

04/2013 VOL.56 NO.04

Open Access



interactions' bloggers **always** have something to say



Many of the foremost thinkers in the HCI community—be they up-and-comers or recognized leaders—share their insights, opinions, and observations on *interactions'* popular website.

Join the discussion; argue your stand; get involved!

Follow our bloggers at
<http://interactions.acm.org/blog>

ACM's Career & Job Center!

Are you looking for your next IT job?

Do you need Career Advice?

Visit ACM's Career & Job Center at:

<http://www.acm.org/careercenter>



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 corporate job postings
- Resume posting keeping you connected to the employment market while letting you maintain full control over your confidential information
- An advanced Job Alert system that notifies you of new opportunities matching your criteria
- Career coaching and guidance from trained experts dedicated to your success
- A content library of the best career articles compiled from hundreds of sources, and much more!



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://www.acm.org/careercenter>



Departments

5 **Letter from the ACM Practitioner Board
Developing Tools and Resources
for Those in Practice**
*By Stephen Bourne
and George Neville-Neil*

7 **From the President
Open Access**
By Vinton G. Cerf

9 **Publisher's Corner
An Open Access Partnership**
By Scott E. Delman

10 **Letters to the Editor
A Robot's Roots**

12 **BLOG@CACM
Securing the Future of Computer
Science; Reconsidering
Analog Computing**
Mark Guzdial sees hope in computer
science education efforts in the U.K.
Daniel Reed suggests we should
not be so quick to discard
analog computing.

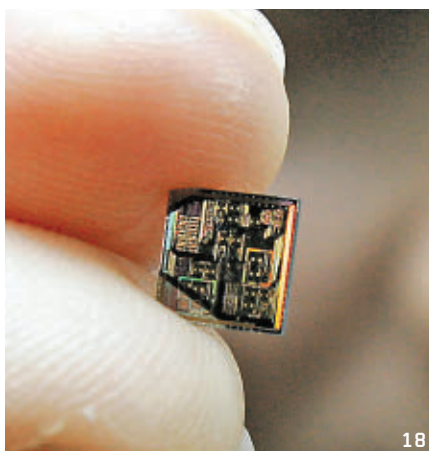
29 **Calendar**

100 **Careers**

Last Byte

104 **Future Tense
Modified Is the New Normal**
How I transcended the baseline for
the sake of art and bioengineering.
By Paul Di Filippo

News



18

15 **Artificial Connections**
Scientists at the Blue Brain Project
are using supercomputers
to simulate neural connections
in a 3D model of a slice
of mammalian brain.
By Nidhi Subbaraman

18 **Inexact Design—Beyond
Fault-Tolerance**
In a new approach to making
computers more efficient, called
“inexact,” “probabilistic,” or
“approximate” computing, errors
are not avoided; they are welcomed.
Some call it “living dangerously.”
By Gary Anthes

21 **Looking Back at Big Data**
As computational tools open up
new ways of understanding history,
historians and computer scientists
are working together to explore
the possibilities.
By Leah Hoffmann

Viewpoints



38

26 **Technology Strategy and Management
Are the Costs of ‘Free’ Too
High in Online Education?**
Considering the economic
implications as educational
institutions expand online
learning initiatives.
By Michael A. Cusumano

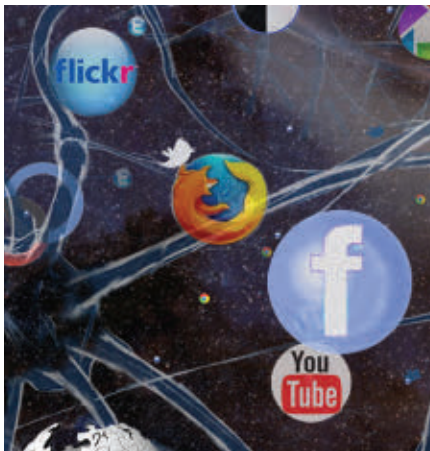
30 **Emerging Markets
Peacebuilding in a Networked World**
Harnessing computing and
communication technologies in
fragile, conflict-stressed nations.
By Michael L. Best

33 **Code Vicious
Code Abuse**
One programmer's extension is
another programmer's abuse.
By George V. Neville-Neil

35 **Viewpoint
Cyber-victimization and
Cybersecurity in China**
Seeking insights into cyberattacks
associated with China.
By Nir Kshetri

38 **Viewpoint
The Problem with Hands-Free
Dashboard Cellphones**
Lawmakers misunderstand user
experience of technology interface.
By Robert Rosenberger

Practice

42 **The Evolution of Web Development for Mobile Devices**

Building websites that perform well on mobile devices remains a challenge.

By Nicholas C. Zakas

49 **How Fast is Your Website?**

Website performance data has never been more readily available.

By Patrick Meenan

56 **FPGA Programming for the Masses**

The programmability of FPGAs must improve if they are to be part of mainstream computing.

By David F. Bacon, Rodric Rabbah, and Sunil Shukla



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

Contributed Articles

64 **Why Computer Talents Become Computer Hackers**

Start with talent and skills driven by curiosity and hormones, constrained only by moral values and judgment.

By Zhengchuan Xu, Qing Hu, and Chenghong Zhang

75 **Offline Management in Virtualized Environments**

How to run virtual machines together with physical machines, especially when sharing computational resources.

By Nishant Thorat, Arvind Raghavendran, and Nigel Groves

Review Articles

82 **Techniques and Applications for Sentiment Analysis**

The main applications and challenges of one of the hottest research areas in computer science.

By Ronen Feldman

Research Highlights

91 **Technical Perspective**
Understanding Pictures of Rooms

By David Forsyth

92 **Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding**

By Huayan Wang, Stephen Gould, and Daphne Koller

**About the Cover:**

Social media sites thrive on opinion-sharing, making sentiment analysis techniques one of the hottest research areas in CS. The ability to mine and monitor opinions about any product, service, or offering holds huge advantages, but nothing is ever as easy as it looks (p. 82). Cover illustration by Charis Tsevis.



Association for Computing Machinery
Advancing Computing as a Science & Profession



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

Ronald Boisvert and Jack Davidson

Members-at-Large

Eric Allman; Ricardo Baeza-Yates;

Radia Perlman; Mary Lou Soffa;

Eugene Spafford

SGB Council Representatives

Brent Hailpern; Joseph Konstan;

Andrew Sears

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

Ronald F. Boisvert; Jack Davidson

Board Members

Marie-Paule Cani; Nikil Dutt; Carol Hutchins;

Joseph A. Konstan; Ee-Peng Lim;

Catherine McGeoch; M. Tamer Ozsu;

Vincent Shen; 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 Directors

Mia Angelica Balaquiot

Brian Greenberg

Production Manager

Lynn D'Addesio

Director of Media Sales

Jennifer Ruzicka

Public Relations Coordinator

Virginia Gold

Publications Assistant

Emily Williams

Columnists

Alok Aggarwal; Phillip G. Armour;

Martin Campbell-Kelly;

Michael Cusumano; Peter J. Denning;

Shane Greenstein; Mark Guzdial;

Peter Harsha; Leah Hoffmann;

Mari Sako; Pamela Samuelson;

Gene Spafford; Cameron Wilson

CONTACT POINTS

Copyright permission

permissions@cacm.acm.org

Calendar items

calendar@cacm.acm.org

Change of address

acmhlp@cacm.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 acmmidasales.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 Prabhakar Raghavan

Board Members

Hsiao-Wuen Hon; Mei Kobayashi;

William Pulleyblank; Rajeev Rastogi

VIEWPOINTS

Co-Chairs

Susanne E. Hambrusch; John Leslie King;

J Strother Moore

Board Members

William Aspray; Stefan Bechtold; Judith

Bishop; Stuart I. Feldman;

Peter Freeman; Seymour Goodman;

Mark Guzdial; Richard Heeks;

Rachelle Hollander; Richard Ladner;

Susan Landau; Carlos Jose Pereira de Lucena;

Beng Chin Ooi; Loren Terveen;

Jeannette Wing

PRACTICE

Chair

Stephen Bourne

Board Members

Erick Allman; Charles Beeler; Bryan Cantrill;

Terry Coatta; Stuart Feldman; Benjamin Fried;

Pat Hanrahan; Tom Limoncelli;

Marshall Kirk McKusick; Erik Meijer;

George Neville-Neil; Theo Schlossnagle;

Jim Waldo

The Practice section of the CACM

Editorial Board also serves as

the Editorial Board of *queue*.

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; James Larus; Igor Markov;

Gail C. Murphy; Shree Nayar; Bernhard

Nebel; Lionel M. Ni; Sriram Rajamani;

Marie-Christine Rousset; Avi Rubin;

Krishan Sabnani; Fred B. Schneider;

Abigail Sellen; Ron Shamir; Yoav Shoham;

Marc Snir; Larry Snyder; Manuela Veloso;

Michael Vitale; Wolfgang Wahlster;

Hannes Werthner; Andy Chi-Chih Yao

RESEARCH HIGHLIGHTS

Co-Chairs

Stuart J. Russell and Gregory Morrisett

Board Members

Martin Abadi; Sanjeev Arora; Dan Boneh;

Andrei Broder; Stuart K. Card; Jon Crowcroft;

Alon Halevy; Monika Henzinger;

Maurice Herlihy; Norm Jouppi;

Andrew B. Kahng; Xavier Leroy;

Mendel Rosenblum; David Salesin;

Guy Steele, Jr.; David Wagner;

Alexander L. Wolf; 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 © 2013 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 acmhlp@cacm.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



Association for
Computing Machinery



Printed in the U.S.A.

Developing Tools and Resources for Those in Practice

The next time you are talking with a software practitioner in his or her twenties or thirties, try asking what he or she thinks of ACM.

To the extent these young practitioners recognize ACM at all, it is likely from dim memories of their undergraduate days. To many of this generation, ACM remains incorrectly stereotyped as an organization for lab-coated researchers or elbow-patched professors but not targeted to individuals like themselves who design, implement, deliver, and deploy software for a living.

The ACM Practitioner Board aims to change that: first, by growing and broadening ACM's offerings to better serve practitioners' needs better; second, by informing practitioners about what ACM already offers that can help them.

Consisting of 10 volunteers led by Stephen Bourne, chair, and George Neville-Neil, co-chair, the Practitioner Board was chartered to help ACM develop products and services to support and enhance the professional and technical development of practicing computing professionals.

In particular, the board aims to:

- ▶ Oversee ACM's professional development resources;
- ▶ Contribute to and direct the development of ACM-specific products and services for computing professionals;
- ▶ Establish programs to increase professional recognition for practicing computing professionals and managers;
- ▶ Provide resources and support for managing a computing career;
- ▶ Develop and maintain foundational models that identify and frame skill sets, jobs, and career paths in computing;
- ▶ Actively promote and advocate for the computing profession; and
- ▶ Evaluate the level of certification appropriate for ACM to award and oversee any ACM certification program.

Perhaps the Practitioner Board's best-known project is *acmqueue*, which first appeared in 2003 as a collection of specially developed technical articles

delivered via a print magazine. Today, *acmqueue* is a website delivering practitioner-oriented technical content with a goal of bridging the gap between research and practice. For many practitioners who are not ACM members, the articles in *acmqueue* are the primary gateway to ACM. In 2012 the *acmqueue* website had a half-million unique visitors; a number that has grown every year over the past decade. The ACM Queue editorial board also develops content for the practitioner-specific section of *Communications*, aptly called Practice; and *acmqueue* content is a primary source of practitioner-oriented content in the ACM Digital Library.

The Professional Development Committee, chaired by Stephen Ibaraki, operates as an arm of the Practitioner Board and develops new products—most recently a series of moderated webinars. In 2012 these webinars covered subjects ranging from the marriage of cloud computing with smart devices, to security, big data, and recommender systems. The Professional Development Committee also oversees the popular Tech Packs, annotated bibliographies from trusted experts who point

users to the most relevant materials for each subject covered.

The Practitioner Board regularly publishes interviews and case studies and more recently has produced video profiles as a way of highlighting work done by impressive up-and-coming practitioners, researchers, developers, and architects. This year the board's efforts are focused on reaching a more global audience. Board member Karin Breitman, of EMC and the Brazilian Computer Society, has begun work on a pilot project to translate *acmqueue* content into Portuguese to reach the thriving practitioner community in Brazil. Efforts are also under way to work more closely with the Indian and Chinese practitioner communities in partnerships with the professional societies that already exist in those countries.

We also realize that many engineers in industry are not ACM members, and we encourage those who seek a career in software to try an ACM membership and become part of the community.

The ACM Practitioner Board hopes all ACM members become more familiar with our offerings—if you are not already. Consider discussing our products with or distributing them to colleagues; and if you are an educator, introduce our offerings to your students.

If you think of something that ACM could or should be doing for practitioners, please let us know. Our goal is to broaden ACM's appeal to everyone working in the computer sciences—from those in the classroom and lab to those shipping software and deploying systems. 

Stephen Bourne is chair and George Neville-Neil is co-chair of ACM's Practitioner Board.

© 2013 ACM 0001-0782/13/04

Our goal is to broaden ACM's appeal to everyone working in the computer sciences.

Latin American eScience Workshop 2013

Sponsored jointly by Microsoft Research and FAPESP
May 13 – 15, 2013 – São Paulo, Brazil

In Brazil, researchers have been working together to increase our understanding of tropical ecosystems, human impact on the environment, biogenetics, and biodiversity. These efforts are providing new opportunities to improve our capabilities in data-intensive research and strengthen the eScience research community. From May 13 to 15, 2013, we will host a special eScience Workshop in the city of São Paulo, Brazil. The event will bring together more than 150 participants, including students and researchers from all over the world, to explore collaboration and research opportunities in areas such as environmental sciences, bioenergy, biodiversity, health and digital humanities. More information about the workshop and the registration can be found at the workshop website.

<http://www.fapesp.br/eventos/latam2013>

World-Renowned Journals from ACM

ACM publishes over 50 magazines and journals that cover an array of established as well as emerging areas of the computing field. IT professionals worldwide depend on ACM's publications to keep them abreast of the latest technological developments and industry news in a timely, comprehensive manner of the highest quality and integrity. For a complete listing of ACM's leading magazines & journals, including our renowned Transaction Series, please visit the ACM publications homepage: www.acm.org/pubs.

