicable to AR technologies; others will need to be rethought in this new context.

Challenges with single applications. We first consider threats and challenges limited in scope to a single AR application.

Output. Users must place significant trust in AR applications that overlay real-world visual, auditory, or haptic perceptions with virtual feedback. Devices providing immersive feedback can be used by malicious applications to deceive users about the real world. For example, a future malicious application might overlay an incorrect speed limit on top of a real speed limit sign (or place a fake sign where there is none), or intentionally provide an incorrect translation for real-world text in a foreign language. More generally, such an application can trick users into falsely believing that certain objects are or are not present in the real world.

Malicious applications can use similar techniques to cause sensory overload for users. By flashing bright lights in the display, playing loud sounds, or delivering intense haptic feedback, applications could physically harm users. Such attacks are not unprecedented: attackers have targeted epilepsy forums, posting messages containing flashing animated gifs to trigger headaches or seizures.²⁴ Emerging AR platforms must consider and prevent these types of attacks.

These output attacks are more serious in immersive AR applications than they are in today’s desktop or handheld computing scenarios both because it is more difficult for users to distinguish virtual from real feedback and because it may be more difficult for users to remove or shut down the system. As a last resort for output attacks, users must be able to easily and reliably return to the real world, that is, with all output de-

vices verifiably turned off.

In the near term, removing the system is a simple way to achieve this return to reality. However, future wearable systems may be difficult or impossible for users to remove (for example, contact lenses²³ or implanted devices), and today’s non-wearable systems may already be difficult for users to evade. For example, several automotive manufacturers have produced windshields that display augmented content over the user’s view of the road.⁵ In these cases, the system should have a trusted path for the user to return to reality, analogous to Ctrl-Alt-Del on Windows computers. Determining the best such sequence, or the right input mode (for example, gestures or speech), requires research for each AR system. Another approach may be to reserve a trusted region of the display that always shows the real world.

Input. AR applications will undoubtedly face similar input validation and sanitization challenges as conventional applications. For example, a translation application that parses text in the real world may be exploited by maliciously crafted text on a sign. Traditional input validation techniques are likely to apply, but the designers of AR

systems should be aware of their necessity in this new context.

Data access. To provide their intended functionality, AR applications may require access to a variety of sensor data, including video and audio feeds, GPS data, temperature, accelerometer readings, and more. As in desktop and smartphone operating systems, an important challenge for AR systems will be to balance the access required for functionality with the risk of an application stealing data or misusing that access. For example, a malicious application may leak the user’s location or video feed to its backend servers. The existing proof-of-concept PlaceRaider attack³⁴ shows that smartphone sensors can be used to gather enough information to create three-dimensional models of indoor environments.

Unlike most of today’s desktop and smartphone applications, complex AR applications will require rich, always-on sensing. For example, an application that automatically detects and scans QR codes requires constant access to video stream data, as does an application that automatically detects when the user is entering a password on another device and provides pass-

Table 1. Summary of commercial and emerging AR technologies.

	Commercially Available Today	Experimentally Only
<i>Sensors (Inputs)</i>	Body-worn RGB cameras GPS (error of 5 meters or more) Accurate motion sensing (for example, Kinect)	Haptic sensors ²⁸
<i>Feedback (Outputs)</i>	Opaque near-eye display Phone display/speaker Invisible Bluetooth earpiece	Transparent near-eye display Embedded displays (contact lenses ²³) Haptic feedback ¹⁷
<i>Services</i>	Simple cloud services (photo gallery) Marker-based tracking ³⁹ Good face detection (not recognition) ³⁷ Expensive or cheap but inaccurate transcription	Complex cloud services (object recognition) Markerless tracking Good face recognition Cheap accurate transcription
<i>Sharing</i>	Selective sharing (photos, videos, location)	Automatic sharing

Table 2. Security and privacy challenges for AR technologies. We categorize these challenges by two axes: challenges related to output, input, and data access, as arise in single applications, multi-application systems, and multiple interacting systems.

	Single Application	Multiple Applications	Multiple Systems
<i>Output</i>	Deception attacks Overload attacks Trusted path to reality	Handling conflicts Clickjacking	Conflicting views
<i>Input</i>	Input validation	Resolving focus	Aggregate input
<i>Data Access</i>	Access control for sensor data Bystander privacy	Cross-app sharing	Cross-system sharing

word assistance (as we will discuss). As a result, these privacy risks are much greater than in conventional systems.

AR systems should take approaches that limit these risks. For example, individual applications will likely not need access to all sensor data. Perhaps an application only requires access to a portion of the screen when the user is in a certain location, or only needs to know about certain objects the system recognizes (for example, via the Kinect's skeleton recognizer), rather than needing access to the entire raw camera feed. AR system designers must consider the appropriate granularity for these permissions, and the design of usable permission management interfaces will be important. Existing manifest or prompt-based solutions as used in smartphones are unlikely to scale in a usable way, and the long-term (rather than one-time) data access needs of AR applications make the application of in-context access control solutions like user-driven access control²⁸ not straightforward.

Always-on cameras and other sensors will also create a privacy risk for bystanders, which Krevelen and Poelman identify as a challenge for widespread social acceptance of AR.³⁶ Bystanders should be able to opt out of or be anonymized (for example, blurred) in the recordings of others; prior work has examined such issues.^{9,31} AR users may need methods to prove to skeptical bystanders that such safeguards are in place. Legislation or market forces may lead to cameras that respond to

requests from other devices or the environment; news reports suggest that Apple has considered adding such a capability to the iPhone to prevent videotaping of live events, such as concerts.⁴ Cameras may also alert bystanders while recording, such as by flashing a light³⁶ or by providing access to more complex policy information.¹⁹

The CVDazzle project¹⁰ pursues a different approach—using makeup to confuse face detection algorithms—that provides privacy without compliant cameras. The key limitation is that CVDazzle is painstakingly hand-tuned for one particular face detection algorithm. A research question is to find a general algorithm for synthesizing makeup that fools face detection.

Challenges with multiple applications. Though AR applications are often conceived and prototyped in isolation, we can expect future AR platforms, like those built on Google Glass or the Microsoft Kinect, will support multiple applications running simultaneously, sharing input and output devices, and exposing data and APIs to each other (see Figure 3). Researchers must anticipate these developments and ensure an “operating system for augmented reality” is designed with appropriate considerations for security and privacy.

Output. In a multi-application AR system, applications will share output devices, including displays, audio output, and haptic feedback. Conflicts among multiple applications attempting to use these output devices can

lead to security concerns. For example, a malicious application might try to obscure content presented by another application (for example, visually or aurally covering up a correct translation with an incorrect one).

Nevertheless, output sharing will be necessary to provide desirable functionality in AR systems. For example, a user may wish to simultaneously view content overlaid on their view of reality from multiple applications, such as directions supplied by a maps application, a social feed summarizing the activity of nearby friends, the track currently playing in a music application, and so on. Thus, the naive solution, in which only one application controls the display at a time (as in Android today, for instance), is insufficient.

Thus, future AR systems must handle conflicts between multiple applications attempting to produce output. For example, five applications may all want to annotate the same object (for example, with a translation subtitle), and the system will need to prioritize them. Furthermore, it may be important for users to know which content was generated by which application—for instance, whether an annotated product recommendation comes from a friend or an advertiser. AR system designers must create interfaces that make the origins of displayed content clear to or easily discoverable by users.

Traditional attacks based on the manipulation of output may require new approaches or new formulations in the AR context. For example, in today's systems, applications can mount clickjacking attacks that trick users into clicking on sensitive user interface elements from another application (for example, to post something on the user's social media profile). These attacks generally work either by manipulating the display of the sensitive element—by making it transparent or partially obscuring it in a clever way—or by suddenly displaying sensitive elements just before users click in a predictable place. Future applications on AR systems may develop new techniques for tricking users into interacting with elements, and system designers must anticipate these threats. For example, an AR application could attempt to trick a user into interacting with an object in the physi-

Figure 3. Multi-application AR. Emerging and future AR platforms will support multiple applications running simultaneously, sharing input and output devices, and exposing data and APIs to each other. In a multi-application AR system, applications like those depicted in this mockup will share output devices, including displays, audio output, and haptic feedback. Conflicts among these applications can result in security concerns.




cal, rather than the virtual, world.


Input. Users will likely not interact with AR systems using traditional input methods like clicking on a mouse or even using a touchscreen. Instead, users may increasingly interact with these systems using subtle input to haptic sensors (for example, embedded in gloves), using voice, or with the aid of gaze-tracking technologies. With these input techniques and multiple running applications, it will be non-trivial for the system to resolve which application is in focus and should thus receive input.

For example, today's voice interactions happen either following an explicit user action indicating the destination application (for example, clicking on the "Siri" button on an iPhone) or on systems in which only one application can ever receive voice input (for example, on the Xbox). When multiple applications are active and might receive voice or other input at any given time, there must be either a usable way for users to bring applications into focus, or for the system to determine the correct intended destination for input commands when focus is ambiguous. We emphasize that future AR systems are likely to run multiple applications simultaneously, many of them running and listening for input without having any visible output. Improperly designed focus resolution may make it easy for malicious applications to steal user input intended for another application (for example, to steal a password intended for the login box of another application). For example, a malicious application may attempt to register a similar-sounding verbal keyword as another, sensitive application, intentionally increasing input ambiguity.

Data access. As in traditional operating systems, AR applications will likely wish to expose APIs to each other, and users may wish to share virtual objects between applications. Researchers must explore appropriate access control models for cross-application sharing. Certainly lessons from traditional access control design can be applied in this space, but new technologies and environments may require new approaches. For example, copy-and-paste and drag-and-drop are established user gestures for sharing data between traditional applications



We argue that now is the time to consider AR security and privacy issues, while the technologies are still young and malleable.



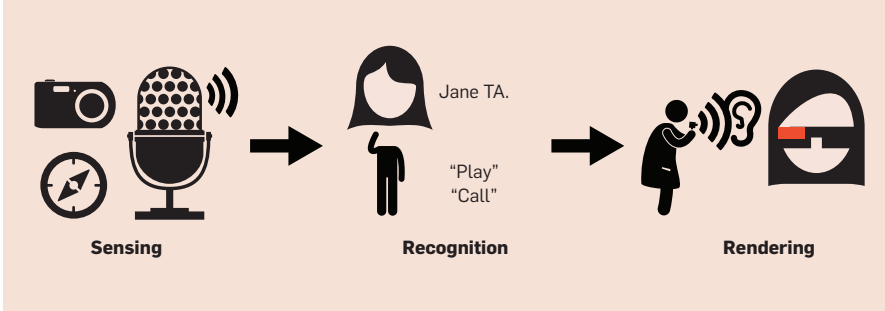
and thus have access control implications. A long line of work in desktop and smartphone systems has attempted to map user actions to application privileges (examples include Miller²¹ and Roesner et al.²⁸); AR systems will need to evolve new user gestures to indicate sharing intent. Additionally, AR systems are unlikely to display applications in labeled windows the way that traditional desktop operating systems do, so new interaction paradigms are needed to enable users to identify applications and indicate which application should receive shared data.

Challenges with multiple systems. Moving beyond a single AR system running multiple applications, we consider the interactions between multiple AR systems belonging to different users. Prior work in AR proposes collaborative applications among multiple users of an AR system. These applications include multiplayer games,^{11,32,40} telepresence for remote conferencing,¹⁶ and face-to-face collaboration.²⁶ These types of applications pose additional security and privacy challenges.

Output. Multiple users may have differing views of the world presented by their respective AR systems. For example, different users may see different virtual advertisements superimposed on real-world billboards, or different users watching a presentation may be shown different content based on their access levels (such as, one user may see top-secret footnotes while others do not). Such conflicting views will require users to manage mental models of who can perceive which information, lest they accidentally reveal private information intended only for themselves. Addressing this concern will require innovations in interface design for aiding users in this task.

Input. A rise in the complexity of AR systems and applications will be tightly coupled with a rise in the number and complexity of sensor inputs provided by enabling technologies. This abundance of sensor input from many users will in turn lead to novel collaborative sensing applications, which can themselves feed data back into AR applications. For example, Google already uses data collected by users' smartphones to estimate traffic conditions, which is then reported back to user's phones.⁸ This type of data is necessary to enable

Figure 4. AR pipeline. AR applications (1) gather sensory data, from which they (2) extract objects with high-level semantics. Finally, they (3) render on top of the user's senses.



future AR applications displayed on the car's windshield, for example.

However, this type of aggregate input can be used by malicious users to fool the data collection systems. For example, a review site might leverage location tracking to measure a restaurant's popularity by noting the average number of people present during the day. A canny restaurateur may then pay people to stand in the restaurant without buying anything. The restaurant's measured popularity rises but has no relationship to its quality.

AR technologies that constantly collect data will drive the adoption of such collaborative sensing applications; thus, these security concerns will increase in importance. As another example, the Community Seismic Network aggregates accelerometer sensor data of many individuals to detect and predict earthquakes; an attacker could manipulate the sensors to "spoof" unusual seismic activity, for example, by encouraging many individuals monitored by the project to jump at once in the context of an unrelated game. (For example, *Improv Everything*¹³ asks users to play provided audio files as a designated time and follow the audio instructions.) Trusted sensors³⁰—while important to prevent other attacks—do not help in these cases, as real-world conditions are manipulated.

Data access. In addition to showing different content to different users, communicating AR systems will allow users to share virtual content with each other. For example, one user may create a virtual document within their private AR system and later choose to share its display with the systems of other users. Some sharing may even be implicit; imagine an AR system that automatically uses the camera feeds of nearby users to pro-

vide a given user with a real-time 3D model of him or herself.

The implicit or explicit sharing of data across separate AR systems can enable valuable applications. However, appropriate access control models and interfaces are needed to allow users to manage this sharing. Today, users already have difficulty forming mental models of their privacy settings on services like Facebook because of the complexity of relationships between people and data items.¹⁸ The vast amount of data collected by AR systems and the integration of virtual objects with the real world will make this problem only more difficult.

Defensive Directions

Here, we outline several defensive directions for AR technologies. First, some of the security and privacy challenges associated with AR technologies are similar to those faced by smartphones today, such as the privacy of sensor data and cross-application sharing. In some cases, an appropriate defensive direction for AR is to adapt smartphone solutions. For example, permission manifests and the app store review process may be adopted in the short term.

In the long term, however, there are several reasons that approaches in the AR context must differ from smartphone solutions. First, an analysis of the resource needs of smartphone applications²⁸ showed that most require only one-time or short-term access to most resources, making solutions that require in-context user interactions (such as user-driven access control²⁸) feasible. By contrast, AR applications will require long-term or permanent access to sensor data at a scale beyond smartphone applications. Further, AR resource access will not be as clear

to users and to bystanders as in the smartphone context—for example, an AR system's camera will always be on, whereas a smartphone's camera, even if turned on by malware, provides much less data while the phone is in the user's pocket. Thus, we argue it is important to consider full-fledged future AR contexts when designing solutions in this space.

Along these lines, new research into AR-specific solutions will be needed. For example, researchers have begun considering operating system support specific to AR.⁷ AR applications—and the underlying OS—naturally follow the pipeline shown in Figure 4, so research can be characterized accordingly, and different research models can assume different boundaries between the application and the OS. In the first stage, sensing, an application (or the OS) gathers raw sensory data such as audio, video, or radio waves; research here includes limiting which sensed information is collected (for example, polite cameras^{9,31}) or limiting its use (for example, retention policies). Second, in the recognition stage, machine-learning algorithms extract objects with high-level semantics: as an example, the figure shows a Kinect skeleton, a face, the associated name, and voice command triggers. Related research includes changing objects to cause false negatives (for example, *CVDazzle*¹⁰) and policies governing application access to objects.¹⁵ Finally, the application (or the OS) renders on top of the user's senses, such as vision and hearing. Research here includes uncovering invariants that must be respected to avoid harming the user and building a performant "trusted renderer" that respects these invariants.

Not all AR defensive directions will consist of technical solutions. Some challenges may call for social, policy, or legal approaches; for example, potential policies for bystander opt-outs and compliant cameras, as discussed earlier. Other issues will similarly benefit from non-technical approaches.

Finally, we call for an AR testbed for researchers working in this space. Most experimental AR applications today rely on the Microsoft Kinect or smartphone platforms like Layar; both involve only single applications running at one time, thereby hiding

challenges that arise as AR systems increase in complexity.

Novel Applications

Though AR technologies create important security and privacy concerns, there is also an unexplored opportunity for them to enhance security and privacy through their application to existing problems. Here, we consider opportunities for new security and privacy enhancing applications enabled by AR technologies and systems. Our list is undoubtedly incomplete; we hope to see rich future work in this area.

Leveraging personal views. AR systems that integrate heads-up or other personal displays can leverage these personal views to address existing security and privacy concerns—in particular, protecting private data and improving password management.

Personal displays present a strong defense against shoulder surfing, as users may interact with applications visible in their own view only. For example, someone using a laptop on an airplane today exposes everything they view and type to their seat neighbors, and researchers have demonstrated that footage from low-cost cameras can be used to reconstruct a user's typing on a virtual mobile keyboard.²⁵ A personal heads-up display combined with a haptic sensor for discreet input would allow for greatly improved privacy.^a

Personal displays further enable encrypted content in the real world that can only be decrypted by the AR systems of the intended recipients. For example, a company can post encrypted notices on a bulletin board that employees can read through their company-issued AR systems, but that visitors to the company's building cannot. (Storing only the key, not the encrypted content, on a server accessible by the AR system requires adversaries to find the physical notice rather than simply compromise the company's

servers.) Precursors of such a system are possible today using smartphones and 2D barcodes that encode URLs to data with appropriate access control; augmented heads-up displays will prevent the need for a manual scan.

AR systems can also act as an enhanced password manager for users, presenting passwords or password hints via the personal display. For example, a display could outline the appropriate characters the user must enter on legacy devices like ATM PIN pads. Users could then be assigned strong passwords, as they would never need to actually remember them. This application requires markerless tracking and a system design that properly protects the stored passwords.

As a concrete example, we have implemented a prototype password manager application consisting of a Google Glass application and a browser (Chrome) extension (see Figure 5). The Chrome extension modifies the browser's user interface to display a QR code representing the website currently displayed to the user (the website in the browser's address bar). Users can ask the Google Glass application to scan these QR codes and consult its password database by us-

ing the voice command “OK Glass, find password.” If the user has previously stored a password for that website, the application displays the password; otherwise, the user can enroll a new password by asking the Chrome extension to generate an enrollment QR code and asking the Glass to store the new password using the “enroll password” voice command. We have made the code for our prototype available at <https://github.com/froeschele/GlassPass>.

By designing the QR code displayed by the browser extension to include a secret shared between the browser and the phone, this application could furthermore serve as phishing protection, as websites would not be able to create and display forged QR codes that would map to legitimate passwords in the password manager.

Leveraging complex sensor systems. AR systems benefit from the combination of multiple input and sensing devices, which can be combined to enhance digital and physical security, privacy, and safety.

Future systems can leverage AR technologies to *detect privacy or security conditions* of which the user should be alerted. For example, rather than rely-

Figure 5. Prototype AR Password Manager. Our Chrome extension (background) displays a QR code representing the current website. In response to the voice command “find password,” our Google Glass application (foreground) scans this QR code and displays the stored password for that site privately in the heads-up display.



^a We observe that see-through displays, such as that used by Google Glass, may not be fully private from external observers. For example, images taken of the display using a telephoto lens may be used to reconstruct the content of the screen, similar to reconstructing content from screen reflections.³ Future research should fully characterize this threat and design appropriate defenses.

ing on compliant cameras to shield users from unwanted recording, a system could alert users when it detects camera lenses pointed at them, using (for instance) computer vision to detect the glint of light reflecting off a lens.³⁵ It could also detect some forms of eavesdropping, for example, a laser microphone pointed at a window.

Such systems could also *detect physical deception attempts*. For example, an AR system could estimate the size and shape of an ATM card slot, then issue a warning if it appears a card-skimming device has been added. Similarly, existing work on computerized interpretation of facial expressions¹² could be applied to behavior-based lie detection.³⁸ One of our colleagues refers to this application as “spidey sense.”

Beyond storing passwords, AR systems can be used for implicit authentication of their users. The plethora of sensors attached to people using these technologies can be used to authenticate them with biometric and behavioral characteristics. Prior work has examined the possibility of such mechanisms on mobile phones^{14,27} AR systems would provide far more powerful authentication. Similarly, sensor data could be used to help with authorization and access control decisions.

Beyond the sensors attached to an individual (for example, Alice), the sensors of bystanders could also be used to authenticate her by providing the authentication system with third-party visual, audio, and other sensory views of Alice. This third-party authentication system would distribute trust to systems and persons with no incentive to falsely authenticate Alice.

Conclusion

AR systems, with their sophisticated and pervasive input, output, and processing capabilities, have the potential to significantly benefit many users. To complement ongoing innovations in AR technologies, we argue now is also the time to define a roadmap for protecting the computer security and privacy of AR systems—before these systems become widely deployed and their architectures become entrenched. To catalyze this roadmap, we consider new security and privacy challenges posed by these systems, and we explore opportunities afforded by these tech-

nologies to create novel privacy- and security-enhancing applications.