ACM Transactions on Interactive Intelligent Systems



ACM Transactions on Interactive Intelligent Systems (TIIS). This quarterly journal publishes papers on research encompassing the design, realization, or evaluation of interactive systems incorporating some form of machine intelligence.

ACM Transactions on Computation Theory



ACM Transactions on Computation Theory (ToCT). This quarterly peer-reviewed journal has an emphasis on computational complexity, foundations of cryptography and other computation-based topics in theoretical computer science.

PLEASE CONTACT ACM MEMBER
SERVICES TO PLACE AN ORDER
Phone: 1.800.342.6626 (U.S. and Canada)
+1.212.626.0500 (Global)
Fax: +1.212.944.1318
(Hours: 8:30am–4:30pm, Eastern Time)
Email: acmhelp@acm.org
Mail: ACM Member Services
General Post Office
PO Box 30777
New York, NY 10087-0777 USA



Association for
Computing Machinery

Advancing Computing as a Science & Profession

www.acm.org/pubs



Vinton G. Cerf

DOI:10.1145/2436256.2436258

Open Access

OPEN ACCESS CONTENT” is a term that holds different meanings depending on the perspective and context. For some, it denotes “free of charge”; for others, it may mean “downloadable.” In principle, most definitions revolve around the notion that content is easily found and freely available. It is a topic that has been widely discussed and, with the advent of the Internet and the Web along with “digital publishing,” it has become an important touch-point for the research community.

For many, discussions about open access ultimately lead to the economics of publishing, which has changed dramatically in recent years, as most content has moved online and print has increasingly become a secondary medium. With print, there is a significant cost to produce physical copies, to distribute, archive, and provide access to them. In the digital world, printing and distribution costs may ultimately disappear, but the creation, distribution, and archiving of digital content rely on more complex technology than print and costs for such technology are not insignificant. Access costs in the digital world are related to digital communication network access while in the physical world they revolve around postage and transportation. And the costs of composing and making digital content discoverable will likely never go away completely. In the digital world, the traditional print functions of a librarian are still needed, since most would agree that full-text search is not a substitute for a skilled research librarian who adds value through context, experience, and personal interaction with a searcher.

It is easy to fall into the trap of comparing apples and oranges when discussing the economics of open access. For example, one might compare the cost of operating a real-world library

with its shelves of books, magazines, CDs, to the cost of a high-density, high-capacity disk-drive system. We must think more broadly about the function of curating, cataloging, and indexing content, and maintaining storage systems (whether print-based or digital) to illuminate the more relevant considerations. There is, however, another trap I find myself falling into occasionally: comparing digital content that is *merely* the equivalent of print content; that is, static papers, text, imagery, and so on. That this is an overly narrow notion is readily understood as one begins to think about interactive presentations of research results, archiving of research data, *and* software needed to interpret, analyze, or present research results.

As our capacity to store digital information grows, it is predictable that the need and appetite for storing research data and analytical software will also grow. There are consequences to this direction. For one, we need to assure the data formats, the analytic software, and other metadata can be preserved *and understood* for hundreds if not thousands of years. Curating the collection of digital reports, interactive applications, and large-scale databases is a substantial challenge. I have written on the topic of *bit rot* in the past and the term applies here as well. Not only do we need to archive the raw bits of reports, data, metadata, and application software, we must also maintain a system context in which all of this material remains accessible and usable.

For some, this leads to the conclusion that we may need software emulators of older hardware, older compilers for past source code languages, OS copies, and instantiations of applications relevant to published research reports and data. One can readily reach the conclusion that maintaining such a diverse archive is challenging and costly.

Another aspect of high-value archiving is completeness. While it is arguable that many unaffiliated parties may maintain some content independently, there is great value in knowing that *all* content of a particular class can be found in a readily accessible archive. The ACM Digital Library is a prime example of an archive valued for its comprehensive character. Indeed, the value and important contribution of some of its content is recognized years after publication. Therefore maintaining a comprehensive collection contributes to scholarship in a critical way.

As the research community moves toward digital publication of content, algorithms, analytical software, data, and metadata, it seems inescapable that business models will be needed to assure the longevity, utility, and comprehensive nature of archival information. There appear to be many ways in which these costs may be defrayed. Research grants may cover some or all of these costs, subscriber fees may provide another path, historical “page charges” in the print world may be replaced by their equivalent for maintaining a sort of *digital vellum* in which content and its surrounding ecosystem can be made eternal and enduring.

This topic is the subject of ongoing discussion throughout ACM. Practical steps toward resolving alternatives are being taken, as detailed by ACM Publications Board co-chairs Ronald Boisvert and Jack Davidson in the February 2013 issue of *Communications* (p. 5).

The recent steps taken by ACM are important but not the end of the story. I look forward to exploring these and other ideas with members of ACM, the research community, and the organizations that sponsor research in all fields, including our own.

Vinton G. Cerf, ACM PRESIDENT

The Ultimate Online Resource for Computing Professionals & Students

ACM DL DIGITAL LIBRARY

<http://www.acm.org/dl>



Association for
Computing Machinery

Advancing Computing as a Science & Profession



DOI:10.1145/2436256.2436278

Scott E. Delman

An Open Access Partnership

During the past 30–40 years, the trend in science publishing has been toward niche publication with literally thousands of highly specialized and targeted journals being launched.

Niche communities supported these publications via submissions, peer review, and readership. Some publishers were better than others at ensuring this information was available or “accessible” to those qualified readers and some publishers and editorial boards were more successful at filtering and selecting, via high-quality peer review, the most important information to disseminate to their communities. The publishers and editorial boards who did this well were rewarded with prestige, high-impact factors, and new quality submissions, which in turn fueled their continued success. Over time, authors knew well which were the best publications in which to publish to benefit their careers and reach their targeted audiences, and qualified readers knew well which publications were worthy of their readership.

Enter the Internet...searching the scholarly literature became far easier. Search engines like Google or Google Scholar in particular made it easier to find articles on specific topics, but of course the algorithms Google and other search engines use to identify and display specific results are a bit of a black box that is changing all the time. Results are mixed and while it is far easier now to find targeted information, the quality of the results is less assured. Everyone knows this and there is little dispute. Nevertheless, the ability to find information easily and ubiquitously is alluring and caused some to question why all information was not at everyone's fingertips equally. The slogan “information wants to be free” became a moral imperative for an impassioned

group of highly vocal and influential scientists (and politicians), who framed the debate as “us vs. them” with publishers on one side and authors, librarians, and readers on the other.

This mentality confuses me. Authors and readers benefited tremendously from this “scholarly publishing” enterprise they helped establish, but the vocal few would have you believe publishers are all evil and created this enterprise all by themselves and are the only ones who benefited. Such black-and-white depictions are rarely true. Publishers (commercial and learned societies alike) took a financial risk in launching new publications by investing in the infrastructure required to disseminate high-quality publications, and niche scholarly communities rewarded the successful publishers with increased authorship, subscriptions, and readership. In return, those associated with the most successful of these publications receive prestige, career advancement, and the visibility that comes with publishing one's work in a respected journal. And despite what is written in the blogosphere, the costs of publishing scholarly information remain significant and the infrastructure required to not only prepare manuscripts for publication but to curate, disseminate, and preserve the literature in a responsible way has become far more complex in recent years. Anyone who carefully studies the infrastructure involved in maintaining a scholarly publishing program like ACM's in comparison to a WordPress blog or basic website will know what I am talking about.

While some publishers have almost certainly taken advantage of their success (ACM is not one of them), the basic value proposition that exists today between scholarly publishers and their niche communities continues to work well for the vast majority of scientists, scholars, and readers worldwide. If this were not the case, one would likely see hundreds of thousands of authors, reviewers, and editorial boards quitting their respective roles with established journals en masse, but this is not happening. The value proposition is almost certainly still there for most and the notion that within these niche communities there are enormous groups of disenfranchised qualified readers who have no way to gain access to the published literature is by and large simply untrue.

This is not to say that opening access to all established scholarly journals worldwide would not result in increased readership and tangible social benefits. Of course it would, and in my opinion we should all work closely together to move definitively toward an open access model for scholarly publishing. But framing this move as a moral imperative and a revolution that must happen overnight, damn the consequences, is the wrong approach and quite frankly an irresponsible one at that. What is much needed is a rational discussion between long-standing successful partners, who acknowledge the mutually beneficial roles each play in the publication process and work closely together to find a sustainable way forward.

Scott E. Delman, PUBLISHER

A Robot's Roots

I WOULD LIKE to add a bit of etymological history concerning the word “robot” to Vinton G. Cerf’s President’s Letter “What’s a Robot?” (Jan. 2013). The Czech word “robota” shares a common root with the Russian “работа” (“rabota”), as well as with the German “Arbeit,” dating to the Dark Ages of idealized German-Slavic unity in the forests of Eastern Europe. The word robota means forced labor and differs from “práce,” which means any kind of work, including that of a free man, as well as creative work. Práce shares a common root with the Greek “πράξις” (“praxis”), inferring free human existence. The accepted wisdom as to the origin of the word robot says that when Karel Čapek, a mid-20th century Czech author (1890–1938), needed a special word for an artificial slave in his 1920 play *R.U.R. (Rossum’s Universal Robots)*, he turned to his brother Josef, who suggested the neologism “robot,” deriving it from “robota.” The Čapek brothers cooperated often, co-authoring several plays and books. Josef was a modern painter (a favorite of collectors today) and illustrated many of Karel’s books, especially those for children. The brothers also embraced English culture and democracy. Karel died shortly after part of Czechoslovakia was annexed by Nazi Germany in 1938, and Josef was arrested by the Gestapo and died in the Bergen-Belsen concentration camp April 1945.

Ivan Ryant, Prague, Czech Republic

Vinton G. Cerf cited a common misconception that the word “robot” is derived from the Russian word “rabota” (work). The origin of robot is actually more subtle: Unlike Russian, which has only one word for work, the Czech language (the native language of Karel Čapek, who coined the term “robot”) has two; the general term is “práce”; the second, “robota” (similar to the Russian word), means “forced labor,” as in the labor of a servant. Čapek chose robota since his intent was for robots to be servants to humanity.

Both the Russian word rabota and the Czech word robota derive from the same Slavic word “rab” (slave), because in earlier times, work could be seen as so undignified, even shameful, no self-respecting noble person would do it. Later, when attitudes changed and work was seen as dignified (and it was shameful to be a non-working social parasite), the original word for work in Russian lost its shameful association, while Czech added a new word to describe the dignity of work.

Vladik Kreinovich, El Paso, TX

Get a Job

In his Viewpoint “What College Could Be Like” (Jan. 2013), Salman Khan wrote how the core value proposition of U.S. higher education is increasingly untenable, as reflected in the rising costs of tuition and in graduate unemployment. In response, he envisioned a new kind of university built around the industry co-op. However, it would do little to address the challenges facing U.S. higher education today. Rather, we should see it as another round in the ongoing attempt by market-based education reformers to de-socialize U.S. higher education.

The recession is the dominant cause of both rising tuition and graduate unemployment today. Tuition has increased to help offset lost revenue from state funding and private endowments, not increased costs.² Along with the long-term issue of rising tuition,¹ the growing cost of student services, facilities, and operations have outstripped the growth in the cost of instruction. Khan’s proposal—focused on reducing the cost of instruction through on-line lectures—would do little to stop or even slow this trend.

Perhaps Khan’s hypothetical university could generate extra revenue through student co-ops. In this respect, his focus on computer science hides a larger issue: CS graduates are in high demand, while the opposite holds for nearly every other discipline. Outside computing, more co-ops (if possible)

would create more undercompensated intern positions, not full-time positions.

But as academics and intellectuals, as well as citizens, we should be more concerned over how Khan’s proposal would reduce education to job training. Where do the ideas of Plato’s *Republic*, Karl Marx’s *Capital*, or Thomas Hobbes’s *Leviathan* fit into a “co-op education”? Or, less ambitious, where does a basic understanding of the U.S. Constitution and system of governance fit? Should we assume these subjects, along with “art and literature,” would be deferred to “nights and weekends” as elective pursuits? Such a proposal fails to treat them as having merit equal to technical skills or as an integral part of a broader humanistic education.

Khan claimed students view college primarily as a means to a job (his core value proposition), yet enrollment in co-ops (available for credit in nearly every U.S. engineering school) remains modest. Rather than free students to pursue co-ops, Khan’s proposal would shackle them to the demands of a nine-to-five job, restricting their freedom elsewhere.

Carved in Bedford Limestone on the main building of my alma mater, the University of Texas at Austin, are the words: “Ye shall know the truth and the truth shall make you free.” Perhaps Khan would substitute “Get a job.”

Gilbert Bernstein, Stanford, CA

References

1. Desrochers, D.M. and Wellman, J.V. *Trends in College Spending 1999–2009*, Sept. 2011; http://www.deltacostproject.org/resources/pdf/Trends2011_Final_090711.pdf
2. Hurlburt, S. and Kirshstein, R.J. *Spending, Subsidies, and Tuition: Why Are Prices Going Up? What Are Tuitions Going to Pay For?*, Dec. 2012; <http://deltacostproject.org/resources/pdf/Delta-Subsidy-Trends-Production.pdf>

Look Beyond North America

Anita Jones’s Viewpoint “The Explosive Growth of Postdocs in Computer Science” (Feb. 2013) covered an important topic but failed to say explicitly that her argument did not include Europe or Asia. The Taulbee survey (<http://cra.org/resources/taulbee/>), from which

she drew her data, is limited to North America, an important job market but not the only one. There are probably more CS faculty and researchers outside (than in) North America. And more than half of the top 200 universities in CS worldwide, as ranked by the Shanghai Jiao Tong University index, are outside North America (<http://www.shanghairanking.com/SubjectCS2012.html>).

Communications missed an opportunity to address the wider ACM audience on a potentially global (not just North American) phenomenon. Moreover, looking beyond North America would be a good way to help achieve the vision Vinton G. Cerf outlined in his President's Letter "Growing the ACM Family" in the same issue.