Acknowledgments

This work is supported in part by the National Science Foundation (Grants CNS-0846065, CNS-0905384, and a Graduate Research Fellowship under Grant DGE-0718124) and by a Microsoft Research Fellowship. We thank Luis Ceze, Lydia Chilton, Alexei Czeskis, Nicki Dell, Tamara Denning, Karl Koscher, Brandon Lucia, Alex Moshchuk, Bryan Parno, Karin Strauss Helen Wang, and anonymous reviewers. **C**

References

- Azuma, R.T. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments* 6 (1997), 355–385.
- Azuma, R., Baillet, Y., Behringer, R., Feiner, S., Julier, S. and Macintyre, B. Recent advances in augmented reality. *IEEE Computer Graphics and Applications* 21, 6 (2001), 34–47.
- Backes, M., Chen, T., Duermuth, M., Lensch, H. and Welk, M. Tempest in a teapot: Compromising reflections revisited. *IEEE Symposium on Security and Privacy* (2009).
- Business Insider. This apple patent will shut down your camera at live concerts; <http://www.businessinsider.com/iphone-concert-patent-2011-6>.
- CNN. Augmented-reality windshields and the future of driving, 2012; <http://virtual.vtt.fi/virtual/proj2/multimedia/alvar.html>.
- Costanza, E., Kunz, A. and Fjeld, M. Mixed reality: A survey. *Human Machine Interaction*. Springer-Verlag, 2009, 47–68.
- D'Antoni, L., Dunn, A., Jana, S., et al. Operating system support for augmented reality applications. In *Proceedings of USENIX Workshop on Hot Topics in Operating Systems* (2013).
- Google. Crowdsourcing road congestion data; <http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html>.
- Halderman, J.A., Waters, B. and Felten, E.W. Privacy management for portable recording devices. In *Proceedings of the 3rd ACM Workshop on Privacy in Electronic Society* (2004).
- Harvey, A. CVDazzle: Camouflage from Computer Vision; <http://cvdazzle.com/>.
- Henrysson, A., Billinghurst, M., and Ollila, M. Face to face collaborative AR on mobile phones. In *Proceeding of the 4th IEEE/ACM International Symposium on Mixed & Augmented Reality* (2005).
- Hoque, M.E., McDuff, D. and Picard, R.W. Exploring temporal patterns in classifying frustrated and delighted smiles. *IEEE Transactions on Affective Computing* 3 (2012), 323–334.
- Improv Everywhere. The Mp3 Experiments, 2012; <http://improveverywhere.com/missions/the-mp3-experiments/>.
- Jakobsson, M., Shi, E., Golle, P., and Chow, R. Implicit authentication for mobile devices. In *Proceedings of the 4th USENIX Workshop on Hot Topics in Security* (2009), USENIX.
- Jana, S., Molnar, D., Moshchuk, A. et al. Enabling fine-grained permissions for augmented reality applications with recognizers. Tech. Rep. MSR-TR-2013-11, Microsoft Research, Feb. 2013.
- Kato, H. and Billinghurst, M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *IEEE/ACM Workshop on Augmented Reality* (1999).
- Laycock, S. and Day, A. A survey of haptic rendering techniques. *Comp. Graphics Forum*. 26, 1 (2007), 50–65.
- Madejski, M., Johnson, M. and Bellovin, S.M. The Failure of Online Social Network Privacy Settings. Tech. Rep. CUCS-010-11, Dept. of Comp. Science, Columbia University, 2011.
- Maganis, G., Jung, J., Kohno, T. et al. Sensor Tricorder: What does that sensor know about me? In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications* (2011), ACM.

- Microsoft. Kinect for Windows, 2012; <http://www.microsoft.com/en-us/kinectforwindows/>.
- Miller, M.S. Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control. Ph.D. thesis. Johns Hopkins University, Baltimore, MD, 2006.
- Papagiannakis, G., Singh, G. and Magnenatthalmann, N. A survey of mobile and wireless technologies for augmented reality systems. *Computer Animation and Virtual Worlds* 19 (2008), 3–22.
- Parviz, B. For your eye only. *IEEE Spectrum* 46 (2009), 36–41.
- Poulsen, K. Hackers assault epilepsy patients via computer. *Wired* (2008); <http://www.wired.com/politics/security/news/2008/03/epilepsy>.
- Raguram, R., White, A.M., Goswami, D. et al. iSpy: automatic reconstruction of typed input from compromising reflections. In *Proceedings of the 18th ACM Conf. Computer and Communications Security*.
- Reitmayr, G. and Schmalstieg, D. Mobile collaborative augmented reality. In *Proceedings of the 4th International Symp. on Augmented Reality* (2001).
- Riva, O., Qin, C., Strauss, K., and Lymberopoulos, D. Progressive authentication: Deciding when to authenticate on mobile phones. In *Proceedings of the 21st USENIX Security Symposium* (2012).
- Roesner, F., Kohno, T., Moshchuk, A. et al. User-driven access control: Rethinking permission granting in modern operating systems. *IEEE Symposium on Security and Privacy* (2012).
- Saponas, T.S., Tan, D.S., Morris, D. et al. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd ACM Symposium on User Interface Software and Technology* (2009).
- Saroui, S. and Wolman, A. I am a sensor, and I approve this message. In *Proceedings of the 11th Workshop on Mobile Computing Systems and Applications* (2010), ACM.
- Schiff, J., Meingast, M., Mulligan, D.K., Sastry, S. and Goldberg, K.Y. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In *Proceeding of the 2007 Int'l Conference on Intelligent Robots and Systems*.
- Starner, T., Leibe, B., Singletary, B. and Pair, J. Mindwarping: Towards creating a compelling collaborative augmented reality game. In *ACM Intelligent User Interfaces* (2000).
- Sutherland, I.E. A head-mounted three-dimensional display. In *Proceedings of the Fall Joint Computer Conference, American Federation of Information Processing Societies* (1968).
- Templeman, R., Rahman, Z., Crandall, D.J., and Kapadia, A. Placeraider: Virtual theft in physical spaces with smartphones. CoRR abs/1209.5982 (2012).
- Truong, K., Patel, S., Summet, J. and Abowd, G. Preventing camera recording by designing a capture-resistant environment. *Proceedings of Ubicomp* (2005).
- Van Krevelen, D. and Poelman, R. A survey of augmented reality technologies, applications, and limitations. *The International Journal of Virtual Reality* 9 (2010), 1–20.
- Viola, P. and Jones, M. Robust real-time object detection. *International Journal of Computer Vision* 57 (2004), 137–154, Hingham, MA.
- Vrij, A., Edward, K., Roberts, K. and Bull, R. Detecting deceit via analysis of verbal and nonverbal behavior. *Journal of Nonverbal Behavior* 24 (2000), 239–263.
- VTT Technical Research Centre Of Finland. Alvar Software Library, 2009. <http://cnn.com/2012/01/13/tech/innovation/ces-future-driving/>.
- Wagner, D., Pinteric, T., Ledermann, F. and Schmalstieg, D. Towards massively multi-user augmented reality on handheld devices. In *Proceedings of the 3rd International Conference on Pervasive Computing* (2005).
- Zhou, F., Duh, H. B.-L. and Billinghurst, M. Trends in augmented reality tracking, interaction and display: A review of 10 years of ISMAR. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (2008).

Franziska Roesner (franzi@cs.washington.edu) is a Ph.D. candidate at the University of Washington, Seattle.

Tadayoshi Kohno (yoshi@cs.washington.edu) is an associate professor at the University of Washington, Seattle.

David Molnar (dmolnar@microsoft.com) is a researcher at Microsoft Research, Redmond, WA.

Copyright held by Owner(s)/Author(s). Publication rights licensed to ACM. \$15.00.

research highlights

P. 98

Technical Perspective A ‘Reasonable’ Solution to Deformation Methods

By Joe Warren

P. 99

Bounded Biharmonic Weights for Real-Time Deformation

By Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine-Hornung

Technical Perspective

A ‘Reasonable’ Solution to Deformation Methods

By Joe Warren

GEOMETRY PLAYS A key role in the world of modern computing. In science and engineering, mathematical models of geometry are crucial for applications like simulation and manufacturing. In the arts and entertainment, mathematical models of geometry are ubiquitous in applications like games and movies, and are useful even for image editing. Developing new variants of these models that are both intuitive and computationally efficient remains an area of intense research in computer graphics.

One classical problem in this area is that of deformable modeling; that is, deforming a given shape into a target shape via some type of mathematical method (ideally an interactive one). Solutions to this problem lie at the core of most current computer animation systems. One standard approach for these methods is to view the shape as being comprised of a deformable material and having a collection of embedded shape handles. Typically, these shape handles are either points or piecewise linear shapes connecting these points that a user may manipulate interactively. In this framework, the deformation method computes a “reasonable” deformation of the shape based on some simple physical model.


One interesting question here is what corresponds to a “reasonable” deformation. Historically, the simplest mathematical models for deformations have been piecewise polynomials. In the one-dimensional case, these models (such as B-splines) are simple, intuitive, and expressive. For two-dimensional shapes, piecewise linear deformations are again simple and intuitive. However, higher-order piecewise polynomial models lack the smoothness and flexibility needed in many applications. Since univariate polynomials are the solutions to simple differential equations, many modern deformation schemes model smooth deformations as the solutions to a par-

tial differential equation (PDE). Most deformation methods based on PDEs typically focus on either harmonic functions (solutions to Laplace’s equation) or, more recently, biharmonic functions (solutions to the iterated Laplace’s equation).

In this vein, Jacobson et al. construct a deformation method that allows a wide range of handle types (points, line segments, open and closed polygons) and produces deformations that are biharmonic functions. In this framework, the positions of the handles are treated as boundary conditions for the associated PDE. An important fact to note is that the use of biharmonic functions in place of the harmonic functions is due to the inclusion of isolated handles. Harmonic functions are typically used to model the idealized deformation of thin elastic membranes while biharmonic functions are typically used to model deformations of thin elastic plates. Interpolating an isolated handle with a thin membrane (harmonic) will lead to non-smooth deformations that often fold back on themselves. Interpolating an isolated handle with a thin plate (biharmonic) normally yields a smooth deformation with no fold back.

Jacobson et al.’s main technical innovation is the addition of linear inequality constraints to their model to ensure the resulting solutions to the harmonic equation are bounded. Most modeling methods based on partial differential equations restrict themselves to linear constraints to ensure the solution process is interactive. The drawback with these purely linear methods is that the solution to a problem with non-negative boundary values may occasionally have a solution that is not strictly non-negative everywhere. In practice, this issue makes modeling a desired deformation as a combination of deformations formed by perturbing one shape handle at time less intuitive.

The mathematical solution to this problem is to use a combination of linear equality and inequality constraints to ensure the resulting solutions to the PDE are non-negative. Using the authors’ formulation of these constraints, the resulting problem is both bounded and convex and, as a result, can be solved efficiently using a standard sparse quadratic programming solver. In practice, the method discretizes the shape and pre-computes independent deformations for each individual handle. This main solution process is done as a pre-computation and takes on the order of, at most, seconds for most examples. Since the problem is linear, the desired deformation corresponding to a specific set of handle locations can be computed interactively by taking linear combinations of the pre-computed deformations for each individual handle. Locality and convexity ensure the resulting deformations follow the boundary conditions in an intuitive manner.

The method is an excellent example of the state of the art in deformation methods. The method combines the use of the biharmonic equation to achieve smooth local deformations while restricting these deformations to be non-negative and have no local minima through the use of linear inequality constraints. Since the deformation is computed as a solution to the PDE only over the shape, the resulting deformations depend only on the positions of the handles that are nearby in terms of distance inside the shape. More generally, the paper points in the direction for interesting future work on the use of advanced mathematical methods for representing complex deformations and the use of sophisticated numerical solvers to compute these deformations in practice. 

Joe Warren (jwarren@rice.edu) is a professor of computer science at Rice University, Houston, TX.

Copyright held by Author.

Bounded Biharmonic Weights for Real-Time Deformation

By Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine-Hornung

Abstract

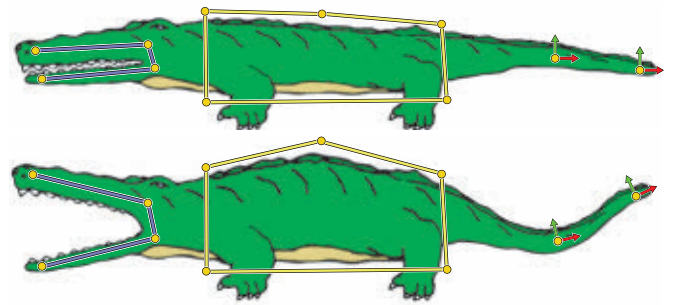
Changing an object's shape is a basic operation in computer graphics, necessary for transforming raster images, vector graphics, geometric models, and animated characters. The fastest approaches for such object deformation involve linearly blending a small number of given affine transformations, typically each associated with bones of an internal skeleton, vertices of an enclosing cage, or a collection of loose point handles. Unfortunately, linear blending schemes are not always easy to use because they may require manually painting influence weights or modeling closed polyhedral cages around the input object. Our goal is to make the design and control of deformations simpler by allowing the user to work freely with the most convenient combination of handle types. We develop linear blending weights that produce smooth and intuitive deformations for points, bones, and cages of arbitrary topology. Our weights, called bounded biharmonic weights, minimize the Laplacian energy subject to bound constraints. Doing so spreads the influences of the handles in a shape-aware and localized manner, even for objects with complex and concave boundaries. The variational weight optimization also makes it possible to customize the weights so that they preserve the shape of specified essential object features. We demonstrate successful use of our blending weights for real-time deformation of 2D and 3D shapes.