Toby Walsh, Kensington, NSW, Australia

Less Negative Reviewing, More Conference Quality

Bertrand Meyer's blog post "When Reviews Do More Than Sting" (Feb. 2013) is an opportunity to reflect on how CS academic publishing has evolved since it was first posted at [blog@cacm](http://blog.cacm.acm.org) (Aug. 2011). Meyer rightly identified rejection of 80% to 90% of conference submissions as a key source of negative reviewing, with competitors ready to step in with even higher rejection rates, eager to claim the quality mantle for themselves.

In recent years, we have seen that conference quality can be improved and constructive reviewing facilitated, even when a greater proportion of papers is accepted. At least six conferences, including ACM's Special Interest Group on Management of Data (SIGMOD), Computer-Supported Cooperative Work (CSCW), and High Performance Embedded Architectures and Compilers (HiPEAC), incorporate one or more paper-revision cycles, leading to initial reviews that are constructive rather than focused on grounds for rejection. Giving authors an opportunity to revise also provides a path toward accepting more submissions while still improving overall conference quality.

Analyses by Tom Anderson of the University of Washington and George Danezis of Microsoft Research suggest there is little or no objective difference among conference submissions that reviewers rank in the top

10% to 50%. Many conferences could even double their acceptance rates without diminishing their quality significantly, even as a serious revision cycle would improve quality.

This change in the CS conference process would blend conference and journal practices. Though journal reviews may not always be measured and constructive, on balance they are, and, in any case, revision cycles are a way for conferences to be more collegial.

Jonathan Grudin, Redmond, WA

Simulation Is the Way Forward in AI

Robert M. French's main argument in his article "Moving Beyond the Turing Test" (Dec. 2012) is that the Turing test is "unfair" because we cannot expect a machine to store countless facts "idiosyncratic" to humans. However, the example behavior he cited does not hold up, as I outline here. He was careful in selecting it, as it came from one of his own articles, so, we might be justified inferring that other "quirky" facts about human behavior that might "trip up" a computer are, likewise, also no reason to discard the Turing test.

The example involved the "idiosyncrasy" that humans cannot separate their ring fingers when their palms are clasped together with fingers up-right and middle fingers are bent to touch the opposite knuckle. He then asked, "How could a computer ever know this fact?" How indeed? We did not know it either but discovered it only by following French's invitation to try to separate our own ring fingers. So, too, a computer can discover facts by simulating behavior and compiling results. The simulation would use the computer's model of the anatomy and physiology of human hands and fingers, together with the laws of related sciences (such as physics and biology), to compute the "open and close" behavior of each pair of fingers from some initial configuration.

If the model encapsulates our understanding well enough, the open-and-close motion would be 0 only for the pair of ring fingers. Moreover, following a combination of visualization and logic, an explanatory model might reason why separating the two ring fingers is not possible and under what conditions it might be. One could ask

whether French ever asked a competent specialist why the motion is not possible; I myself have not asked but assume there is some explanation.

Idiosyncratic facts about human behavior are not "unfair." That any behavior can be understood (described computationally) is the fundamental assumption of science.

Most of French's argument about the way forward in AI evolving from brute force with unprecedented volumes of data, speed of processing, and new algorithms should be weighed with a caveat: Trying to sidestep "Why?" belongs in the category of "type mismatch."

Turing thought computers could eventually simulate human behavior. He never proposed the Turing test as the way forward in AI, suggesting instead abstract activities (such as playing chess) and teaching computers to understand and speak English, as a parent would normally teach a child. He said, "We can only see a short distance ahead, but we can see plenty there that needs to be done." I say, let's not be in such a hurry to bid farewell to the Turing test.

Nicholas Ourusoff, New London, NH

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.

© 2013 ACM 0001-0782/13/04

Coming Next Month in
COMMUNICATIONS

Collaboration with a Robotic Scrub Nurse

Strategies for Tomorrow's 'Winners-Take-Some' Digital Goods Markets

The Promise of Consumer Technologies in Emerging Markets

The Science of Computer Science

Moving from Petaflops to Petadata

GPU Ray Tracing

Plus all the latest news about quantum computing and interactive proofs, data and presidential elections, and transient electronics.

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:10.1145/2436256.2436260

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

Securing the Future of Computer Science; Reconsidering Analog Computing

Mark Guzdial sees hope in computer science education efforts in the U.K. Daniel Reed suggests we should not be so quick to discard analog computing.



Mark Guzdial
"The U.K. is Taking Steps to Improve Computing Education in Schools"

<http://cacm.acm.org/blogs/blog-cacm/132934-the-uk-is-taking-steps-to-improve-computing-education-in-schools/fulltext>
September 28, 2011

Google's CEO Eric Schmidt critiqued the lack of computing education in U.K. schools in a recent speech in Edinburgh: "I was flabbergasted to learn that today computer science isn't even taught as standard in U.K. schools. Your IT curriculum focuses on teaching how to use software, but it doesn't teach people how it's made. It risks throwing away your great computing heritage."

Schmidt went on to lament the growing divergence between science and arts and called on educators to "reignite children's passion for science, engineering, and math."

A recent issue of *The Economist* raised the question: "Where is Britain's Bill Gates?" Two of ACM's leaders in computing education, Eric Roberts of Stanford University and Andrew McGettrick of the University of Strathclyde, wrote a letter in reply, to help in understanding that question:

British universities produce too few graduates with the special software-development skills that drive the high end of the industry. Universities in Britain find it harder than their American counterparts to develop innovative teaching and curriculums because of national benchmarks that are often highly prescriptive. Such benchmarks force universities to rely on written exams to measure achievement, which can undermine the all-important spirit of innovation and creativity. Written exams are rarely the best measure of software expertise.

Speaking last year to students at Stanford, Mark Zuckerberg said that

he likes hiring Stanford graduates because "they know how to build things." If British universities could focus more of their attention on teaching students to write applications at the leading edge of the technological revolution, the budding Bill Gateses of Britain would have an easier time of it.

Fortunately, computing educators in the U.K. can point to a couple of areas of real progress. The first is a recently announced effort to teach software development in U.K. schools. The new initiative is welcomed by the British Computer Society and is supported by IT companies like Microsoft, IBM, Cisco, and HP.

The second step may have even greater impact. A report by the The Royal Society, the world's oldest scientific organization, on computing at schools is expected in the next few months. The new report is expected to call for increased computer science education in the primary and secondary grades, and it is expected to get some real attention coming from The Royal Society.

Meanwhile, in the U.S., we are still struggling to get significant computer science into the nation's schools. It is a hard problem because the U.S. education system is so decentralized—literally, the primary and secondary schools are defined at 51 places (in each of the 50 states, plus Puerto Rico). The common core standards (a set of education standards coming from the nation's governors, not from the federal government) have now been finalized. Unfortunately, computer science did not end up being part of those

standards, despite the “Computing in the Core” coalition. While that was disappointing, a new bill was just introduced into the U.S. Congress to bolster K-12 computer science education in the U.S. Part of the new Computer Science Education Act is an effort to help each of the states develop computing education for their programs.

Many of us in the U.S. will be watching carefully the developments in the U.K. We will be eager to see the success of their efforts toward improving computing education, and then we will aim to apply the lessons learned here.



Daniel Reed
**“Analog Computing:
 Time for a Comeback?”**

<http://cacm.acm.org/blogs/blog-cacm/135154-analog-computing-time-for-a-comeback/fulltext>

October 8, 2011

In the early days of the automobile, there was a lively competition among disparate technologies for hegemony as the motive power source. Steam engines were common, given their history in manufacturing and locomotives, and electric vehicles trundled through the streets of many cities. The supremacy of the internal combustion engine as the de facto power source was by no means an early certainty. Yet it triumphed due to a combination of range, reliability, cost and safety, relegating other technologies to historical curiosities.

Thus, it is ironic that we are now assiduously re-exploring several of these same alternative power sources to reduce carbon emissions and dependence on dwindling global petroleum reserves. Today’s hybrid and electric vehicles embody 21st century versions of some very old ideas.

There are certain parallels to the phylogenetic recapitulation of the automobile now occurring in computing. Perhaps it is time to revisit some old ideas.

Analog History

Use of the word “computer” conjures certain images and brings certain assumptions. One of them, so deeply ingrained that we rarely question it, is that computing is digital and electronic. Yet there was a time not so long ago when those adjectives were neither readily assumed nor implied when dis-

cussing computing, just as the internal combustion engine was not de rigueur in automobile design.

The alternative to digital computing—*analog computing*—has a long and illustrious history. Its antecedents lie in every mechanical device built to solve some problem in a repeatable way, from the sundial to the astrolabe. Without doubt, analog computing found its apotheosis in the slide rule, which dominated science and engineering calculations for multiple centuries, coexisting and thriving alongside the latecomer, digital computing.

The attraction of analog computing has always been its ability to accommodate uncertainty and continuity. As Cantor showed, the real numbers are non-countably infinite, and their discretization in a floating-point representation is fraught with difficulty. Because of this, the IEEE floating-point standard is a delicate and ingenious balance between range and precision.

All experimental measurements have uncertainty, and quantifying that uncertainty and its propagation in digital computing models is part of the rich history of numerical analysis. Forward error propagation models, condition numbers, and stiffness are all attributes of this uncertainty and continuity.

Hybrid Futures

I raise the issue of analog computing because we face some deep and substantive challenges in wringing more performance from sequential execution and the von Neumann architecture model of digital computing. Multicore architectures, limits on chip power, near threshold voltage computation, functional heterogeneity and the rise of dark silicon are forcing us to confront fundamental design questions. Might analog computing and sub-threshold computing bring some new design flexibility and optimization opportunities?

We face an equally daunting set of challenges in scientific and technical computing at very large scale. For exascale computing, reliability, resilience, numerical stability and confidence can be problematic when input uncertainties can propagate, and single and multiple bit upsets can disturb numer-

ical representations. How can we best assess the stability and error ranges on exascale computations? Could analog computing play a role?

Please note that I am not advocating a return to slide rules or pneumatic computing systems. Rather, I am suggesting we step back and remember that the evolution of technologies brings new opportunities to revisit old assumptions. Hybrid computing may be one possible way to address the challenges we face on the intersecting frontiers of device physics, computer architecture, and software.

A brave new world is aborning. Might there be a hybrid computer in your hybrid vehicle?

Readers' comments:

Digital computers have one fatal shortcoming—they use a clock (or trigger) to change from one state to another. During (or absent) a clock trigger, they are deaf, dumb, and blind. The more precision in time that is demanded, the more state cycles are forced. Analog devices operate deriving their end functions from input conditions without having to walk through the state changes to get there. If we can ever get over the fact that precision is not accuracy, analog systems may make a comeback. (Digital air data computers are one of the logical absurdities—aerodynamic data has no business going from analog physics through digital math, to analog output (control position) when none of the steps in between have any need of step-wise state change computation.

—James Byrd

It may not be known to this community, but some of us have been doing what Daniel Reed's article suggests. A single-chip analog computer, that can solve differential equations up to 80th order, often faster than a digital computer, and without any convergence problems, was described a few years ago: see G. Cowan, R. Melville, and Y. Tsividis, “A VLSI analog computer/digital computer accelerator”, IEEE Journal of Solid-State Circuits, vol. 41, no. 1 (January 2006), pp. 42–53. There is a lot more that can be done in this area.

—Yannis Tsividis, Columbia University

Mark Guzdial is a professor at the Georgia Institute of Technology. **Daniel Reed** is vice president of Technology Policy at Microsoft.

© 2013 ACM 0001-0782/13/04



MentorNet

e-Mentoring for diversity in engineering and science

The Extra Edge for ACM Members – MentorNet Matching

Have you ever asked yourself:

- What's it like to work in industry?
- What is graduate school like, and is it for me?
- How do I manage a career and a life?

Check out MentorNet!

ACM partners with MentorNet to promote e-mentoring between students and professionals.

- As **protégés**, students gain invaluable career advice, encouragement and support.
- As **mentors**, professionals lend expertise to help educate and inspire young professionals.

Protégés are matched in one-on-one email relationships with mentors—from industry, academia, and government—who have applicable experience in relevant technology, engineering, and scientific fields.

Mentors, Protégés help each other:

“My mentor, Brian Kernighan, helped me navigate graduate school. Having learned the value of mentoring, I became a mentor myself...”

— Mary Fernandez,
Principal Technical Staff Member,
ATT Labs - Research

“I am fortunate to have a mentor who spends his time in not only answering my questions, but also in directing my career path....”

— Emeka (Chukwuemeka
Nwankwo), student at Nnamdi
Azikiwe University Awka, Nigeria

Who can be a protégé?

ACM Members who are Undergraduates, Graduates, Post-Doctoral students, or Untenured Faculty.

Who can mentor?

ACM Professional Members

How does the E-Mentoring Program Work?

1. Register in the MentorNet Community by providing your name, valid email address, and username and password. MentorNet Community Registration form at <http://www.mentor.net/join.aspx>.
2. Sign in and click on the appropriate Find a Mentor or Be a Mentor button.
3. The official e-mentoring relationship lasts approximately 8 months.