1. INTRODUCTION

Interactive deformation is the task of assisting the user to alter an object's shape. In the case of 2D cartoon deformation, we could ask the user to manually reposition each pixel of the image, but this is unnecessarily tedious. The space of coherent configurations of the 2D shape is much smaller than the space of all possible positions for every pixel of the image. Hence, we would rather the user provide only a few, high-level constraints like "open the mouth," "enlarge the belly," or "bend the tail" (Figure 1). The rest of the shape should immediately deform in an intuitive manner. We may interface such high-level constraints to the user with handle structures, like skeletons composed of rigid bones, enclosing cages, and selected regions or points.

With these handles, interactive space deformation becomes a powerful approach for editing raster images, vector graphics, geometric models, and animated characters. This breadth of possibilities has led to an abundance of methods seeking to improve interactive deformation with real-time computation and intuitive use. Real-time performance is critical for both interactive design, where tasks

Figure 1. Our deformation method supports arbitrary combinations of control handles, such as points, bones, or cages. This versatility allows choosing the right tool: bones control rigid parts, cages reshape areas, and points transform flexible parts. Influence weights for each handle are precomputed at bind time, so high-quality deformations can be computed in real time with low CPU usage. Throughout the paper, colored frames illustrate linear transformations specified at point handles.

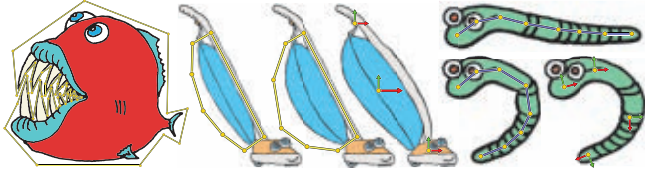


require creative exploration, and interactive animation, where deformations need to be computed repeatedly, often sixty or more times per second. Among all deformation methods, linear blending and its variants dominate practical usage thanks to their speed: each point on the object is transformed by a weighted combination of a small number of affine transformations.

In a typical workflow, the user constructs a number of handles and the deformation system binds the object to these handles; this is termed *bind time*. The user then manipulates the handles (interactively or programmatically) and the system deforms the shape accordingly; this is *pose time*. Unfortunately, existing linear blending schemes are not always easy to use. The user must choose a particular handle type *a priori*, and different types have different advantages (Figure 2). Skeleton-based deformations offer natural control for rigid limbs, but are less convenient for flexible regions. Generalized barycentric coordinates improve volumetric or area control over classical lattice-based free-form deformations, but still require construction of closed or nearly closed cages that fully encapsulate transformed objects and can be tedious to manipulate. In contrast, variational techniques often support arbitrary handles at points or regions, but at a greater pose-time cost.

The original version of this paper was published in *Proceedings of SIGGRAPH '11*, July 2011, ACM.

Figure 2. Different user control structures are appropriate for some situations and inappropriate for others. Setting up an enclosing cage can be both tedious and unintuitive: weaving around the *Pirahna's* teeth. A cage is necessary for precise articulation of the *Vacuum*, where scaling at points would be too crude. Points provide loose and smooth control of the *Worm*, but a skeleton deformation is too rigid and overly complex.



Real-time object deformations would be easier with support for all handle types above: points, skeletons, and cages. Points are quick to place and easy to manipulate. They specify local deformation properties (position, rotation, and scaling) that smoothly propagate onto nearby areas of the object. Bones make some directions stiffer than others. If a region between two points appears too supple, bones can transform it into a rigid limb. Cages allow influencing a significant portion of the object at once, making it easier to control bulging and thinning in regions of interest.

Our goal is to supply influence weights for a linear blending scheme that produce a smooth and intuitive deformation for handles of arbitrary topology (Figure 1). We desire real-time interaction for deforming high-resolution images and meshes. We want smooth deformation near points and other handles, so that they can be placed directly *on* animated surfaces and warped textures. And we seek a local support region for each handle to ensure that its influence dominates nearby regions and disappears in parts of the object controlled by other handles.

Our solution computes blending weights automatically by minimizing the Laplacian energy subject to upper and lower bound constraints. Because the related Euler-Lagrange equations are biharmonic, we call these weights *bounded biharmonic weights* and the resulting deformation *bounded biharmonic blending*. The weights are computed once at bind time. At pose time, points on the object are transformed in real time by blending a small number of affine transformations. Our examples demonstrate that bounded biharmonic blending produces smooth deformations, and that points, bones, and cages have intuitive local influences, even on objects with complex and concave boundaries. Our weight computation requires space discretization and optimization, which could be a drawback in some applications, but the generality of our formulation also makes it possible to provide additional control over the energy minimization, for example, to define weights that preserve the shape of specified essential object features.

2. PREVIOUS WORK

Variational, or energy-minimizing, methods are known to compute high-quality shape-preserving deformations

for arbitrary handles on the surface,^{4,6,9,22} and some variational methods work with bones²⁵ or can be extended to other off-surface handles.⁵ The primary drawback of these techniques is that they rely on optimization at pose time. Although system matrices can be prefactored and back-substitution can be implemented on a GPU, it is not an embarrassingly parallel problem like linear blend skinning, and is therefore much slower. Even with significant performance tuning¹⁹ or model reduction,^{7,23} pose-time optimization is too slow to deform high-resolution objects at high frame rate, as necessary, for example, for video games.

Most methods that are fast at pose time compute the transformation of each point on the object by using a weighted blend of handle transformations. To perform the blending, some methods use moving least squares,¹⁶ some use dual quaternions,¹⁴ but most use linear blend skinning (LBS).¹⁵ With LBS, the affine transformations of the handles are linearly averaged with different weights to transform each vertex. Although linearly blending rotations leads to well-known artifacts, LBS has been a popular technique for skeletal animation for over two decades because it is simple, predictable, and the pose-time computation can be implemented very efficiently on a GPU. In addition to skeletal animation, most cage-based deformation methods^{8,12,13} are effectively LBS, where the handle (cage vertex) transformations are restricted to be translations and the focus is choosing the weights. Additionally, the reduced-model variational shape deformation methods mentioned above use LBS to go from the reduced model to the full model.

The choice of weights for LBS determines whether the affine transformations of the handles affect the shape intuitively. In some cases, weights that have a closed form in terms of the handle structure have been used,^{13,17} but more often they are precomputed at bind time or are painted by hand. In Section 3.1, we formulate the desirable properties of LBS weights, and in Section 3.2 we discuss previous weight choice schemes in the context of these properties.

3. BOUNDED BIHARMONIC WEIGHTS

Our goal is to define smooth deformations for 2D or 3D shapes by blending affine transformations at arbitrary handles. Let $\Omega \subset \mathbb{R}^2$ or \mathbb{R}^3 denote the volumetric domain enclosed by the union of the given shape S and cage controls (if any). We denote the (disjoint) control handles by $H_j \subset \Omega, j = 1, \dots, m$. A handle can be a single point, a region, a skeleton bone (such that H_j consists of all the points on the bone line segment), or a vertex of a cage. The user defines an affine transformation T_j for each handle H_j , and all points $\mathbf{p} \in \Omega$ are deformed by their weighted combinations:

$$\mathbf{p}' = \sum_{j=1}^m w_j(\mathbf{p}) T_j \mathbf{p}, \quad (1)$$

where $w_j: \Omega \rightarrow \mathbb{R}$ is the weight function associated with handle H_j . Note that cages are generally understood as closed polygons in 2D or polyhedra in 3D containing S or

part of it, but our framework is agnostic to the cage topology and treats a cage simply as a collection of simplices, with the requirement that these simplices transform linearly as the cage vertices are translated. Hence, open cages are possible (Figure 3). We do not consider cage faces (line segments in 2D or triangles in 3D) as handles; they receive linear weights, as we will see in Section 3.1. Note also that for skeleton bones connected by joints, we formally include each joint point in one single bone of those that share it (we assume that the skeleton is never torn apart, i.e., that all bones sharing a joint transform the joint to the same location). In practice, we constrain the weights at shared points to be equally distributed between the overlapping bones to maximize the symmetry of our weights.

3.1. Formulation

We propose to define the weights w_j as minimizers of a higher-order shape-aware smoothness functional, namely, the Laplacian energy, subject to constraints that enforce interpolation of the handles and several other desirable properties:

$$\operatorname{argmin}_{w_j, j=1, \dots, m} \frac{1}{2} \int_{\Omega} (\Delta w_j)^2 dV \quad (2)$$

$$\text{Subject to: } w_j|_{H_k} = \delta_{jk} \quad (3)$$

$$w_j|_F \text{ is linear} \quad \forall F \in \mathcal{F}_c \quad (4)$$

$$\sum_{j=1}^m w_j(\mathbf{p}) = 1 \quad \forall \mathbf{p} \in \Omega \quad (5)$$

$$0 \leq w_j(\mathbf{p}) \leq 1, j = 1, \dots, m, \quad \forall \mathbf{p} \in \Omega, \quad (6)$$

where \mathcal{F}_c is the set of all cage faces and δ_{jk} is Kronecker's delta. Figure 4 shows an example of w_j computed for point handles.

The following properties possessed by our weight functions w_j allow for intuitive and high-quality deformations.

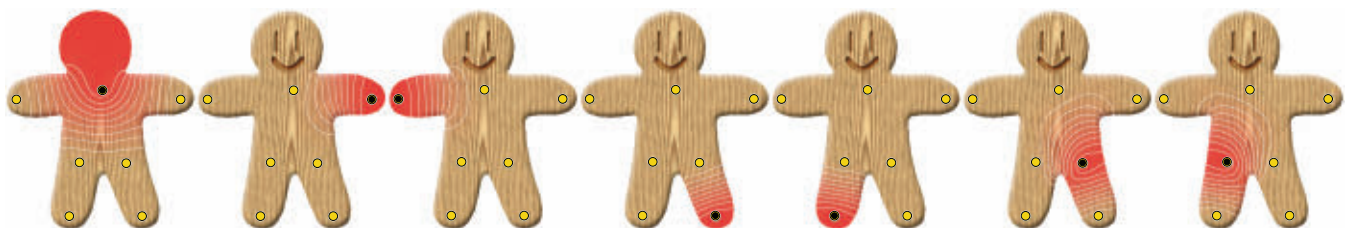
Smoothness: Lack of smoothness at the handles causes visible artifacts in 2D textured shapes (Figure 5) and prevents placing handles directly on 3D shapes. Note that by calculus of variations, minimizing the Laplacian energy (2) amounts to solving the Euler-Lagrange equations, which are the biharmonic PDEs in this case: $\Delta^2 w_j = 0$. Equivalently, we could formulate our blending weights as minimizers of the linearized thin-plate energy, as it leads to the same biharmonic PDE (see e.g., Botsch and Sorkine⁶). The bounded biharmonic weights are C^1 at the handles and C^∞ everywhere else, provided that the posed boundary conditions are smooth. This is always the case, with the exception of skeletal joints and cage vertices: for bones connected by joints, the weights must have a discontinuity at the joints since w_j on bone H_j is 1 and it must be zero on the adjacent bone. However, this does not lead to smoothness problems for the actual deformations because the joints are always transformed to the same location by all emanating bones.

Explicit linear interpolation constraints (4) on cage faces are required to achieve expected behavior, since otherwise the cage faces would not deform linearly when translating cage vertices. These linear constraints on cage faces preclude smoothness of the weights at the cage vertices. Therefore, our deformations are not smooth at cage vertices, but they are smooth everywhere else, including across cage faces.

Figure 3. Deformation of the leaning tower (original is shown left). Cages provide more exact control over area than other handle types.



Figure 4. Bounded biharmonic weights are smooth and local: the blending weight intensity for each handle is shown in red with white isolines. Each handle has the maximum effect on its immediate region, and its influence disappears in distant parts of the object.



Nonnegativity: Negative weights lead to unintuitive handle influences, because regions of the shape with negative weights move in the opposite direction to the prescribed transformation. We explicitly enforce nonnegativity in (6), since otherwise biharmonic functions (as in Botsch and Kobbelt³) are often negative, even if all boundary conditions are nonnegative (right).

Shape awareness: Informally, shape awareness implies intuitive correspondence between the handles and the domain Ω . The influence of the handles should conform to the features of the shape and fall off with geodesic (as opposed to Euclidean) distance. The best shape-aware behavior one can hope for is when the weights w_j depend on the metric of Ω alone and do not change for any possible embedding of Ω . Our weights are shape-aware since the bi-Laplacian operator is determined solely by the metric.

Partition of unity: This classical property (also seen in, e.g., Bézier or NURBS) ensures that if the same transformation T is applied to all handles, the entire object will be transformed by T . We enforce this property explicitly in (5) since nonnegative biharmonic weights do not sum to 1, unlike unconstrained biharmonic weights.

Locality and sparsity: Each handle should mainly control a shape feature in its vicinity, and each point in Ω should be influenced only by a few closest handles. Specifically, if every locally shortest path (in a shape-aware sense) from a point \mathbf{p} to H_j passes near some other handle, then H_j is “occluded” from \mathbf{p} and $w_j(\mathbf{p})$ should be zero. We observed this property of our weights in all our experiments.

No local maxima: Each w_j should attain its global maximum (i.e., value of 1) on H_j and should have no other local maxima. This property provides monotonic decay of a handle’s influence and guarantees that no unexpected influences occur away from the handle. This property was experimentally observed in all our tests; likely, it is facilitated by imposing the bound constraints

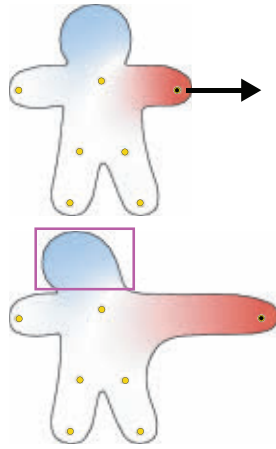
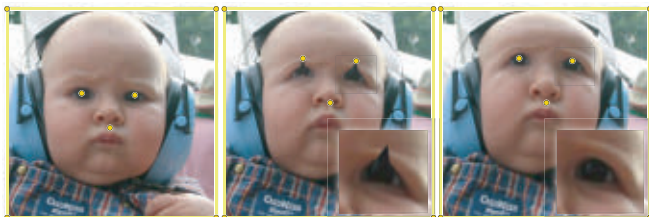


Figure 5. Weights must be smooth everywhere, especially at internal handles, which are likely to correspond to important features. Weights that have discontinuities at handles, like harmonic coordinates (center), introduce tearing artifacts with even slight transformations of the handles. Our weights are smooth, as shown on the right.



(6). Without these constraints, the biharmonic functions in general do not necessarily achieve maxima at the handles and cause deformation artifacts. While bounded biharmonic weights *often* do not have local extrema, it is certainly not the case that they are always monotonic.^a In fact, local extrema invariably appear on shapes with long appendages geodesically equidistant from handles. Dealing with this non-monotonic behavior requires a dedicated optimization.¹¹

3.2. Comparison to existing schemes

Existing schemes formulate and satisfy subsets of these properties, but not all. For example, Shepard’s¹⁷ and similar weights (used in embedded deformation²³ and moving least squares image deformation¹⁶) are dense and not shape-aware. Cage-based schemes do not support arbitrary handles: for example, extending harmonic coordinates¹² to handles in the cage interior results in a lack of smoothness (see Figure 5). Heat diffusion weights² suffer from the same problem. Natural neighbor interpolation²¹ is one of the few schemes that guarantees locality, but it is also not smooth at handles. Biharmonic weights without constraints³ are smooth, but can be negative (or greater than one), have local maxima away from handles, and can result in nonlocal influences. Table 1 shows the properties satisfied by several methods.

A number of methods have most recently been focusing on locally preserving or prescribing angles.²⁴ While they have elegant formulations in terms of complex analysis, these methods are, in general, restricted to 2D.

3.3. Shape preservation

The energy minimization framework supports incorporating additional energy terms and constraints to customize the weight functions. One example of a useful addition is making all points of a specified region $\Pi \subset \Omega$ undergo the same transformation, that is, have all the weight functions be constant on Π ($\nabla w_j|_{\Pi} = 0$). Since typically we only prescribe translations, rotations, and uniform scales at handles, this implies that Π will undergo a similarity transformation in 2D and an affine

Table 1. A summary of the properties of six methods for choosing blending weights. Our method often satisfies all necessary properties. We have empirical evidence but no formal proof for locality and sparsity; our weights often have no local maxima.

Property	Ours	Method			
		[2, 12]	[3]	[17]	[21]
Smoothness	Y	–	Y	Y	–
Nonnegativity	Y	Y	–	Y	Y
Shape awareness	Y	Y	Y	–	–
Partition of unity	Y	Y	Y	Y	Y
Locality, sparsity	Y*	–	–	–	Y
No. local maxima	–	Y, –	–	–	Y

Y*: empirically confirmed, –: often, but not always.

^a Contrary to our original publication’s observations.

transformation in 3D, so that the shape of Π will be preserved. Similarly to the rigidity brush in Igarashi et al.,⁹ the user can paint Π with a (possibly soft) brush, creating a mask $\rho: \Pi \rightarrow \mathbb{R}^+$; we then add a least-squares term to our energy minimization:

$$\sum_{j=1}^m \frac{1}{2} \int_{\Pi} \rho \|\nabla w_j\|^2 dV. \quad (7)$$

See Figure 6, where the shape-preservation brush helped retain the shape of the man's eye while deforming the nose. Note that this is different from placing a handle because no explicit transformation needs to be prescribed by the user; the painted region transforms according to its weights.

3.4. Implementation

We discretize our constrained variational problem (2) using linear finite elements in order to solve it numerically with quadratic programming (we use the flattened mixed finite element method (FEM) formulation for fourth-order problems, as discussed in Jacobson et al.¹⁰). Assuming that the object S is given as a 2D polyline or triangle mesh in 3D, we sample vertices on all provided skeleton bones and cage faces, and mesh the domain Ω in a way compatible with all of the handles and the vertices of S . The result is a triangle/tetrahedral mesh \mathcal{M} whose vertices $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ include all discretized H_j 's and the object itself. The weights become piecewise-linear functions whose vertex values we

are seeking; we denote them by column vectors $\mathbf{w}_j = (w_{1,j}, w_{2,j}, \dots, w_{n,j})^T$.

The Laplacian energy (2) is discretized using the standard linear FEM Laplacian $M^{-1}L$ where M is the lumped mass matrix (with Voronoi area/volume M_i of vertex \mathbf{v}_i on each diagonal entry i) and L is the symmetric stiffness matrix (i.e., the cotangent Laplacian):

$$\begin{aligned} \sum_{j=1}^m \frac{1}{2} \int_{\Omega} (\Delta w_j)^2 dV &\approx \sum_{j=1}^m \frac{1}{2} (M^{-1}L\mathbf{w}_j)^T M (M^{-1}L\mathbf{w}_j) \\ &= \frac{1}{2} \sum_{j=1}^m \mathbf{w}_j^T (LM^{-1}L) \mathbf{w}_j. \end{aligned} \quad (8)$$

We impose the constraints (3)–(6) using the discretized handles. To discretize the additional shape-preservation energy term (7), we employ the linear FEM gradient operator G (see its derivation in Botsch and Sorkine⁶). $G\mathbf{w}_j$ is a vector of stacked gradients, one gradient per element (triangle in 2D and tetrahedron in 3D; since we deal with linear elements, the gradient over an element is constant). Let R be a diagonal matrix containing the integrals of the user brush ρ over each element, and let \bar{M} be the per-element mass matrix (i.e., for each triangle/tet i , \bar{M}_i contains its area/volume). Then the energy term in (7) is discretized as

$$\sum_{j=1}^m \frac{1}{2} \int_{\Pi} \rho \|\nabla w_j\|^2 dV \approx \sum_{j=1}^m \frac{1}{2} \mathbf{w}_j^T (G^T R \bar{M} G) \mathbf{w}_j. \quad (9)$$

Note that the matrix $G^T R \bar{M} G$ is a kind of weighted linear FEM Laplacian, and therefore its sparsity pattern is a subset of the main energy matrix $LM^{-1}L$, creating no new non-zeros. Hence, adding this energy term does not increase the optimization complexity.

We use Triangle¹⁸ for 2D-constrained Delaunay meshing and TetGen²⁰ for constrained tetrahedral meshing to create the discretized domains. In 2D, we configure Triangle to create triangles of near uniform size and shape. For all our 2D examples, Triangle takes less than a second, even for detailed images which require pixel-size triangles. In 3D, we configure TetGen to create rather graded tet meshes to reduce complexity (Figure 7); for the *Armadillo* mesh of 43,234 vertices and 120 vertices sampled internally along bones, the resulting tet mesh has 46,898 vertices. For the *Armadillo* and all our 3D examples, TetGen takes a few seconds.

Figure 6. The generality of our optimization framework makes it possible to compute weights that respect salient object features. Marking a region preserves the shape of an eye (middle), which would otherwise be distorted (right).

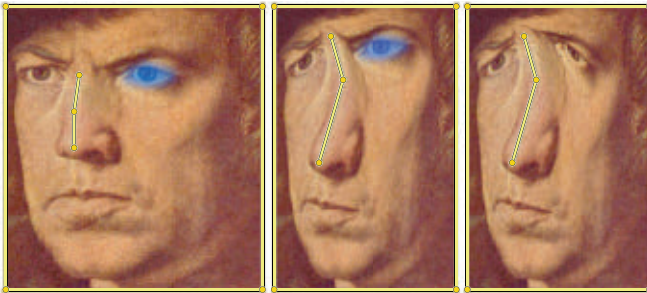


Figure 7. The human skeleton embedded in the *Armadillo* does not control the tail or the ears, but points are easily added for additional expressiveness in each pose. Left: a cutaway of the *Armadillo* shows the graded tet mesh produced by TetGen. The inner tetrahedra are much larger than those near the surface. This keeps the discretization complexity reasonable.



The energy terms in (8) and (9) are quadratic in the unknowns w_j and convex, and (3)–(6) are linear equality and inequality constraints. We use MOSEK¹ as a sparse quadratic programming solver to compute the weights for all of the handles simultaneously.

Since the time necessary to solve the quadratic program is superlinear in the number of unknowns, dividing it into several smaller subproblems allows for a significant speedup. Notice that the optimization of each handle is independent of the rest if we drop the partition of unity constraint (5). We have implemented this strategy, solving for each w_j separately and then normalizing the weights for each vertex in a postprocess. We have observed mostly negligible average differences between this faster solution and the original one, often resulting in visually indistinguishable deformations (Figure 8). Larger differences occasionally occur far from handles, but the weights have the same qualitative behavior: smoothness and observed local support. For the *Gargoyle* in Figure 9, for example, computing the weights for 7 handles separately is 50 times faster than computing them simultaneously. We report the timings as well as the difference between the original and these faster weights in Section 4.