Association for
Computing Machinery

Advancing Computing as a Science & Profession

Learn more at <http://www.acm.org/mentor.net> and <http://www.mentor.net/>

Artificial Connections

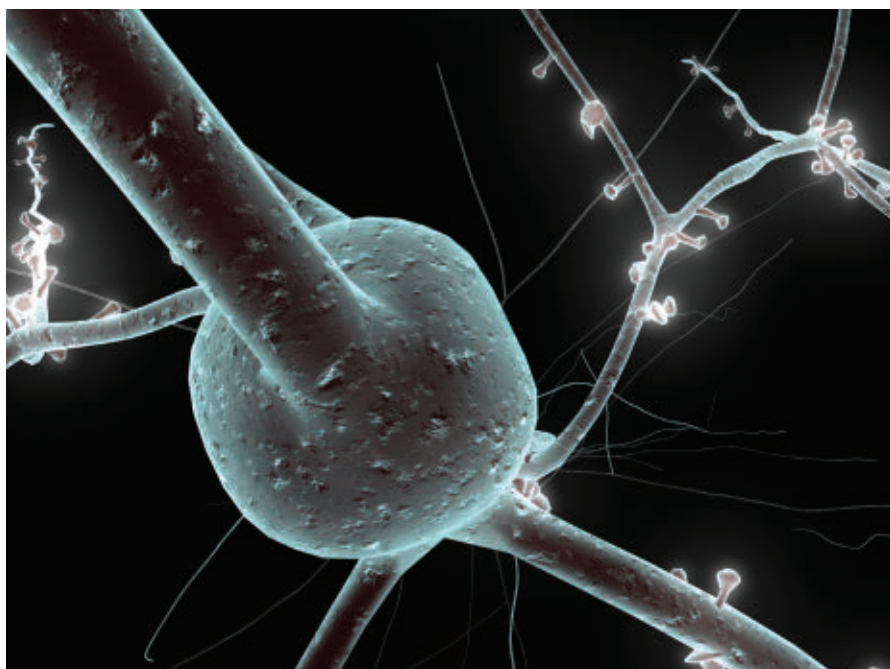
Scientists at the Blue Brain Project are using supercomputers to simulate neural connections in a 3D model of a slice of mammalian brain.

AT ANY GIVEN MOMENT, a live mammalian brain is crackling with electrical activity. Neural impulses are zipping along at the speed of a race car, activating millions of a hundred trillion possible connections. The complexity of the brain makes it incredibly difficult to study experimentally. Neuroscience has broken down brain behavior into piecemeal problems, to understand it in bite-sized parts. For one man, even that is not enough.

Henry Markram, chief of the Blue Brain Project, wants to recreate the brain in all its complexity, from the genetic level up to the molecular level to individual cells that form smaller and larger circuits. Run on advanced supercomputers, this digital brain would incorporate several separate individual theories under one unified model.

So far, researchers at the Blue Brain Project have only simulated small sections of brain tissue. But the hope is that with enough complexity fed into it, a future digital brain will exhibit brain-like complex behavior, and serve as a model system to study brain function, cognition, and disease.

The Blue Brain Project is headquartered at the École Polytechnique Fédérale de Lausanne in Switzerland.



The synapses of a mammalian brain. The Blue Brain Project seeks to create the brain in all its complexity on advanced supercomputers.

The project brings together bench work by experimental neuroscientists, code developed by computational neuroscientists, and simulations crafted by visual design experts. Work continues in partner labs in Israel, Spain, and the U.S. to develop separate parts of the project, which will eventually feed into the master simulation.

Now in its seventh year, the Blue Brain Project has made healthy strides toward its goal. In 2008, Blue Brain Project researchers successfully simulated a tiny sliver of a newborn rat brain called a neocortical column—tissue about 0.5 millimeters across and 1.5 millimeters tall—the equivalent of a processor on a laptop. Once the model was in place,

they gave the virtual structure a jolt of electricity, and watched neural connections form spontaneously in familiar, natural ways—without having been programmed to connect as they did.

The Blue Brain simulations have yet to demonstrate brain-like behavior, like image recognition or simple memory tasks. Some researchers within the field question whether the path to a model brain that reasons involves reproducing the natural brain in exacting, dense, biological detail. Such skepticism about the role of large-scale brain models was raised anew in November 2012, when a group from the University of Waterloo built a model brain called SPAUN, with two million simple neurons that could perform eight different complex tasks—the kinds of tasks the Blue Brain simulations are far from copying. The SPAUN simulation can “remember” a string of elements in a list, or complete a number sequence by identifying the rule that drove it.

Chris Eliasmith, who has been developing the Waterloo approach for more than 10 years, explains that projects like Markram’s, and his own “SPAUN” approach, share a goal: to understand how the nervous system can be organized to give brain-like behavior, although each uses a different approach. “We got actual function,” Eliasmith says; “it’s actually doing thinking.” Theoretical modelers are skeptical about just adding detail to a model, he explains, and their view is that the brain is collectively more than just a collection of neurons.

Work at the Blue Brain Project’s Swiss HQ carries on. A few months ago, a small group within the BBP described another key result: they created a structurally accurate model of a small section of the brain and, using a Blue Gene/P supercomputer, simulated how connections between the neurons form.

It was a successful virtual model of a biological mechanism, but also an example of how simulations based on years of experimental data can yield insights that are difficult to achieve solely through studying slices of brain tissue.

Location, Location, Location

Neurons vary in shape, but are typically stringy structures with a long shaft—the axon—ending in a cell body. Out of the cell body emerge branching structures called dendrites, making the overall neuron structure look something like a skinny tree. When dendrite meets axon, and information is exchanged—that location is called a synapse. Every neuron has between 1,000 and 10,000 synapses.

The formation of a synapse—this active space of exchange between neuron and neuron—is influenced by the structure and location of neurons. A study published in October in *Proceedings of the National Academy of Sciences* sought to answer the broad question: How do neurons form meaningful connections between each other, and what does location have to do with it?

“Where the synapses are positioned on the tree makes a difference to where

that neuron fires,” says Sean Hill, the primary author on the paper and executive director of the International Neuroinformatics Coordinating Facility. Hill is also a former project coordinator on the Blue Brain Project.

Recognizing where neurons form meaningful connections with other neurons is an essential part of understanding how, at the fundamental level, a neural network forms, regenerates, and works. According to existing theories, neurons follow a trail of chemical breadcrumbs that direct their growth and synapse formation. But where did location come in?

In this reconstruction, researchers focused on the neocortical microcircuit of a two-week-old rat. A neocortical microcircuit is a basic computational unit of the brain, a fundamental circuit at the core of what the brain does. Neocortical circuits process sensory inputs like touch and sound, and the associative functions that link them together. The study has broader implications, because neocortical microcircuits are repeated several times in a single brain, and versions of the circuit with very small differences attend to a range of basic cognitive functions. Among mammals, the neocortical microcircuit is repeated with similar design principles.

In one version of the virtual reconstruction, researchers used 298 different kinds of neurons. Every neuron was unique, placed at the location and level in the column in which it was naturally found. In another version of

In Memoriam

Wallace Feurzig, 1927–2013

Wallace Feurzig, a longtime designer and advocate of computer learning environments, died on January 4, 2013.

Born in Chicago in 1927, Feurzig earned a Ph.B. and B.S. from the University of Chicago, and an M.S. in mathematics from the Illinois Institute of Technology. He worked briefly at Argonne National Laboratory and at the University of Chicago, where he hosted the first computer on campus.

Feurzig worked for more than 50 years at Bolt, Beranek & Newman (now BBN

Technologies). He founded the company’s educational technology department, where he developed software tools to enhance students’ self-directed exploration and investigation. He worked with educators in workshops, conferences, and schools to help them implement these new technologies.

Working with colleagues both within and outside BBN, he created some of the first interactive computer programs (Socratic System, Mentor) that sustain investigative dialogue with the student; mathematics

learning environments (Algebra Workbench) that help students carry out tedious manipulations so they can concentrate on strategic issues; Logo, a programming language expressly designed for children, which continues to play a large role in early computer education, particularly in Europe; a language (Function Machines) that represents mathematical functions visually; and computer microworlds that enable students to explore and experiment with topics such as genetics or relativity.

Feurzig wrote several books, many chapter contributions, and hundreds of research papers and reports. His first contribution to *Communications of the ACM*, the article “Algorithm 23: MATH SORT,” appeared in November 1960. Most recently, an article he co-authored with Nancy Roberts, “Computer Modeling in Science Education: Toward a Research and Planning Agenda,” was published in November 1994 as part of the *Proceedings of Supercomputing ’94*, and is available in the ACM Digital Library.

the neocortical microcircuit column, the team engineered the reconstruction with just one kind of neuron, repeated over and over.

Just by arranging the different kinds of neurons at the right level in the 3D space, in the majority of cases, model synapses occurred at locations in which biological synapses would have formed in a natural system. “We had no expectation that just by throwing on the right neurons at the right levels, that it would match the biological data,” Hill says.

The modeled synapses were compared to what Hill calls an “unprecedented dataset”—biological synapse mapping data that the group collected over more than a decade. The team used whole patch clamp techniques—inserting tiny electrodes into adjacent neurons—to check for communication between neurons. Active synapses were marked in tissue by injecting a small quantity of dye.

This simulation offers one possible indication for why there are so many different kinds of neurons in the brain. When different classes of unique neurons were positioned at specific locations in a neocortical microcircuit column, he explains, the synapses in the model formed at the same statistical location every time. With every single neuron within a class constant, the synaptic arrangement is not as predictable or consistent.

“[The work] is a fundamental step, as it gives construction rules for cortical circuitry that accelerates our building of structurally accurate models,” says Felix Schürmann, general project manager at the Blue Brain Project, and a co-author on the PNAS study.

Alongside the cortical simulation, running in parallel, there’s plenty more research reaching goals on the path to a more complete brain model. Other works, published in *PLOS Computational Biology* and the *Journal of Physiology* in the last year, are function studies that bear on the structural details published by Hill and team. Taken together, Schürmann explains, these represent steps that will eventually build up to functional virtual tissue.

This is Your Brain on a Supercomputer

In the meantime, the charismatic leader of the Blue Brain Project, Henry

“We had no expectation that just by throwing on the right neurons at the right levels, that it would match the biological data.”

Markram, has already set his eyes on the next big goal. It is called the Human Brain Project, and anticipates the supercomputing power that science will have access to in the next 10 years. The goal of the Human Brain Project is to scale up the unified computer model followed by the Blue Brain Project, but applying it, this time, to the whole human brain. Like the BBP, the HBP will feed into its model everything neuroscience knows about the human brain, and keep it running on a supercomputer. The simulation will include some, but not all, molecular dynamics that influence the brain, like hormonal influences. For example, the model could factor in the input from, say, the thyroid gland, without needing to simulate the thyroid itself.

“[It] won’t contain all the molecular interaction,” Richard Walker, senior science writer at the BBP clarifies. “We hope the level of molecular interaction will be sufficient.” The functioning of the brain would be tested in stages (as the Blue Brain Project currently does). For example, if the model got as far as a model rat brain, it would be asked to navigate a model maze to see how it performed, Walker explains.

Markram estimates that scientists could be running simulations of the whole brain within the next decade using exaflop-scale supercomputers, and has begun rallying support for this grander Human Brain venture. With more than 100 collaborators at 87 institutions waiting at the ready, Markram in late January received approximately €1 billion in funding from the European Commission as part of the Future and Emerging Technologies Flagships Initiatives.

European Commissioner for IT Neelie Kroes says the objective of the program, which will provide the funding over a 10-year period, is to “keep Europe competitive, to keep Europe as the home of scientific excellence.”

Markram’s plan is that other scientific groups will borrow time on the whole brain simulator to develop new diagnostic tests for diseases, or test new theories about how diseases progress and develop. “[The HBP] will furnish a framework for what we know, while enabling us to make predictions of what we do not,” Markram wrote in his introduction to the Human Brain Project published in *Scientific American* in June 2012. “Those predictions will show us where to target our future experiments to prevent wasted effort.”

In the coming decade, Markram anticipates advances in supercomputing that will make the Human Brain Project possible, as well as limitations to that same computational power needed to run such a brain. He believes the Human Brain Project itself will suggest new insights for how to build supercomputers based on the brain’s natural design, giving back to the same field that made it possible in the first place. **C**

Further Reading

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., Rasmussen, D. A large-scale modeling of the functioning brain. *Science* 338, 1202–1205, 20, Nov. 2012.

Hill, S. L., Wang, Y., Riachi, I., Schürman, F., Markram, H. Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits, *Proceedings of the National Academy of Sciences*, 18, Sept. 2012.

Markram, H. The Blue Brain Project, *Nature Reviews Neuroscience*, 7, Feb 2006.

Markram, H., and Segev, I. *Augmenting Cognition*, EPFL Press, Lausanne, Switzerland, 2011.

TEDGlobal
Henry Markram at TED 2009: Supercomputing the brain’s secrets, http://www.ted.com/talks/henry_markram_supercomputing_the_brain_s_secrets.html, July 2009.

Nidhi Subbaraman is a freelance science and technology writer based in New York City.

© 2013 ACM 0001-0782/13/04

Inexact Design— Beyond Fault-Tolerance

In a new approach to making computers more efficient, called “inexact,” “probabilistic,” or “approximate” computing, errors are not avoided; they are welcomed. Some call it “living dangerously.”

KRISHNA PALEM, a computer scientist at Rice University, has an unorthodox prescription for building faster computers. “If you are willing to live with inexact answers, you can compute more efficiently,” he says. If you ask him whether one of his processors is working correctly, he’s apt to answer, “Probably.”

He’s not joking, and his reasoning is based on two sound principles. Palem has shown that you can get dramatic energy savings by slightly relaxing the requirement that a computer be 100% correct. Further, he says there are many applications where some errors are completely acceptable and even unnoticeable by users.

Palem’s ideas, which date to 2003, sound like a kind of fault-tolerance, a discipline that dates to the dawn of computing. There is, however, a crucial difference: traditional fault-tolerance aims to produce reliable results from unreliable components; Palem’s devices are intended to produce unreliable results from *unreliable* components.

In the Beginning

In 1952, the computing pioneer John von Neumann, in a lecture at Caltech on “error in logics,” argued that computational errors should not be viewed as mere “accidents,” but as “an essential part of the process.” That is, error has to be treated as an intrinsic part of computing.

In subsequent years, the relatively unreliable electromechanical relays and vacuum tubes of von Neumann’s day gave way to highly reliable semiconductor chips, and worries over errors seemed less pressing, except in mission-critical applications where elaborate fault-tolerant (error-correcting) mechanisms sometimes were required.



An application-specific integrated circuit (ASIC) chip developed by a team from Nanyang Technological University and Rice University. In 2012, that team, with colleagues from other institutions, unveiled an “approximate” computing PCMOs chip that is 15 times more efficient than standard chips.

Now, Palem argues, the pendulum is swinging back, as the march of Moore’s Law drives computer circuits to ever-smaller sizes. At the nano scale, they become increasingly difficult to manufacture uniformly and are more likely to fail. At the smallest dimensions, energy waste and heat barriers start playing an increasingly dominant role.