Once the weights are computed, the deformation itself is real-time even for very large meshes, since it is computed with a GPU implementation of linear blend skinning (1).

In the cage-based systems of the various barycentric coordinate methods, the only inputs are the translations of the cage vertices. In our system, the user provides a full affine transformation at each handle. Depending on the

Figure 8. Dropping the partition of unity constraint (5) greatly optimizes the precomputation of our weights without losing quality. Left: the mean absolute difference between the original and faster weights at each vertex, over all handle weights at that vertex. Deformations with the same handle configuration using our original (middle) and faster (right) weights are visually indistinguishable.

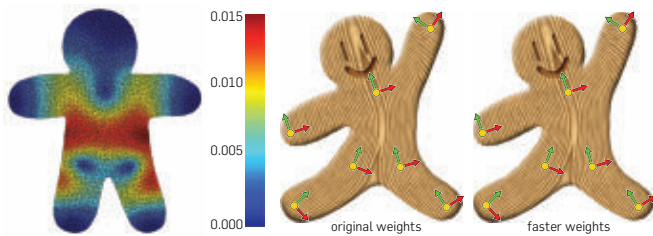


Figure 9. The *Gargoyle* is deformed using an internal skeleton and point handles.



application, the user may choose to specify only translations, that is, identity rotations and scales. However, nontrivial rotations are often necessary to achieve a desired effect, and these could be tedious to specify manually. We found it easier to have rotations inferred from the user-provided translations using run-time optimization in the reduced subspace of linear blend skinning with bounded biharmonic weights, similar in spirit to Der et al.⁷

4. RESULTS

Bounded biharmonic blending combines intuitive interaction with real-time performance. Its controls unify three different interaction metaphors so that simple tasks remain simple and complex tasks become easier to achieve.

Experiments. Points are a particularly elegant metaphor for manipulating flexible objects.⁹ Although similar transformations could be accomplished with bones, the *Octopus* in Figure 10 and *Worm* in Figure 2 illustrate the simplicity of direct point manipulation of supple regions and highlight the inappropriateness of using rigid bones for the same task.

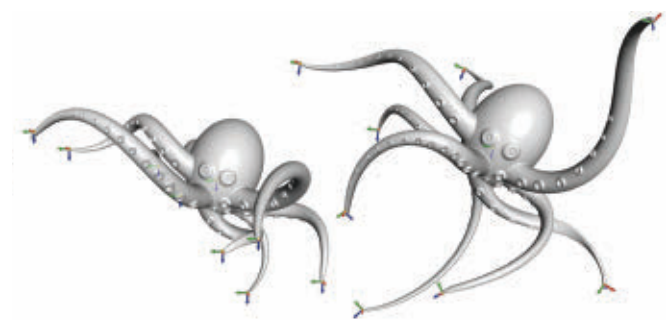
In contrast to previous techniques,^{9, 12} our approach deforms shapes smoothly even when the handle transformations are large. Figure 5 illustrates the importance of smoothness to minimize texture tearing.

We have observed our weights to be local and free of spurious local maxima in all examples tested. Figure 11 compares the support regions of our weights to the biharmonic functions of Botsch and Kobbelt,³ which are globally supported and contain many local extrema.

Some tasks are more easily accomplished by controlling both points and lines. Figure 12 demonstrates our weights with smooth point-based warping while the external cage maintains or resizes the image boundary. Cages are ideally suited for precise area control. In Figure 3, we use an arbitrary collection of open and closed lines to manipulate the shape and orientation of the tower. These deformations and fine adjustments, needed to account for perspective distortions, are difficult to achieve with points or lines alone.

Our approach generalizes naturally to 3D. At bind time, the optimization distributes the weights over the volume so that linear blending delivers smooth deformation at run time. This scheme ensures real-time performance and low CPU utilization even for high-resolution meshes. We note

Figure 10. Points articulate the flexible *Octopus*.



that cages can be even more tedious to set up in 3D than in 2D, particularly when they are required to envelop objects fully. For tasks such as hand manipulation shown in Figure 13 (left), skeletons are easier to embed and use to manipulate a 3D object. Skeletons still suffer from joint-collapse problems and lack the precise volume control offered by cages, and our approach supports and simplifies the combined

Figure 11. Fifty point handles (black and yellow) are randomly placed in a square domain. Left: the sign for the black handle's unconstrained biharmonic weight (red for positive and blue for negative regions). The local maxima and minima are shown as red and blue dots, respectively. Right: the support regions for bounded biharmonic weights of the black handles. In this and all other tests, the weights are local. Here, there are no spurious local maxima.

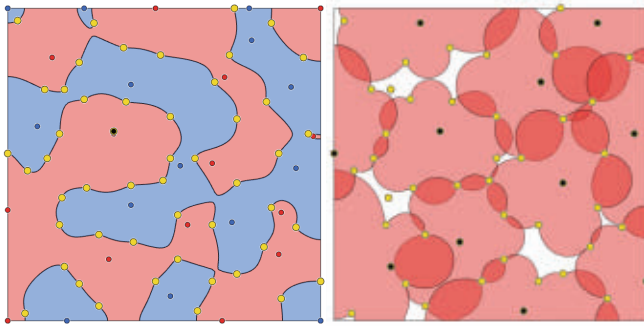


Figure 12. Point handles deform the image by blending the affine transformations specified at each point. The cage on the boundary maintains the rectangular image shape or allows its resizing.

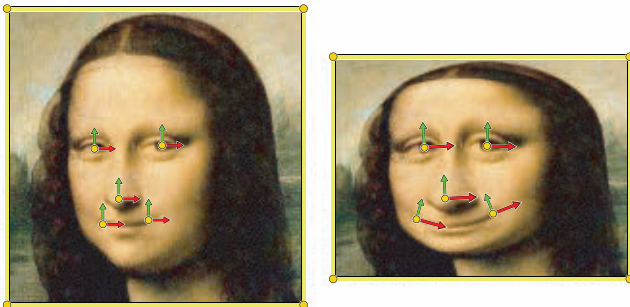
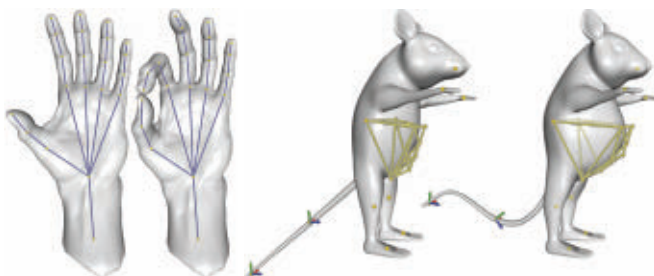


Figure 13. Cages that fully envelop 3D objects such as the *Hand* are difficult to set up. Skeletons are often easier to embed and manipulate. When cages are needed for precise volume control, our scheme makes them easier to use by allowing partial cages that only overlap parts of the object.



use of bones and cages. In particular, our approach supports partial cages that control parts of the object but are not required to surround it fully. Figure 13 (right) shows a simple cage used to enlarge the belly of the *Mouse*. As always, partial cages can be combined with points and bones, and this combined use of all three metaphors is often the most powerful.

Figure 7 shows combined use of points and skeletons. We create a sequence of poses of the *Armadillo* by embedding a human skeleton. The skeleton does not control the tail or the ears, so their deformations need to be adjusted. This is done most directly by attaching a few points. Direct surface manipulation makes it easy to bend the tail into a more realistic pose and expressively curl the ears. Likewise in Figure 9, point handles are a natural and simple choice of control for stretching and bending the wings of the *Gargoyle*. Bounded biharmonic weights combine the motion of the skeleton and configuration of these points to yield smooth deformations.

Discussion. We have tested our method on a MacPro Quad-Core Intel Xeon 2.66 GHz computer with 8 GB memory. The bind time measurements of our unoptimized code are reported in Table 2. One limitation of our solution is the optimization time needed to compute the weights at bind time. We discretized the problem using linear FEM, although other choices may be more efficient, such as the multiresolution framework used in, for example, Botsch et al.⁵ and Joshi et al.¹² Generating bounded biharmonic weights in 3D requires a discretization of the volume. Note that once a volume is computed, an arbitrary embedded object (e.g., polygon soup) may be deformed without regard for its topology.

Our bounded biharmonic weights do not have the linear precision property, that is, they do not necessarily reproduce linear functions. This property is necessary for cage-based deformations (e.g., Joshi et al.¹² and Ju et al.¹³) that apply the deformation solely by interpolating the positions of cage vertices, because otherwise, they would distort the shape when the cage is rotated. In contrast, our approach allows arbitrary transformations

Table 2. Statistics for the various examples in the paper. $|\Omega|$ is the number of triangles of the input 3D model, $|\Omega|$ is the number of elements in the discretization of Ω , BT/h is the bind time per handle in seconds. E_{mean} and E_{max} are, respectively, the mean and max absolute difference between our original weights with (5) enforced explicitly and our faster weights where each handle's weights are solved independently and then normalized. Mean and max values are taken over both handles and vertices.

	$ \Omega $	$ \Omega $	BT/h	E_{mean}	E_{max}
Gingerman		5,040	0.1397	0.0043	0.058
Frowny		5,442	0.0906	0.0045	0.090
Alligator		7,019	0.1779	0.0013	0.055
Pisa		12,422	0.3174	0.0025	0.060
Mona Lisa		32,258	1.2417	0.0050	0.11
Gargoyle	20,000	46,003	1.1939	0.0043	0.18
Hand	28,692	51,263	3.1268	0.0020	0.37
Mouse	26,294	112,355	8.4464	0.0041	0.11
Armadillo	86,442	142,073	12.0870	0.0041	0.40

to be supplied at the handles and blends them over the shape; we therefore do not have to rely on linear precision to be able to work with rotations. A comparison for translational deformation between the linearly-precise harmonic coordinates¹² and our cages is shown in Figure 14 with a rectangular image.

In this paper, we have only experimented with linear blending deformations based on (1); however, our weights are also useful with more advanced methods of transformation blending, such as dual quaternions.^{11,14}

5. CONCLUSION

We have shown how to unify all popular types of control handles for intuitive design of real-time blending-based deformations. This allows users to freely choose the most convenient handles for every task and relieves them from the burden of manually painting blending weights.

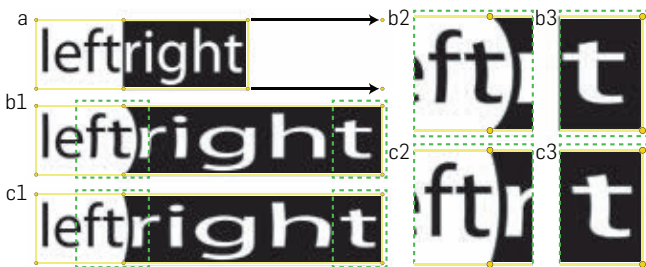
In future work, we would like to optimize the efficiency of our bind-time precomputation. Aside from looking at alternative discretizations and numerical approaches, further analysis will help to reduce the dimensionality of the quadratic program: the observed locality property implies that the weights vanish on significant portions of the domain, which could thus be removed from the minimization.

Skinning-based deformations are prone to foldovers and self-intersections, as the deformation mapping is not always injective. Our method is no exception. Building a reduced model with our weights for simulation and contact handling is worth exploring in this context. We also plan to study the mathematical properties of the bounded biharmonic weights to determine the necessary conditions for which the observed locality and maximum principle hold.