Indeed, energy use has become the grand challenge of computing. *The New York Times* recently (Sept. 22, 2012) reported that in 2010, data centers used 2% of all the electricity consumed in the U.S., with Google’s data centers alone consuming almost 300 million watts. According to the newspaper, Amazon had been cited for 24 violations of air-quality regulations over a three-year period in Northern Virginia, mostly from its big arrays of diesel-powered backup

generators. And energy use is an issue at the level of individual users, with people carrying more and more portable devices featuring power-hungry applications like video that drain their batteries ever faster.

The semiconductor industry’s latest (2011) International Technology Roadmap for Semiconductors puts the issue in stark terms: “Power consumption has become the key technical parameter that controls feasible semiconductor scaling: device roadmaps are now power-driven, and operating frequencies have flattened in several key market domains.”

In 2003, Palem published his idea for saving energy. “I proposed the idea of designing computing switches that you know are going to be inaccurate, with the amount of inaccuracy

being a design parameter,” he says. These designs are now referred to as “inexact designs.”

In an inexact design, no attempt is made to correct for some inaccuracies; no attempt is made to ensure absolute reliability as von Neumann recommended. The designs are appropriate in applications such as video, where an incorrect pixel here or there is unnoticed or can be tolerated; in audio, where a slight distortion is acceptable, or in machine learning (classification) problems. A computing process that works by iteration might converge faster if it makes a few errors on the way, and it can still provide a satisfactory result. Some of those applications exist in mobile devices, where energy savings for long battery life are especially desirable.

Palem and his colleagues tried out the idea, which they also call “probabilistic computing,” first in CMOS. In 2004, they showed in simulations that “probabilistically correct CMOS gates” (or switches), since dubbed PCMOs gates, could be made 300% more energy efficient by reducing the probability of accuracy 1.3%, from .988 to .975.

In May 2012, researchers including Palem and colleagues at Nanyang Technological University (NTU) in Singapore, Switzerland’s Center for Electronics and Microtechnology, and the University of California at Berkeley unveiled a PCMOs adder that produced an incorrect value .25% of the time, but was 3.5 times more efficient as a result. By relaxing the probability of correctness to .92, the chips became 15 times more efficient. Palem is director of the Rice-NTU Institute for Sustainable and Applied Infodynamics, which did the work.

Palem points out that in the number 5,263, the “3” is much less important than the “5.” In adding two such numbers, what if the processor simply ignored the low-order digit (the low-order bits)? Perhaps it wouldn’t matter to the usefulness of the application. He and his colleagues have done just that, by a process called “circuit pruning,” in which the chip is literally stripped of the energy-consuming circuitry needed for the low-importance bit processing. “Not all bits are created equal,” Palem says. “This allows us to design computer circuits where the value of information guides investment.”

“Not all bits are created equal. This allows us to design computer circuits where the value of information guides investment.”

Electing to use single-precision rather than double-precision math in an application is in essence a kind of “bitwise pruning,” says Avinash Lingamneni, a graduate student at Rice who has worked with Palem. But that is often too crude a method, he points out, because the optimum amount of pruning might lie between the two. Also, he says, “we found that instead of reducing the bit-width, we can actually modify the structure and parameters of the arithmetic operators themselves (which are often designed for the worst-case scenarios), leading to even more resource-savings than what could be obtained by bit-width pruning for the same accuracy loss.”

Related to circuit pruning is a principle called “confined voltage scaling.” The pruning results in lower energy use, but also speed improvements; for example, pruning might result in a threefold energy savings and a twofold speed increase. But if the designer is willing to go with the same speed as the original chip, then the pruning results in the chip running at a lower voltage, and energy savings rise to a factor of 4.8 in Palem’s studies.

Application-Driven Designs

So far, the work of Palem and his colleagues has been limited to hardware, via static pruning. However, dynamic “gating” of portions of the chip by software is also possible. An application could choose to forgo using portions of the chip if the application designers felt the benefits in energy savings or speed were worth the loss of accuracy. The logic added for gating comes at a cost, because the instructions required

Milestones

NAE Honors

NATIONAL ACADEMY OF ENGINEERING NEW MEMBERS, FOREIGN ASSOCIATES

The National Academy of Engineering (NAE) has elected 69 new members and 11 foreign associates, bringing its total U.S. membership to 2,250 and the number of foreign associates to 211.

New members in the computer sciences include:

- Anant Agarwal, president of online learning initiative edX and professor, electrical engineering and computer science department, Massachusetts Institute of Technology.
- David L. Dill, professor, department of computer science, Stanford University.
- Abbas El Gamal, Hitachi America Professor in the School of Engineering and professor and chair, department of electrical engineering, Stanford University.
- Edward W. Felten, professor of computer science and public affairs and director, Center for Information Technology Policy, Princeton University.
- Helen Greiner, CEO and founder, CyPhy Works Inc.
- Eliyahu (Eli) Harari, co-founder, retired chairman, and CEO, SanDisk Corp.
- Maurice Herlihy, professor of computer science, Brown University.
- Eric Horvitz, distinguished scientist and managing co-director, Microsoft Research.
- Pradeep S. Sindhu, vice chairman, CTO and founder, Juniper Networks.
- Ken Xie, founder, president, and CEO, Fortinet Inc.
- Elias Zerhouni, president, global R&D, Sanofi.

New NAE Foreign Associates in the computer sciences include:

- Shlomo Shamai (Shitz), William Fondiller Chair in Telecommunications, department of electrical engineering, Technion-Israel Institute of Technology.
- Ji Zhou, president, Chinese Academy of Engineering.

to turn portions of the chip on and off also consume energy.

Of course, a small error in a single calculation might compound unacceptably through a chain of operations, each of which makes a similar small error. But Rice's Lingamneni says most of the applications targeted so far have small chains of serial calculations. Also, it is often possible to design successive blocks of instructions so that a given block is likely to offset the error from the previous block. "We have algorithmic frameworks in place [for] cascaded inexact computations, especially for embedded and signal processing applications," he says. The algorithms favor approaches where errors are additive and hence accumulate relatively slowly, and not multiplicative, where they might grow very rapidly.

Indeed, hardware-based pruning requires some knowledge of what will run on the chip. "We are studying workloads where the behavior of the application tells us how to distribute the error," Palem says. "You tune it to match the needs of the application." For example, researchers at the Rice-NTU Institute are exploring the use of inexact circuits in an energy-efficient hearing aid that ultimately could fit inside the ear canal. A multidisciplinary team that includes neuroscientists and a linguist is studying the hearing process to determine which parts of the audio signal are most important and which parts can be ignored. Researchers are also designing a low-power graphics controller for communities without electricity, and a high-efficiency "machine-learning engine" for pattern recognition, dictionary classification, and search.

The principles of probabilistic computing are applicable elsewhere in computer systems, Palem says, adding that he and his colleagues are also looking into their use in memories that are less than perfect. Further out, he predicts their use in nanoscale devices such as molecular computers, which are inherently prone to error.

IBM's Blue Gene series of massive supercomputers uses both the traditional concepts of fault-tolerance and the newer concepts behind probabilistic computing, which it calls "approximate computing." "The Watson 'Jeopardy' machine is a deep Q&A machine, where the answer is not required to

"We are studying workloads where the behavior of the application tells us how to distribute the error. You tune it to match the needs of the application."

always be precisely correct and you can still win the game," says Pradip Bose, IBM's manager of reliability- and power-aware microarchitectures.

The system can often forgo exactness at a low level as well. Bose offers this example: a program has a statement, "If $A > B$, then do..." Conventional computers will precisely compute A , precisely compute B , then precisely compare them, he says. Yet the logical test can possibly be done satisfactorily by comparing just a few bits. "You can save a lot of power by not always doing the full computation," he says.

Neural systems, which attempt to mimic the brain, are the subject of much research (see "Artificial Connections," p. 15). "The brain's computations are very inexact, and these architectures are following the same idea," Bose says. "Humans have no trouble walking without solving differential equations in their brains, as a robot might. What we do is approximate, but we don't fall."

Traditional fault-tolerant systems, which correct for errors, could be adapted to fail probabilistically and reap efficiencies thereby, Bose says. For example, IBM's production POWER7 server uses a technique called Guardband to save energy by dynamically lowering operating voltage, a process Bose calls "living dangerously" because of the increased probability of processor error. Guardband blocks these errors by dynamically monitoring results and decreasing the processor's clock frequency if errors start to occur. However, says Bose, if the application will tolerate some error, you could allow for lower voltages and en-

ergy consumption. "If you are willing to live even more dangerously, then you can save even more power," he says.

"Living dangerously" may be the price we pay to help solve the computer power problem, but that will require a good understanding of the applications that will run on these energy-efficient systems. William Pulleyblank, a professor of operations research at the U.S. Military Academy and a former Blue Gene architect at IBM, puts it this way: "If it's a video game, and rocks are tumbling over a cliff, and a rock defies gravity for a second, I may not care. But if it's a controller in the Space Shuttle, I may care a lot." **C**

Further Reading

Chakrapani, L., George, J., Marr, B., Akgul, B., Palem, P.

Probabilistic design: a survey of probabilistic CMOS technology and future directions for terascale IC design, VLSI-SoC: Research Trends in VLSI and Systems on Chip, December 2007. <http://www.ece.rice.edu/~al4/visen/2008springer.pdf>

European Semiconductor Industry Association, Japan Electronics and Information Technology Industries Association, Korean Semiconductor Industry Association, Taiwan Semiconductor Industry Association, U.S. Semiconductor Industry Association
International roadmap for semiconductors, 2011. <http://www.itrs.net/Links/2011ITRS/Home2011.htm>

Lala, Parag K.

Self-checking and fault-tolerant digital design, Elsevier Science, The Morgan Kaufmann Series in Computer Architecture and Design, June 2000. http://www.elsevier.com/wps/find/bookdescription.cws_home/677912/description#description

Palem, K., Chakrapani, L., Kedem, Z., Lingamneni, A., Muntimadugu, K.

Sustaining Moore's law in embedded computing through probabilistic and approximate design: retrospects and prospects, International Conference on Compilers, Architectures, and Synthesis for Embedded Systems, Grenoble, France, October 11–26, 2009 <http://www.ece.rice.edu/~al4/visen/2009cases.pdf>

Shanbhag, N.R., Mitra, S., de Veciana, G., Orshansky, M., Marculescu, R., Roychowdhury, J., Jones, D., Rabaey, J.M.

The search for alternative computational paradigms, Design & Test of Computers, IEEE, July–Aug. 2008 <http://dl.acm.org/citation.cfm?id=1440404.1440427>

Gary Anthes is a technology writer and editor based in Arlington, VA.

© 2013 ACM 0001-0782/13/04

Looking Back at Big Data

As computational tools open up new ways of understanding history, historians and computer scientists are working together to explore the possibilities.

MAKING SENSE OF reams of data seems like a uniquely modern problem. Yet historians have been doing it for centuries, reviewing archived sources, constructing analytical frameworks, and fashioning stories and arguments. Now, computational tools—along with a proliferation of digital source materials—are opening up new ways of understanding history, and historians and computer scientists are coming together to explore the possibilities. Housed under the broader rubric of digital history (which encompasses everything from digital publishing to digital texts and research resources), the nascent field of computational history promises nothing less than to change the way we interact with the past.

For Adam Jatowt, a professor at the University of Kyoto's Department of Social Informatics, computational history offers a way to integrate a personal interest in history with his research in document summarization techniques. After receiving a 2011 research grant from the Japan Science and Technology Agency, Jatowt and collaborator Ching-man Au Yeung began to investigate the concept of collective memory through the large-scale text mining of a dataset of international news articles. First, they analyzed the frequency of references to past years; unsurprisingly, with the exception of peaks that correspond to crucial events like the end of World War II, these tend to decline over time. (Jatowt refers to the decline, which is exponential, as the "remembering curve," in homage to the "forgetting curve" through which 19th century psychologist Hermann Ebbinghaus described the decline of personal memory retention.)

Jatowt and Yeung then used a com-

"How many documents can a historian read each year? Machines can scan a much larger amount of content and find connections."

mon topic detection model known as Latent Dirichlet Allocation (LDA) to obtain the probability distributions of topics in their corpus. This enabled the researchers to compare the topics that are now linked to a given year with

those actually discussed during that year, "so we can compare the current view of the past with what was popular at the time," Jatowt explains. Some events were important in their day and are now forgotten; the importance of other events is clear only in retrospect. Jatowt and Yeung give the example of the year 1978, which is now thought to mark the end of China's Cultural Revolution; at the time, they point out, Chinese media focused more on events like the signing of the Treaty of Peace and Friendship between China and Japan.

"Computational history is a complement to the historian's work," says Jatowt. "How many documents can a historian read each year? Machines can scan a much larger amount of content and find connections."

Benjamin Schmidt, a graduate student in Princeton's history department and visiting graduate fellow at



An 1885 U.S. railroad map. The collaborative project *Railroads and the Making of Modern America* leverages computation to visualize the growth of the railroad network by exploring data from cartoons, poetry, maps, timetables, and abandoned track lines.

the Cultural Observatory at Harvard, agrees. “Computation is really the only way to grapple with large datasets,” he asserts. “You can’t read a few books and understand things like the way language changes over time.” Yet computational history, says Schmidt, has the potential to do more than crunch data in support of traditional theses about the past. It also gives researchers a way to make sense of resources that are intractable to traditional scholarship. On his blog Sapping Attention, Schmidt has experimented with visualizing the vast amounts of data from shipping logs, enabling viewers to see ships’ trajectories over time. “Data lets historians tell engaging stories that aren’t narratives, and that tap into a source of explanations slightly removed from the actions of individuals or networks,” he wrote in one post.

Researchers at Google explored the point in a recent study of fame. Citing the popularly held belief that technology has shortened both news cycles and attention spans, James Cook, Atish Das Sarma, Alex Fabrikant, and Andrew Tomkins mined a collection of 20th century news articles to analyze whether the average famous person’s time in the spotlight—as measured by the duration of a single news sto-

ry about that person and the overall duration of public interest in him or her—has changed over time. The conclusion: “through many decades of rapid technological and societal change, through the appearance of Twitter, communications satellites, and the Internet, fame durations did not increase, neither for the typical case nor for the extremely famous.”