Acknowledgments

We are grateful to Jaakko Lehtinen, Bob Sumner, and Denis Zorin for illuminating discussions, Scott Schaefer for the *Gingerman* and *Pisa* images, and Yang Song for

Figure 14. We show the trade-off between locality and linear precision within a cage. The rest pose of an image of text (a) is stretched horizontally by using harmonic coordinates (b1), and our bounded biharmonic weights (c1). Harmonic coordinates' response is more global (b2) than ours (c2). On the other hand, HC maintains the vertical lines in letter T near the deformed handles (b3), while our weights reveal their lack of linear precision (c3).



helping implement the rigidity brush and 2D remeshing. This work was supported in part by NSF award IIS-0905502, ERC grant iModel (StG-2012-306877), SNF award 200021_137879, an Intel Doctoral Fellowship, and a gift from Adobe Systems. □

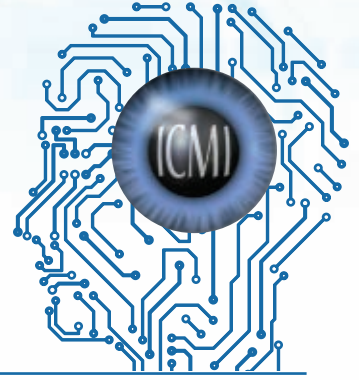
References

- Andersen, E.D., Andersen, K.D. The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. *High Performance Optimization*. H. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds. Kluwer Academic Publishers, 2000, 197–232.
- Baran, I., Popović, J. Automatic rigging and animation of 3D characters. *ACM Trans. Graph.* 26, 3 (2007), 72:1–72:8.
- Botsch, M., Kobbelt, L. An intuitive framework for real-time freeform modeling. *ACM Trans. Graph.* 23, 3 (2004), 630–634.
- Botsch, M., Pauly, M., Gross, M., Kobbelt, L. PriMo: coupled prisms for intuitive surface modeling. In *Proceedings of SGP* (2006), 11–20.
- Botsch, M., Pauly, M., Wicke, M., Gross, M. Adaptive space deformations based on rigid cells. *Comput. Graph. Forum* 26, 3 (2007), 339–347.
- Botsch, M., Sorkine, O. On linear variational surface deformation methods. *IEEE TVCG* 14, 1 (2008), 213–230.
- Der, K.G., Sumner, R.W., Popović, J. Inverse kinematics for reduced deformable models. *ACM Trans. Graph.* 25, 3 (2006), 1174–1179.
- Floater, M.S. Mean value coordinates. *Comput. Aided Geom. Design* 20, 1 (2003), 19–27.
- Igarashi, T., Moscovich, T., Hughes, J.F. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.* 24, 3 (2005), 1134–1141.
- Jacobson, A., Tosun, E., Sorkine, O., Zorin, D. Mixed finite elements for variational surface modeling. In *Proceedings of SGP* (2010).
- Jacobson, A., Weinkauff, T., Sorkine, O. Smooth shape-aware functions with controlled extrema. In *Proceedings of SGP* (2012).
- Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T. Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3 (2007).
- Ju, T., Schaefer, S., Warren, J. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 3 (2005), 561–566.
- Kavan, L., Collins, S., Zara, J., O'Sullivan, C. Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph.* 27, 4 (2008).
- Magnenat-Thalmann, N., Laperrière, R., Thalmann, D. Joint-dependent local deformations for hand animation and object grasping. In *Graphics Interface* (1988), 26–33.
- Schaefer, S., McPhail, T., Warren, J. Image deformation using moving least squares. *ACM Trans. Graph.* 25, 3 (2006), 533–540.
- Shepard, D. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of ACM National Conference* (1968), 517–524.
- Shewchuk, J.R. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *WACG* (1996), 203–222.
- Shi, X., Zhou, K., Tong, Y., Desbrun, M., Bao, H., Guo, B. Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics. *ACM Trans. Graph.* 26, 3 (2007), 81:1–81:10.
- Si, H. TETGen: a 3D delaunay tetrahedral mesh generator, 2003. <http://tetgen.org>.
- Sibson, R. A brief description of natural neighbor interpolation. *Interpolating Multivariate Data*. V. Barnett, ed. Volume 21. John Wiley & Sons, 1981, 21–36.
- Sorkine, O., Alexa, M. As-rigid-as-possible surface modeling. In *Proceedings of SGP* (2007), 109–116.
- Sumner, R.W., Schmid, J., Pauly, M. Embedded deformation for shape manipulation. *ACM Trans. Graph.* 26, 3 (2007), 80:1–80:7.
- Weber, O., Ben-Chen, M., Gotsman, C., Hormann, K. A complex view of barycentric mappings. *Comput. Graph. Forum* 30, 5 (2011).
- Weber, O., Sorkine, O., Lipman, Y., Gotsman, C. Context-aware skeletal shape deformation. *Comput. Graph. Forum* 26, 3 (2007), 265–274.

Alec Jacobson and Olga Sorkine-Hornung [jacobson, Sorkine]@inf.ethz.ch, ETH Zurich, Zurich, Switzerland.

Ilya Baran (baran37@gmail.com), Belmont Technology Inc., Boston, MA.

Jovan Popović (jovan@adobe.com), Adobe Systems, Inc., Seattle, WA.



ICMI 2014

The 16th International Conference on Multimodal Interaction

November 12–16th, 2014

Bogazici University, Istanbul, Turkey

- ✓ Multimodal Interaction Processing
- ✓ Interactive Systems and Applications
- ✓ Modelling Human Communication Patterns
- ✓ Data, Evaluation and Standards for Multimodal Interactive Systems
- ✓ Urban Interactions

<http://icmi.acm.org/2014>

Organising Committee

General Chairs

Albert Ali Salah (*Boğaziçi University, Turkey*)
Jeffrey Cohn (*University of Pittsburgh, USA*)
Björn Schuller (*TUM / Imperial College London, UK*)

Program Chairs

Oya Aran (*Idiap Research Institute, Switzerland*)
Louis-Philippe Morency (*University of Southern California, USA*)

Workshop Chairs

Alexandros Potamianos (*University of Crete, Greece*)
Carlos Busso (*University of Texas at Dallas, USA*)

Demo Chairs

Kazuhiro Otsuka (*NTT Comm. Science Labs, Japan*)
Lale Akarun (*Boğaziçi University, Turkey*)

Multimodal Grand Challenge Chairs

Dirk Heylen (*University of Twente, The Netherlands*)
Hatice Gunes (*Queen Mary University of London, UK*)

Doctoral Consortium Chairs

Justine Cassell (*Carnegie Mellon University, USA*)
Marco Cristani (*University of Verona, Italy*)

Publication Chairs

Alessandro Vinciarelli (*University of Glasgow, UK*)
Zakia Hammal (*Carnegie Mellon University, USA*)

Publicity Chair

Nicu Sebe (*University of Trento, Italy*)

Sponsorship Chair

Aytül Erçil (*Sabancı University, Turkey*)

Local Organization Chair

Hazım Ekenel (*Istanbul Technical University, Turkey*)

Important Dates

Grand challenge proposals	January 15th, 2014
Special session proposals	March 22nd, 2014
Workshop proposals	March 15th, 2014
Long and short paper submissions	May 9th, 2014
Doctoral consortium submissions	July 1st, 2014
Demo proposals	July 15th, 2014

CAREERS

Bucknell University **Visiting Assistant Professor, Computer Science**

The Bucknell University Department of Computer Science invites applications for a one-year visiting position in computer science beginning mid-August 2014. Outstanding candidates in all areas will be considered. Candidates may be teaching undergraduate courses in programming languages and theory of computation. Candidates are expected to have completed or be in the final stages of completing their Ph.D. in computer science or closely related field by the beginning of the 2014 fall semester. A strong commitment to excellence in teaching and scholarship is required.

Bucknell is a highly selective private university emphasizing quality undergraduate education in engineering and in liberal arts and sciences. The B.S. programs in computer science are ABET accredited. The computing environment is Linux/Unix-based. More information about the department can be found at:

<http://www.bucknell.edu/ComputerScience/>

Applications will be considered as received and recruiting will continue until the position is filled. Candidates are asked to submit a cover letter, CV, a statement of teaching philosophy and research interests, and the contact information for three references. Please submit your application to

<http://jobs.bucknell.edu/>

by searching for the "Computer Science Visiting Faculty Position".

Please direct any questions to Professor Stephen Guattery of the Computer Science Department at guattery@bucknell.edu.

Bucknell University, an Equal Opportunity Employer, believes that students learn best in a diverse, inclusive community and is therefore committed to academic excellence through diversity in its faculty, staff, and students. Thus, we seek candidates who are committed to Bucknell's efforts to create a climate that fosters the growth and development of a diverse student body. We welcome applications from members of groups that have been historically underrepresented in higher education.

Centre College **Assistant or Associate Professor** **of Computer Science**

Centre College invites applications for two full time positions (1) tenure-track at rank of assistant or associate professor & (1) one-year visiting at rank of assistant professor in computer science beginning August, 2014. The successful candidate will hold a Ph.D. in computer science or a closely related discipline. For full job description

and to apply, <http://apply.interfolio.com/24475>. For more information about the Computer Science Program at Centre please visit our website (<http://web.centre.edu/csc/>). The deadline for applications is March 17. Centre College is an Equal Opportunity Employer.

Princeton University **Computer Science** **Postdoctoral Research Associate**

The Department of Computer Science at Princeton University is seeking applications for post-doctoral or more senior research positions in theoretical computer science. Positions are for one year with the possibility of renewal.

Candidates should have a PhD in Computer Science or a related field by August 2014. Applications are currently being processed, so we encourage candidates to complete their applications (including letters of recommendation) as soon as possible. Applicants should submit a CV and research statement, and contact information for

three references. Princeton University is an equal opportunity employer and complies with applicable EEO and affirmative action regulations. Apply to: <http://jobs.princeton.edu/>
Req.# 1300791

University of Central Florida **UCF Center for Research in Computer Vision** **Multiple Assistant Professor Positions**

CRCV is looking for multiple tenure-track faculty members in the Computer Vision area. Of particular interest are candidates with a strong track record of publications. CRCV will offer competitive salaries and start-up packages, along with a generous benefits package offered to employees at UCF.

Faculty hired at CRCV will be tenured in the Electrical Engineering & Computer Science department and will be required to teach a maximum of two courses per academic year and are expected to bring in substantial external research funding. In addition, Center faculty are expected



معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute

عضو مؤسسة قطر Qatar Foundation

JOIN THE INNOVATION.

Qatar Computing Research Institute seeks talented scientists and software engineers to join our team and conduct world-class applied research focused on tackling large-scale computing challenges.

We offer unique opportunities for a strong career spanning academic and applied research in the areas of Arabic language technologies including natural language processing, information retrieval and machine translation, distributed systems, data analytics, cyber security, social computing and computational science and engineering.

Scientist applicants must hold (or will hold at the time of hiring) a PhD degree, and should have a compelling track record of accomplishments and publications, strong academic excellence, effective communication and collaboration skills.

Software engineer applicants must hold a degree in computer science, computer engineering or related field; MSc or PhD degree is a plus.

We also welcome applications for post doctoral researcher positions.

As a **national research institute** and proud member of Qatar Foundation, our research program offers a collaborative, multidisciplinary team environment endowed with a comprehensive support infrastructure.

Successful candidates will be offered a highly competitive compensation package including an attractive tax-free salary and additional benefits such as furnished accommodation, excellent medical insurance, generous annual paid leave, and more.