Computational history offers a way to generate dynamic insights as well. Historian William Turkel, a professor at the University of Western Ontario, cites the example of the Old Bailey On-

“Computation is really the only way to grapple with large datasets. You can’t read a few books and understand things like the way language changes over time.”

line, which contains records of the trials held at London’s central criminal court between 1674 and 1913. “The records have been marked up with XML, making it easy, say, to discover how many people were convicted of coining offenses in the 1770s,” he explains. Through the OB Online’s API, researchers can write a program to obtain the same information. “Now suppose I want to publish something about the technology of counterfeiting. I could look up and write down the facts that I need, which could then be published online or on paper. But I also have the possibility of writing a program which dynamically queries the OB via its API. If the information in the OB database is updated, my program will automatically get the latest results. Furthermore, I can mash up live information from a number of different sources, and incorporate new sources as they come online.”

Unlike many historians, Turkel learned how to program as a child, empowering him to create and customize the tools he needs. For his work on Old Bailey, he is collaborating with experts on 18th-century British history and using Mathematica to investigate historical questions. As they work, the group’s interpretations evolve, while Mathematica’s notes and Comput-

ACM Member News

VALERIE TAYLOR AND CMD-IT ENCOURAGE MINORITY, DISABLED PARTICIPATION IN IT



As a graduate student at Berkeley pursuing her Ph.D. in electrical engineering and computer

science, Valerie Taylor was often the only woman and sole minority participant in her classes. She felt isolated, she recalls, as her white male fellow students would tend to communicate far more with each other than with her. Taylor eventually decided to work to ensure future minority students do not have to experience that same feeling of isolation.

Now Regents Professor and Royce E. Wisenbaker Professor

of Computer Science and Engineering at Texas A&M, Taylor in 2010 co-founded the non-profit Center for Minorities and People with Disabilities in IT, <http://www.cmd-it.org>, which she now serves as executive director. The organization’s mission, says Taylor, is to “contribute to the national need for an effective workforce in computing and IT” through synergistic activities related to minorities (including African Americans, Native Americans, Hispanics, and Pacific Islanders) and people with disabilities. It also strives to ensure these people are fully engaged in computing and IT, and promotes innovation that “enriches and enhances and enables these communities so that more equitable and sustainable

contributions are possible by all communities,” she says. [For the organization’s mission statement, see: <http://www.cmd-it.org/about.html>]

Funded by the U.S. National Science Foundation, Department of Energy, and industry, CMD-IT holds workshops on professional development in which minority and disabled students at the undergraduate and master’s levels get to hear industry representatives who also belong to minority groups or are people with disabilities talk about topics such as résumé writing, effective interviews, and networking. They also participate in mock interview sessions and receive valuable feedback. As a result, 86% of participants said they felt better prepared for real interviews, and 93% found company

representatives’ feedback very helpful. “That’s good for both parties,” says Taylor.

A longitudinal evaluation of CMD-IT’s Academic Careers Workshops conducted this year showed very positive results, Taylor says. “Participants said having a community was really important, as was seeing others who look like them talk about effective strategies for navigating the academic ladder.” In addition, “The award rate for proposals across all funding agencies and industry was 53%. That’s pretty good.”

Taylor says the longitudinal evaluation also found unexpected altruism resulting from the workshops—in addition to seeking mentorship, participants also mentor others.

—Karen A. Frenkel

able Document Files enable them to keep prose, code, data, and visualizations together in a live document.

"History is still what we make of the past, but our understanding becomes much more dynamic," asserts Turkel.

Computationally driven dynamism also holds pedagogical promise. In addition to making history more accessible through online tools and research collections, it offers an alternative to the traditional output of historical research: narrative. Resources like the University of Virginia's online Valley of the Shadow project enable visitors to construct their own paths through history by exploring letters, diaries, newspapers, and public records from two communities during the Civil War. Created by UVA historians William Thomas and Edward Ayers, the project was conceived as an "applied experiment in digital scholarship," as they wrote in an introductory article.

Thomas now teaches at the University of Nebraska-Lincoln, where he is working on a collaborative project called Railroads and the Making of Modern America, which leverages computation to explore railroad data like cartoons, poetry, maps, timetables, and abandoned track lines. "The Valley of the Shadow project told big history through personal eyes," Thomas says. "We are looking to use big history to get a perspective on the individual." Thanks to a 2009 grant from the National Endowment for the Humanities' Digging into Data challenge, Thomas and colleagues like Ian Cottingham, in UNL's computer science department, were able to visualize the growth of the railroad network by relating data from annual reports to other geospatial data pulled from censuses, newspapers, and the work of other scholars. "The grant allowed us to bring that data into a common visual spatiotemporal representation," Thomas explains. Doing the same for other datasets, however, remains an ongoing challenge. "There is so much textual and visual data—maps, timetables, newspaper advertisements and articles, and individual writings. Annual reports are different across companies and even within companies from year to year."

As pedagogical tools, and even as scholarship, such projects remain ex-

"As is often the case with interdisciplinarity, the best collaborations tend to be represented by individuals who have training both in the humanities and in applied sciences."

perimental. "We know what it means to read a book," says Schmidt. But what does it mean to engage with an online tool or data visualization? "If you tell students to look at a website... well, sometimes they'll just go look at the website without a whole lot of thought." On the other hand, according to Turkel, students often produce highly creative work in the form of blog posts, wiki entries, websites, Twitter feeds, and YouTube videos.

A more pressing challenge is that while interest in computational history continues to grow, few historians actually know how to program. Collaborations like UNL's railroad project offer one solution. Yet as Turkel points out, "something that may be technically uninteresting from a CS perspective may be very interesting when applied to a problem in the humanities, and vice versa. As is often the case with interdisciplinarity, the best collaborations tend to be represented by individuals who have training both in the humanities and in applied sciences."

To help train others in his field, Turkel and a colleague at the University of Western Ontario's history department, Alan MacEachern, created *The Programming Historian*. First conceived as a set of online lessons designed to teach practical computer programming skills to humanists, it has since evolved into a collaborative open access textbook, with peer-reviewed contributions from volunteers. "Though many historians are

not currently able to do the 'technical heavy lifting,' my experience with [*The Programming Historian*] has convinced me that a lot of people aspire to doing that kind of work," says Turkel.

The University of London's Institute of Historical Research (IHR) also recently released a set of free online training courses in semantic markup and text mining. The courses are tailored to historians but also, as IHR project editor Jonathan Blaney explained in an online report, would "be of benefit to any interested humanities scholars." Funded by JISC, a charity that works to promote the use of digital technologies in U.K. higher education, the courses were posted under a creative commons license in the hopes that others will continue to help develop the material.

Schmidt offers the analogy of learning languages. "If you know an obscure language like Syriac, it opens up all sorts of interesting research. I think historians are starting to realize that computational expertise adds an extra layer of depth to their projects." **G**

Further Reading

Blaney, Jonathan et al.
Histore: Historians' Online Research Environments. <http://historeproject.wordpress.com/>

Ching Man Au Yeung and Adam Jatowt:
"Studying How the Past is Remembered: Towards Computational History through Large Scale Text Mining." *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, Glasgow, UK: ACM Press, 2011. <http://www.dl.kuis.kyoto-u.ac.jp/~adam/cikm11a.pdf>

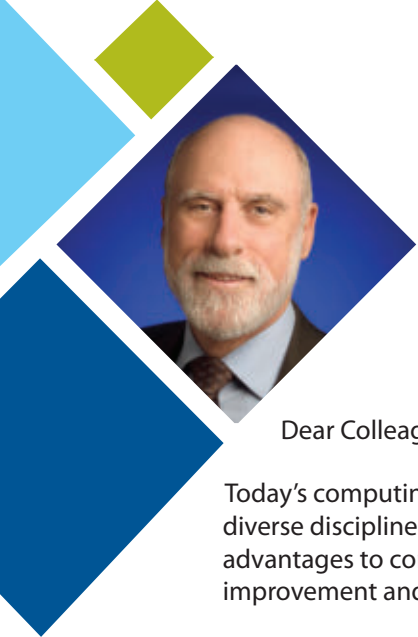
Cook, James et al.
"Your Two Weeks of Fame and Your Grandmother's." WWW 2012, April 16–20, 2012, Lyon, France. <http://www2012.org/proceedings/proceedings/p919.pdf>

Thomas, William G. III and Edward L. Ayers.
The Valley of the Shadow Project. <http://valley.lib.virginia.edu/>

Thomas, William G. III et al.
Railroads and the Making of Modern America. <http://railroads.unl.edu/>

Turkel, William J., Alan MacEachern et al.
The Programming Historian 2. <http://programminghistorian.org/about/project-team>

Leah Hoffmann is a technology writer based in Brooklyn, NY.



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, podcasts and resources for lifelong learning
- Option to join **37 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 multi-faceted 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,

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

All new ACM members will receive an
ACM membership card.

For more information, please visit us at www.acm.org

Professional membership dues include \$40 toward a subscription to *Communications of the ACM*. Student membership dues include \$15 toward a subscription to *XRDS*. Member dues, subscriptions, and optional contributions are tax-deductible under certain circumstances. Please consult with your tax advisor.

RETURN COMPLETED APPLICATION TO:

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

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

Satisfaction Guaranteed!

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 _____



DOI:10.1145/2436256.2436264

Michael A. Cusumano

Technology Strategy and Management

Are the Costs of ‘Free’ Too High in Online Education?

Considering the economic implications as educational institutions expand online learning initiatives.

OVER THE PAST YEAR, there has been a flurry of activity by private entities as well as renowned universities to offer “free” and massive open online courses (MOOCs). The Internet, along with software learning-enhancement tools as well as techniques such as crowdsourcing for grading, are creating a revolution in education. Technology now makes it possible to reach many more students at minimal costs compared to on-campus education. These “distance learning” efforts began years ago, but have gained special prominence recently. There is the success of Kahn Academy (<http://www.kahnacademy.org>), which offers for free nearly 4,000 10-minute lectures on a variety of subjects. Other free online education efforts involving universities include Coursera (<http://www.coursera.org>), a for-profit company formed in 2011 by two Stanford professors. This initiative now has more than 30 partners, including Princeton, Brown, Colum-

bia, Duke, Stanford, the University of Pennsylvania, and Johns Hopkins. As of November 2012, Coursera offered 200 courses and had 1.9 million users. It is supported by \$33 million in venture capital while the partners pay their own operating costs. The management team has yet to announce how it plans to make money and eventually repay investors.⁶

We can expect that free MOOCs will continue to grow and educate millions of students, often in effective and creative ways.

At the Massachusetts Institute of Technology, where I have been teaching since 1986, we have been one of the leaders in free and open online learning. We started with Open Courseware in 2002 (<http://ocw.mit.edu>), which has been supported by the Hewlett and Mellon Foundations, as well as MIT funds, with an annual budget of over \$3 million. It offers a variety of courses, syllabi, and other educational materials, with more than a dozen other universities now as partners. Although the program will soon run out of funding, MIT followed up with MITx and then edX in 2012, a \$60 million joint venture with Harvard.^a The University of Texas, Berkeley, Wellesley College, and others have also joined (<http://www.edx.org>).

We can expect that free MOOCs will continue to grow and educate millions

a P. Cohan, “Will edX put Harvard and MIT out of business?” *Forbes*, May 6, 2012. See also MIT News Office, “What is edX,” May 2, 2012; <http://web.mit.edu/newsoffice/2012/edx-faq-050212.html>



In fall 2012, MIT hosted a symposium examining the influence of technology on new teaching and learning methods in higher education.

of students, often in effective and creative ways. Although dropout rates are about 90%, they may still force universities and colleges to control their costs better and lessen the steep rise in tuition rates that has become an obstacle for many families. So free and open online education should be good for everyone, right? Maybe, but maybe not.

Positive and Negative Network Effects

Initially, “free” and “open” as in Linux and Apache open-source software or Wikipedia as a user-generated encyclopedia came with the assumption that users would participate and contribute to the accumulating knowledge or technology base in a way that was cost-effective and of high quality. As long as there is peer pressure to be a contributor as well as a user, we might even be able to observe positive platform-like “network effects”—the more contributors, the more valuable the knowledge source becomes, which encourages more users and more contributors, ad infinitum.

I worry, however, based on the history of free products and services available on the Internet and their impact on the software products business as well as on the music, video, book publishing, and newspaper and magazine businesses. We have learned that there can also be “negative” network effects. In education, this would occur if increasing numbers of universities and colleges joined the free online education movement and set a new threshold price for the industry—zero—which becomes commonly accepted and difficult to undo. Of course, it is impossible to foresee the future. But we can think about different scenarios, and not all of them are good.

Economists and management researchers see prices as important signals of value. Prices may signal other things, such as status and cost. But if we agree they have something to do with value, then “free” sends a signal to the world that what you are offering costs little and may not be worth paying for. If more or less comparable

products and services are easily available and some are free, users should gravitate toward the free. When universities offer free courses or inexpensive extension school classes as part of their non-profit mission, it is laudable. It is even feasible economically if foundations contribute money to such efforts, which they did for MIT’s Open Courseware (at least initially, and they did not contribute as much as MIT administrators had hoped). Wealthy universities and colleges can subsidize their free or low-fee efforts from other revenue sources—students who pay tuition, donors who add to the endowment, or companies, governments, and foundations that fund research and education.

Some online ventures will probably pursue a middle road and charge for certificates. There will then be more pressure to give course credit for classes completed online. But most non-profit educational institutions have high costs and limited resources, and the quality of potentially free contributions from users will be difficult to

administer. Someone ultimately has to pay for what the online programs give away. Given that there are real costs and quality issues, what are some potential downsides of a free or low-fee college education, especially now that nearly all the elite U.S. institutions have jumped onto the bandwagon?

One real possibility is that universities and colleges that are not so elite will be unable to survive in the new environment. For-profit universities, whose degrees and promises of employment are already being questioned and investigated by the U.S. Congress, will probably be the first institutions to disappear.² That may be a positive consequence for society. I also am not too worried about the survival of MIT, the Ivy League, or comparable schools of very high quality and global reputations, though their ability to charge tuition rates that reflect actual costs may well be threatened, sooner rather than later. I am mostly concerned about second- and third-tier universities and colleges, and community colleges, many of which play critical roles for education and economic development in their local regions and communities.

Paying the Costs of Free

In other industries, we have seen that the real costs of free are absorbed by other parts of the market, with positive as well as negative effects. On the negative side in education, “free” in the long run may actually reduce variety and opportunities for learning as well as lessen our stocks of knowledge. For example, usage of Wikipedia is up, but contributions have been declining steadily over the past several years.⁴ Meanwhile, encyclopedia companies, including the venerable *Encyclopaedia Britannica*, have closed or found it increasingly difficult to sell their traditional products.⁵ Will the world be better off if most encyclopedia companies shut their doors and most people only use Wikipedia? Maybe, but maybe not. We have already seen a major decline in the variety and health of book publishers as well as newspapers and magazines. We lost *Newsweek* in 2012 to bankruptcy and since 2009 have almost lost *The New York Times* more than once except for massive cash infusions by Mexican investor Carlos Slim.¹ Many other newspapers

What are some potential downsides of a free or low-fee education, especially now that nearly all the elite U.S. institutions have jumped onto the bandwagon?

and magazines have failed or had to be bailed out by local and foreign investors with a variety of agendas, and may no longer be the bastions of free speech and press they once were. Web content has replaced a lot of for-fee content, but is the quality and objectivity the same? Again, maybe not.

Free products and services appear over the Internet because the marginal cost of reproducing and delivering a digital good is essentially zero. The marginal cost of adding users to an online class of thousands of students is also close to zero, especially with grading done by computers or free crowdsourcing. But these calculations ignore the expenses associated with research, development, marketing, sales, infrastructure overhead, quality control, and administration. So, yes, digital goods and services such as software products, newspapers, magazines, books, music, video, and even college classes may have close to zero marginal costs and “gross margins” of up to 99%. But if revenues collapse—whether they are software product sales, newspaper subscriptions, or college tuition—then at least some institutions will have another calculation to make: $99\% \text{ of zero} = \text{zero}$.

Companies that survive the onslaught of competition from free alternatives generally have business models and economies of scale and scope that enable them to take advantage of what we call “multi-sided markets.” Their products are really “free, but not free.” They subsidize one side of the market to gain users and make money from

other parts (see “The Evolution of Platform Thinking,” *Communications*, Jan. 2010). For example, Netscape in the 1990s gave away browsers to educational or trial users for free, but sold hundreds of millions of dollars worth of servers and development tools to companies that wanted to set up websites, intranets, and extranets. Then later it sold advertising through its website to companies that wanted to reach users of its browser. Nonetheless, Netscape eventually lost the browser wars after Microsoft started bundling its Internet Explorer browser for free with Windows.³ (Microsoft still gives away Internet Explorer while Windows continues to generate nearly \$20 billion in revenue each year.) Adobe gives away the Acrobat Reader, but every year sells billions of dollars’ worth of other products, such as servers and editing tools. Open source software like Linux is free but the leading distributor, Red Hat, sells more than a billion dollars’ worth of professional services each year (and also pays itself for a lot of Linux development). Google gives away the Android operating system and the Chrome browser for smartphones and tablets, and much other software functionality delivered over its website. But Google is not in the business of selling software products; it primarily sells advertising to companies who want to reach the billions of users of its search tools and other free products and services.

I worry especially because my research going back several years found that about two-thirds of the public software product companies existing in 1998 disappeared by 2006.^b Part of the explanation is the Internet boom, which allowed some fledgling companies to go public, followed by a wave of failures as well as acquisitions led by stronger companies such as Oracle, IBM, Microsoft, Cisco, EMC, SAP, and Adobe. But another reason why many failed or struggled was the increasing prevalence of free or cheap alternatives that were “good enough” and available over the Web. Most software product

^b M. Cusumano. *Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Unpredictable World* (Oxford University Press, 2010), p. 94. Also see M. Cusumano, “The Changing Software Business: Moving from Products to Services,” *IEEE Computer* 41, 1 (2008), 78–85.

companies can never reach a scale big enough to sustain their businesses simply by selling advertising, like Google does. They need to sell services or monetize another side of the market related to the free products (for example, give away the reader but charge for servers, tools, and services). For companies that would like to sell products, “free” as in Google’s software or Microsoft’s bundles can be very destructive, and sometimes fatal, to their business models.

The industries I follow closely are still struggling to recover from the impact of free. *The New York Times* made a mistake when it offered its content for free over the Internet, and is now trying to backtrack and adopt a hybrid model and charge for some usage. This hybrid model is what *The Wall Street Journal* has successfully utilized. Hulu.com, the TV distribution joint venture formed in 2007 and led by NBC, Fox, and Disney-ABC, once gave away all its content for free, with advertising. It has evolved as well to a hybrid model, adding a monthly subscription service with premium content, much like Netflix. The music industry was nearly destroyed by free (and often illegal—remember Napster?) sharing until Steve Jobs found a way to price and distribute songs with Apple’s iTunes service. Music is no longer free, for the most part, and the industry and its creators, the artists and publishers, have survived. Book publishers are still figuring out how to compete with free Web content and new entrants into publishing such as Amazon.

Universities may gain some benefits to their reputations and attract more students and employees or create more scholars by giving away some knowledge for free. But free such as we have seen in software as well as digital music, book publishing, newspapers, magazines, and video, when it works economically, really needs to be more like “free, but not free,” a term we first used with reference to the Netscape browser. The survivors have found indirect ways of covering their costs and generating a surplus.

Do we have more variety and a better world when only a few players survive in an industry? The expansion of free massive open online courses now being offered by elite universities (whose reputations are already high, without

free Web courses) creates the risk that lesser institutions will suffer the fate of many software product companies as well as other producers of digital goods and services. Will two-thirds of the education industry disappear? Maybe not, but maybe! It is hard to believe that we will be better off as a society with only a few remaining mega-wealthy universities. Then there is the other issue of whether online education is truly a desirable substitute for in-class learning and face-to-face interaction. We often say at MIT that the personal networks and bonds our students form while at the university are probably the most valuable part of their education.

Conclusion

Stanford, MIT, Harvard et al. have already opened a kind of Pandora’s box, and there may be no easy way to go back and charge students even a moderately high tuition rate for open online courses. Free learning via the Internet seems here to stay. It is probably most valuable in moderation and as a complement to traditional university education and degrees, not as a substitute. It also will probably force educational institutions to bring down the rising costs of education, as well as the rising prices of tuition. This seems positive but may lead to potentially negative effects and unintended consequences: Elite universities need to ensure the true costs of their MOOCs do not become too high for society as a whole by destroying the economic foundations of less-prominent educational institutions—or of themselves. **■**

References

1. Adams, R. Carlos Slim boosts stake in New York Times again. *The Wall Street Journal* (Oct. 6, 2011).
2. Crotty, J.M. For-profit colleges thrashed in congressional report. *Forbes* (Aug. 2, 2012).
3. Cusumano, M. and Yoffie, D. *Competing on Internet Time: Lessons from Netscape and Its Battle with Microsoft*. Free Press, 1998.
4. Dillow, C. Is Wikipedia in decline? Scientists search for answers in Wikipedia’s numbers. *Fast Company* (Aug. 3, 2009).
5. Eaton, K. Microsoft shutters Encarta as Douglas Adams’ encyclopedia model wins. *Fast Company* (Mar. 31, 2009).
6. Korn, K. More colleges team with for-profit educator. *The Wall Street Journal* (Sept. 19, 2012).

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 Unpredictable World* (Oxford University Press, 2010).

Copyright held by author.

Calendar of Events

April 16–19

International Conference on Multimedia Retrieval, Dallas, TX, Sponsored: SIGMM, Contact: Balakrishnan Prabhakaran, Email: praba@utdallas.edu

April 17–19

Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, Budapest, Hungary, Sponsored: SIGSAC, Contact: Levente Buttyan, Email: buttyan@hit.bme.hu

April 17–19

International Conference on Multimedia Retrieval, Dallas, TX, Sponsored: SIGMM, Contact: Balakrishnan Prabhakaran, Email: praba@utdallas.edu

April 18–20

International Conference on Knowledge Management, Information, and Knowledge Systems, Hammamet, Tunisia, Contact: Ines Saad, Email: ines.saad@supco-amiens.fr

April 21–24

ACM/SPEC International Conference on Performance Engineering, Prague, Czech Republic, Sponsored: SIGMETRICS, SIGSOFT, Contact: Petr Turna, Email: petr.turna@d3s.mff.cuni.cz

April 21–24

International Symposium on Networks-on-Chip, Tempe, AZ, Sponsored: SIGBED, SIGDA, Contact: Karam S. Chatha, Email: kchatha@asu.edu

April 23–26

18th Conference on Animation, Effects, Games, and Transmedia, Stuttgart, Germany, Contact: Thomas Haegeler, Email: thomas.haegeler@filmakademie.de

Emerging Markets

Peacebuilding in a Networked World

Harnessing computing and communication technologies in fragile, conflict-stressed nations.

ONE-QUARTER OF HUMANITY lives in fragile, violent, and conflict-stressed environments. These people's lives are often characterized by the starkest of realities: undernourishment, illiteracy, short life spans, even lack of access to basic sanitation and clean water. Destroyed and diminished infrastructure, weak institutions, and endemic mistrust and suspicion characterize conflict-affected countries. Moreover, these harsh conditions can have global effects as conflict-stressed environments export extremism, regionalize discord, and create worldwide instabilities.

Peacebuilding is the collective processes to end or mitigate conflict, rebuild and reconcile post-conflict, and foster conditions that avoid conflict in the first place. So what is the role and promise of computer and communication technologies in peacebuilding efforts?

Traditional media, such as radio and print journalism, is well established as an essential element in peacebuilding programs. But as access to mobile phones and social media expands—strikingly even in these same conflict-stressed environments—attention is increasingly turning to the transformational promise of such technologies to enhance or replace old media in peacebuilding.

As Shanti Kalathil⁵ and her co-authors have noted, the traditional media formulation of communication tech-



An M-Paisa representative teaches Afghan men how to use the mobile payment system.

nologies and peacebuilding “viewed the receivers of information and ideas sitting passively on the receiving end of a carefully crafted donor message.” But today’s mobile and social media technologies demonstrate how interactive “dialogue is just as important, if not more important” in peacebuilding efforts. As one popular formula puts it—as long as you are talking, you can’t be shooting.

Mobile Phones

Mobile phones are becoming fundamental technologies in enabling this

peacebuilding dialogue. Indeed, mobile phones seem not to only persist in conflict-stressed environments but to flourish. Even the world’s most conflict-affected countries have robust mobile phone networks. Somalia, often topping the global list of failed states, has five mobile phone operators. Daniel Stauffacher⁷ and colleagues have noted how in Somalia “state failure and the accompanying lack of regulatory systems has enabled the creation of one of Africa’s most extensive and resilient cellular telephone systems, accompanied by nu-

merous satellite-based Internet access points” (emphasis added). In some perverse sense, conflict-induced state failure may have actually facilitated mobile phone penetration.

Indeed, through statistical comparison of a country’s cellular penetration against measures of economic, social, and political strength I have found mobile phone penetration seems rather immune to conflict and insecurity.¹ I recently examined data from the Brookings Institute’s Index of State Weakness in the Developing World from 2008 and compared it with 2008 ITU mobile phone penetration figures. The Brookings index ranks states on their economic, political, social, and security strengths. Not surprisingly, a state’s economic strength positively correlates with its mobile phone penetration—richer countries have more phones per capita. Similarly strong positive relationships exist between mobile phone penetration and the Brookings Institute’s indicators for political and social strength. Indeed, all these variables are strongly and positively related to mobile phone penetration. And so it is all the more incredible that the security indicator—which includes factors related to conflict intensities, political stability, coups, gross human rights abuses, and territory affected by conflict—does *not* explain variation in mobile phone penetration levels; these two variables’ relationship is weak and statistically insignificant.

Put simply, mobile phone penetration is sensitive to money, politics, and social development—but seems immune to insecurity and conflict.

What might drive flourishing user adoption of mobile phones in conflict-stressed settings? The West African nation of Liberia is one of the world’s poorest and most conflict-affected countries, having emerged in 2003 from more than 15 years of civil war. With otherwise weak infrastructure, Liberia’s mobile phone sector has been thriving. But why?

Researchers at Georgia Tech surveyed 85 mobile phone users in both the capital city and rural areas, and interviewed industry experts to identify why so many Liberians were using mobile phones.² Directly linked to Liberia’s post-conflict high-threat

What is the role and promise of computer and communication technologies in peacebuilding efforts?

environment, the most common motivation for mobile phone adoption across our informants was the way the phone enhanced their personal security. In a country with deep fissures, high levels of unemployment, and concomitant criminality it is understandable that a mobile phone is seen as providing personal security; a centerpiece requirement to post-conflict peacebuilding. For instance, one phone company manager mentioned that when his company considered removing free calling during late night hours, customers complained. Late at night was when they most needed the ability to make calls without credit on their phone, in case of an emergency situation. Another operator suggested that many users leave their phones on at night for safety rather than switch them off to conserve battery charge.

Even Afghanistan—which suffers from the world’s lowest security index according to the aforementioned Brookings rankings—has a robust mobile phone sector with five operators and more than 13 million subscribers. In this deeply conflict-scarred country, mobile phone applications that move beyond simple voice and text communication are emerging with important peacebuilding implications. M-Paisa (“paisa” means “money” in the local language, Dari) is a mobile banking service that allows users to deposit, withdraw, and transfer money as well as pay for goods and bills with participating vendors. In low-security high-threat areas such as Afghanistan, traveling with large amounts of cash on hand is both particularly common and particularly dangerous. The M-Paisa service responds to

this significant problem by reducing the amount of cash individuals must keep on their person.