For full details about our vacancies and how to apply online please visit <http://www.qcri.qa/join-us/>
For queries, please email QFJobs@qf.org.qa

[f /QCRI.QA](#) [@QatarComputing](#) [in QatarComputing](#) [YouTube QatarComputing](#) www.qcri.qa

to have a vigorous program of graduate student mentoring and are encouraged to involve undergraduates in their research.

Applicants must have a Ph.D. in an area appropriate to Computer Vision by the start of the appointment and a strong commitment to academic activities, including teaching, scholarly publications and sponsored research. Preferred applicants should have an exceptional record of scholarly research. In addition, successful candidates must be strongly effective teachers.

Applicants must submit all required documents at the time of application which includes the following: Research Statement; Teaching Statement; Curriculum Vitae; and a list of at least three references with address, phone numbers and email address.

Apply URL: <http://www.jobswithucf.com/postings/34681>

Applicants for this position will also be considered for position numbers 38406 and 37361.

UCF is an Equal Opportunity/Affirmative Action employer. Women and minorities are particularly encouraged to apply.

University of North Carolina at Chapel Hill

Carolina Health Informatics Program Multiple Faculty Positions in Health Informatics

Several academic units at The University of North Carolina at Chapel Hill are actively seeking candidates in the broadly defined area of health informatics. The selected colleagues will become core

faculty members in the Carolina Health Informatics Program. The successful candidates are expected to start in the 2014-2015 academic year and their tenure home will be located in one of the CHIP partner units currently recruiting candidates.

Five cross-cutting health informatics areas have been identified as important to all seven academic partners: 1) health data access and management (e.g., data warehouses), 2) ontologies and data standards, 3) analytics, 4) systems modeling and simulations, and 5) clinical decision support systems. At UNC, we are particularly interested in candidates with a strong record of research in one or more of these cross-cutting areas. For general queries regarding the positions, please contact Dr. Javed Mostafa: jm@unc.edu. **More details can be found here: chip.unc.edu/faculty-search.**

University of North Carolina Charlotte Assistant Professor – Cyber Security

The Department of Software and Information Systems in the College of Computing and Informatics at UNC Charlotte invites applicants for a tenure-track faculty position with a focus on cyber security at the Assistant Professor level. The Department is dedicated to research and education in Software Engineering and Information Technology applications, with emphasis in the areas of Cyber Security and Privacy, Modeling and Simulation, Human Computer Interaction (HCI), Design, and Healthcare Informatics. The successful applicant must have a strong commitment to

research and education, with an excellent research record that can attract substantial funding to support Ph.D. students. Applicants from all disciplines in cyber security and privacy are encouraged to apply, particularly those who can demonstrate strengths in one of the following areas: formal methods for cyber and cyber-physical systems, science of security, system security, and security and privacy in big data analytics. The Department of Software and Information Systems offers a Security and Privacy program recognized by the National Security Agency as a National Center of Academic Excellence in Education and Research. The Department and the College of Computing and Informatics also host two research centers focused on the area of security: Cyber Defense and Network Assurance Center and a National Science Foundation IUCRC Center on Configuration Analytics and Automation.

Applications must be made electronically at <https://jobs.uncc.edu> (Position No. 6282) and must include a CV, 3 references, statements of teaching and research, and 3 representative publications. Informal inquiries can be made to the Search Committee Chair at sis-search@uncc.edu. Review of applications will begin in January 2014 and continue until the position is filled. All inquiries and applications will be treated as confidential.

Women, minorities and individuals with disabilities are encouraged to apply. UNC Charlotte is an Equal Opportunity/Affirmative Action employer and is an ADVANCE institution, dedicated to increasing diversity in STEM fields.

Applicants are subject to criminal background check.



Senior Lecturer/ Lecturer

Founded in 1951, National College of Ireland www.ncirl.ie offers full and part-time courses to 3,500 students in two schools, Computing and Business. The School of Computing has greatly expanded in the last few years, becoming one of the biggest computing departments in Ireland with over 1200 students at undergraduate and postgraduate level including a PhD programme in Technology Enhanced Learning and a world-class Cloud Competency Centre. The School has over 60 faculty members with strong industrial links and research excellence in technology-enhanced learning, cloud computing, mobile technologies, data analytics, and parallel/web technologies. The National College of Ireland seeks a senior faculty member to lead and significantly advance the scholarly research in the Technology Enhanced learning area and to teach on School programmes.

The successful candidate will primarily have responsibilities in the areas of conducting and publishing significant scholarly research, facilitating a research culture and agenda broadly within the School, securing new extra-mural collaborative research funding in the area of Technology Enhanced Learning and providing scholarly leadership in the School in general.

The ideal candidate will have a PhD in computer science, informatics, learning technologies, or in a closely related field with a minimum of 5 years at faculty level at a university. They will have an international reputation and record of peer-reviewed scholarly publications in the area of Technology Enhanced Learning, Computing, Informatics or a closely related field with a proven record of leadership in securing extra-mural research funding, preferably in collaboration with colleagues

For further information, please visit the website: www.ncirl.ie/vacancies.
The deadline for applications is 30TH April 2014,
(or until the position is filled).



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 be 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**

Are you looking for your next IT job?

Do you need Career Advice?

The **ACM Career & Job Center** offers ACM members a host of career-enhancing benefits:

- A **highly targeted focus** on job opportunities in the computing industry
- **Access to hundreds** of industry job postings
- Resume posting **keeping you connected** to the employment market while letting you maintain full control over your confidential information
- **Job Alert system** that notifies you of new opportunities matching your criteria
- **Career coaching** and guidance available from trained experts dedicated to your success
- **Free access** to a content library of the best career articles compiled from hundreds of sources, and much more!



Visit **ACM's Career & Job Center** at:
<http://jobs.acm.org>



Association for
Computing Machinery

Advancing Computing as a Science & Profession

The **ACM Career & Job Center** is the perfect place to begin searching for your next employment opportunity!

Visit today at <http://jobs.acm.org>

[CONTINUED FROM P. 112] but small but steady glow. The spikes indicate orbital activity, asteroid mining, and so forth; the glow, a self-contained, self-sustaining planetary civilization.”

Heatherington snorted. “Or radio-activity from fallout!”

“No, no. The gamma’s too minor a component, and too steady.”

“Hmm.” Heatherington sat back. “And what’s your interpretation? Why did they stop space exploration?”

“For the same reason we did,” said Nga. “They found that applying algorithms to existing datasets was a much more efficient way of generating new discoveries than accumulating more data. And that the new knowledge enabled better use of available material resources, which was a much more efficient way of creating new wealth than accumulating ever more new resources.”

“Yeah, tell me about it,” said Heatherington. She banged the side of her chair, steel-rigid and feather-light; smacked her thigh, to which the feeling was already coming back, mere weeks after her accident. Her knee moved reflexively—another improvement, another step to toward, well, a step... The nerve-regeneration technique had been found 10 years earlier, buried in the implications of an obscure and long-forgotten biochemistry paper from the 1990s, unearthed by an algorithm.

Even that metaphor would soon be obsolete. Just as biologists, astronomers, and cosmologists kept making discoveries in historic data, so archaeologists hardly ever needed to dig; they could reconstruct almost every ruin and artifact from ever more subtle implications of surface traces detected by satellite and aerial imaging. Research had become Re: Search. And the better its results, the smaller the budgets that could be justified for expensive new hardware, such as telescopes and space probes...

Heatherington spun her chair around and looked at Nga. “Very nice,” she said. “What are you going to do with it? I think it needs some tightening up.”

“Yes, of course!” said Nga. “I was rather hoping that you...”

“Oh yes,” said Heatherington. She grinned, relaxing now that he cut her

“The spikes indicate orbital activity, asteroid mining, and so forth; the glow, a self-contained, self-sustaining planetary civilization.”


in on it. “Very nice, very nice indeed. It’s beautifully... self-referential, isn’t it? All this time people have been wondering about where the aliens were, and the answer was hidden in the data all along.”

Their paper, “A Possible Resolution of the Fermi Paradox: A Preliminary Analysis of Historical Survey Data,” was published, acclaimed, critiqued, and in due course stored on a prismatic crystal storage device about the size of a human fist and containing the complete astronomical records of every human civilization. Now, some 57 million years later, the storage crystal was drilled out of sedimentary strata from what had once been the bottom of the Atlantic Ocean by an exploration team of the Hrrlllth, the only intelligent species in the galaxy to have failed to invent the electronic computer. Driven by resource shortages to mine every moon and asteroid in their system, they had stumbled on the ancient fragments of exotic matter that had given them their warp drive, and the stars.

They had no idea what the crystal was. The team leader took it back to camp, where the science officer used it to build an optical telescope. □

Ken MacLeod (ken@libertaria.demon.co.uk) is the author of 14 novels, from *The Star Fraction* (Orbit Books, London, 1995) to *Descent* (Orbit Books, London, 2014). He blogs at *The Early Days of a Better Nation* (<http://kenmacleod.blogspot.com>) and tweets as @amendlocke.

© 2014 ACM 0001-0782/14/04 \$15.00



ACM Journal on Computing and Cultural Heritage



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



www.acm.org/jocch
www.acm.org/subscribe



Association for
Computing Machinery

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/2590807

Ken MacLeod

Future Tense Re: Search

For some, data collecting will always be more rewarding than data mining.

IT HAD BEEN a long time since anyone in the circular, domed building at the top of the hill had looked at a screen connected to a remote telescope in Chile or Hawaii; longer still, since anyone other than an excited schoolchild had looked through any of the building's own telescopes. But it was still called an observatory, and within it astronomical discoveries were still being made.

"I've done it," said Nga. "Cracked it."

"What?" Heatherington asked.

The young Malaysian postdoc could not stop grinning. "Found the aliens, and cracked the Fermi Paradox."

Heatherington suspected a leg-pull. The search for extraterrestrial intelligence had long since become a weary joke; the Fermi Paradox, or challenge—"If there are aliens, why aren't they here?"—accepted as unanswerable.

"That's a big claim," she said.

"Oh, the biggest!" He gestured at his screen. "See for yourself."

Heatherington rolled her chair from the doorway to the desk. Nga stood behind her as she scrolled through the data. Even with her years of experience, the figures and diagrams were not easy to interpret. She leaned back and looked up.

"Go on," she said. "Talk me through."

Nga reached across her shoulder, fingers flicking as he tabbed and highlighted, zoomed or shrank.

"Start with terabytes of raw historic data, from the Hubble and Kepler and Spitzer and the Webb and all the rest—everything I could find. Focus on stars we know have habitable planets,



whether or not we already detected organic signatures. The trick is we have decades—centuries, if we include the ground-based observations, as I have—worth of data to work with. Then apply the algorithms I've been building for the past few months, to tease out every tiny fluctuation. Far more subtle than those used to detect exoplanets. I'm

wringing out the last drop of significance here, mind you!" His arm waved, making the screen lurch for a moment. "Sorry. Right. Here you go. Two cases of habitable planets that show multiple spikes of anomalous and clearly artificial wavelengths around them for a century or so, then settle in to a still anomalous [CONTINUED ON P. 111]

Computing Reviews



**BEST OF
2013**

**BEST REVIEWS
NOTABLE
BOOKS & ARTICLES**

Online & Print



Association for
Computing Machinery

ThinkCloud

www.computingreviews.com

Take a look at the **Internet's future.**

Find **the leaders** in data communication
and networking from industry and academia,
all in one place, once a year.



The **Annual Festival** of the **ACM** Special
Interest Group on **Data Communication**

Chicago August 17 - 22, 2014