Furthermore, according to a report from the United States Institute of Peace,⁴ M-Paisa has also been useful in mitigating corruption endemic to conflict and post-conflict zones: “Perhaps most significant in terms of the conflict has been the use of M-Paisa to distribute salaries to members of the Afghan National Police. Fifty officers in [the pilot program] received 30% larger payments with the program... because salary poaching from their commanding officers and manual transport methods were reduced.”

Print illiteracy is pervasive in many conflict-stressed environments and nowhere is this truer than in Afghanistan. Therefore, one of the important technical enhancements of the M-Paisa service was the development of an interactive voice response (IVR) system supporting Dari, Pashto, and English. M-Paisa is responding to some of the core realities of this conflict-stressed environment: high levels of insecurity and low levels of literacy.

Social Media

Whether accessed on mobile phones, through cybercafés or other means, social media platforms, and methods to monitor them, are also emerging as important technologies for peacebuilding. One critical component in building and sustaining peace are political developments and democratic reforms that set the stage for broader participatory governance. Social media can support these developments up to and including elections. For example, the social media mapping platform Ushahidi has been used to facilitate crowdsourced monitoring and response in multiple elections across the globe. In fact, this technology was created in early 2008 as a response to failed elections in Kenya.

Research at Georgia Tech is taking the possibilities of peacebuilding through social media a step further. Our software system, built specifically for election and conflict monitoring, can aggregate and analyze multiple social media streams including Twitter, Facebook, Ushahidi instances, blogs, and more. The software, called Aggie, handles large volumes of data

performing in real-time keyword extraction and text analysis, incidence identification, and topic visualization.

In April 2011, Nigeria became a perfect test environment for this technology when the West African nation conducted elections for national and state offices. Previous contests in 2003 and 2007 were marked by accusations of electoral fraud and violent protests across the country. Whether or not the 2011 election would bring similar upheaval was an open question. Working with collaborators at Harvard and with a consortium of youth activists in Nigeria called Enough is Enough, we linked our social media monitoring technology to a response situation room based in Nigeria's capital, Abuja. Our software in real time tagged and flagged social media reports of ballot and election irregularities, sending them both as timeline visualizations and triagable event lists to the Abuja situation room. If, for example, our software identified a report of a polling place running out of ballot papers, that report would be sent to the Abuja team who could call the election commission and inform them of the concern.

Tragically, as the initial presidential returns started to come in, rioting broke out in northern Nigeria after an opposition candidate for President claimed election fraud. Thousands of people were being injured and killed. In response, we quickly reconfigured our software tool to also flag reports of violence, which the Abuja situation room would then triage and if indicated call upon the police or military for action. At the height of the unrest we were receiving 50 reports a second, analyzing them in real time, and forwarding tagged visualizations to Abuja.

Routinely, our social media monitoring technology was out front of both national security forces and traditional media in identifying and initiating responses to electoral irregularities and conflict. Ultimately we believe that by using our software system to connect reports expressed through social media directly to real-time response we were able to attenuate elements of election-related violence and help return the country to peace. Our system was engineered to specifically respond to the peace-sustaining and conflict-response needs

Social media platforms, and methods to monitor them, are emerging as important technologies for peacebuilding.

of the Nigerian election by monitoring, for instance, not just general-purpose social media streams such as Twitter but also crises-response platforms such as Ushahidi.

Technologies for Peacebuilding

The promises of computer and communication technologies for peacebuilding are palpable. For example, in a systematic study of users of interactive rich media technologies and post-conflict reconciliation in Liberia, we have discovered how these computer systems can have deep psychological effects important to peacebuilding and national healing.³ These early results notwithstanding, considerable research is required to fully understand the effect of these systems in helping bring and sustain peace in troubled nations.

M-Paisa in Afghanistan and our social media tracking system in Nigeria demonstrate some of the ways that information and communication technologies can be purpose-built to respond to the realities of conflict-stressed environments and the particularities of peacebuilding. Search for Common Ground⁶ has identified a number of the realities that need to inform ongoing research and development in peacebuilding technologies: collaboration and management of information flows; trust and validation; environmental factors; privacy, security and ethical challenges.

A new engineering and research agenda needs to emerge to tackle these pressing issues and answer questions such as:

- How do we manage the privacy and security needs particular to conflict-stressed environments; when is

anonymity required and when does it diminish trust?

- What are new forms of conflict-durable, rapidly deployable, and self-healing network infrastructures?

- How do we innovate new interface modes and methods for use by communities where a generation is unschooled and there is a pervasive low level of print and computer literacy?

- How do we analyze and validate large volumes of information collected from decentralized sources across social media platforms; how to account for the biases inherent in unequal access to these very systems?

- What new methods are needed to monitor and access the impacts of computer and computer technologies in peacebuilding?

While interest in these technologies is flourishing among scholars, policy makers, and international organizations,^a there remain plenty of unanswered questions—and unmet possibilities—for information and communication technologies and peacebuilding. **G**

a For example ongoing programs at The World Bank (<http://www.infodev.org/en/Project.133.html>), the United States Institute of Peace (<http://www.usip.org/issue-areas/communications-and-media>), and the UN's ICT for Peace Foundation (<http://ict4peace.org>).

References

1. Best, M.L. Mobile phones in conflict-stressed environments: Macro, meso and microanalysis. In *Mobile Technologies for Conflict Management: Online Dispute Resolution, Governance, Participation*. M. Poblet, Ed. Springer, London, 2011.
2. Best, M.L., Etherton, J., Smyth, T., and Wornoy, E. Uses of mobile phones in post-conflict Liberia. *Information Technologies and International Development* 6, 2 (2010), 91–108.
3. Best, M.L., Long, W.J., Etherton, J., and Smyth, T. Rich digital media as a tool in post-conflict truth and reconciliation. *Media, War & Conflict* 4, 3 (2011), 231–249.
4. Himelfarb, S. *Can you help me now? Mobile Phones and Peacebuilding in Afghanistan*. United States Institute of Peace, Washington, D.C., 2010.
5. Kalathil, S., Langlois, J., and Kaplan, A. *Towards a New Model: Media and Communication in Post-Conflict and Fragile States*. The World Bank, Washington, D.C., 2008.
6. Search for Common Ground. *Communications and Peacebuilding: Practices, Trends and Challenges*. United States Institute of Peace, Washington, D.C., 2011.
7. Stauffacher, D., Drake, W., Currian, P., and Steinberger, J. *Information and Communication Technology for Peace: The Role of ICT in Preventing, Responding to and Recovering from Conflict*. United Nations ICT Task Force, New York, 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 (<http://mikeb.inta.gatech.edu/>).

Copyright held by author.



Kode Vicious Code Abuse

One programmer's extension is another programmer's abuse.

Dear KV,

During some recent downtime at work, I have been cleaning up a set of libraries: removing dead code, updating documentation blocks, and fixing minor bugs that have been annoying but not critical. This bit of code spelunking has revealed how some of the libraries have been not only used, but also abused. The fact that everyone and their sister use the timing library for just about any event they can think of is not so bad, as it is a library that is meant to call out to code periodically (although some of the events seem as if they do not need to be events at all). It was when I realized that some programmers were using our socket classes to store strings—just because the classes happen to have a bit of variable storage attached, and some of them are globally visible throughout the system—that I nearly lost my composure. We do have string classes that could easily be used, but instead these programmers just abused whatever was at hand. Why?

Abused API

Dear Abused,

One of the ways in which software is not part of the real world is that it is far more malleable—as you have discovered. Although you can use a screw as a nail by driving it with a hammer, you would be hard pressed to use a plate as a fork. Our ability to take software and transmogrify it into shapes that were definitely not intended by the original author is both a blessing and a curse.



Now I know you said you clearly documented the proper use of the API you wrote, but documentation warnings are like yellow caution tape to New York jaywalkers. Unless there is an actual flaming moat between them and where they want to go, they are going to walk there, with barely a pause to duck under the tape.

Give programmers a hook or an API, and you know they are going to abuse it—they are clever folks and have a fairly positive opinion of themselves, deservedly or not. The APIs that get abused the most are the ones that are most general, such as those used to allocate and free memory or objects, and, in particular, APIs that allow for the arbitrary pipelining of data through chunks of code.

Systems that are meant to transform data in a pipeline are simply begging to be abused, because they are so often written in incredibly generic ways that

present themselves to the programmer as a simple set of building blocks. Now, you may say these were written as building blocks for networking code, or terminal I/O, or disk transactions; but no matter what you meant when you wrote them, if they are general enough, and you leave them in a dark place where other coders can find them, then the next time you look at them, they may have been used in ways unrecognizable to you. What's even better is when people abuse your code and then demand that you make it work the way they want. I love that, I really do—no, I don't!

One example is the handling of hardware terminal I/O in various Unix systems. Terminal I/O systems handle the complexities inherent in various hardware terminals. For those too young to have ever used a physical terminal, it was a single-purpose device hooked to a mainframe or minicomputer that al-

lowed you to access the system. It was often just a 12-inch-diagonal screen, with 80 characters by 24 lines, and a keyboard. There was no windowing interface. Terminal programs such as xterm, kterm, and Terminal are simply a software implementation of a hardware terminal, usually patterned on the Digital Equipment Corporation's VT100.

Back when hardware terminals were common, each manufacturer would add its own special—sometimes very special—control sequences that could be used to get at features such as cursor control, inverse video, and other modes that existed on only one specific model. To make some sense of all the chaos wrought by the various terminal vendors, the major variants of Unix such as BSD and System V created terminal-handling subsystems. These subsystems could take raw input from the terminal and by introducing layers of software that understood the vagaries of the terminal implementations, transmogrify the I/O data such that programs could be written generically to, say, move the cursor to the upper left of the screen. The operation would be carried out faithfully on whatever hardware the user happened to be using at that moment.

In the case of System V, though, the same system wound up being used to implement the TCP/IP protocol stack. At first glance this makes some sense, since, after all, networking can easily be understood as a set of modules that take data in, modify it in some way, and then pass it to another layer to be changed again. You wind up with a module for the Ethernet, then one for IP, and then one for TCP, and then you hand the data to the user. The problem is that terminals are slow and networks are fast. The overhead of passing messages between modules is not significant when the data rate is 9,600bps; but when it is 10Mbps or higher, suddenly that overhead matters a great deal. The overhead involved in passing data between modules in this way is one of the reasons that System V STREAMS is little known or used today.

When the time finally came to rip out all these terminal I/O processing frameworks (few, if any, hardware terminals remain in service), the number of things they had been extended to do became fully apparent. There were things that were implemented using the terminal

The complex connections between bits of software and how modules of code can relate to each other escape many people—users and programmers alike.

I/O systems pretty much as a way to get data into and out of the operating-system kernel, completely unrelated to any form of actual terminal connection.

The reason these systems could be so easily abused was that they were written to be easily extended, and one programmer's extension is another programmer's abuse.

KV

Dear KV,

My company has been working for several weeks on upgrading libraries on our hosted systems. The problem is that we have to stop all our users from running on these systems during the upgrade, and this upsets them. It is nearly impossible to explain that the upgrades need to be atomic. In fact, they do not seem to understand the original meaning of *atomic*.

About to Go Nuclear

Dear Nuke,

Ah yes, ask any programmer about “atomic operations,” and if they have any familiarity at all, they will go on about test and set instructions and maybe build you a mutex or a semaphore. Unfortunately, this micro-level understanding of how to protect data structures or sections of code from simultaneous access does not always translate to the macro world where a whole set of operations must be completed in order to get the job done. For some reason the complex connections between bits of software and how modules of code can relate to each other escape many people—users and programmers alike.

You and I both know that an atomic operation is simply some job that must be completed, without interference, in one transaction. An atomic operation is one that simply cannot be broken down any further. The case of updating libraries of code is actually a good example.

Bits of code all have interdependencies. When a library is changed, all the code that depends on that library must change to remain compatible with the library in question. Modern programs link against tens, hundreds, or sometimes thousands of libraries. If the linkage were all one way—that is, the program connected only to each of the libraries—that would be complex enough. In reality, however, many libraries require other libraries, and so on, until the combinatorial explosion makes my head hurt.

To update a single library in the system, you would need to understand which other libraries depended on it, and how their APIs changed, and if those libraries were also up to date. This is the point where everyone throws in the towel and just upgrades everything in sight, making the size of the atomic operation quite large. Perhaps the easiest way to explain this to your users is to make them graph the connections between the various bits of code, as can be done with systems such as Doxygen. Then, while they are off scratching their heads while attempting to graph the connections, you can bring down the system, upgrade it, and restart it, long before they have figured out the graph.

KV

Q Related articles
on queue.acm.org

Closed Source Fights Back

Greg Lehey

<http://queue.acm.org/detail.cfm?id=945126>

Kode Vicious: Code Rototilling

George Neville-Neil

<http://queue.acm.org/detail.cfm?id=2078497>

You Don't Know Jack about Shared Variables or Memory Models

Hans-J Boehm and Sarita V. Adve

<http://queue.acm.org/detail.cfm?id=2088916>

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

Copyright held by author.

Viewpoint

Cyber-victimization and Cybersecurity in China

Seeking insights into cyberattacks associated with China.

AS A RAPIDLY DIGITIZING economy with over 538 million Internet users circa July 2012, China has been an attractive cybercrime target. The China Internet Network Information Center indicated that in the first half of 2011 217 million Chinese (45% of the Internet population) became malware victims, 121 million had online accounts hacked, and 8% were victimized by scammers. Likewise, the Ministry of Public Security's network security protection bureau noted that 80% of computers had been botnet-controlled.¹

In the media and in international forums, Chinese government officials have repeatedly stressed and emphasized the country's victimization by foreign-originated cyberattacks. They are also concerned about the lack of cooperation or interest from Western countries in fighting cybercrimes. Given China's emphasis on and concern with foreign-originated cyberattacks, this Viewpoint assesses the extent of internally originated cybercrimes and China's capability, willingness, and resources to control them and examines the nature of its cybersecurity-related international engagements.

China's Greater Emphasis On and Concern about Foreign Originated Cyberattacks

China's senior government officials have commonly attributed cyberattacks targeting the country to foreigners. For instance, Gu Jian of the Chinese Minis-



try of Public Security said that over 200 Chinese government websites experienced cyberattacks daily and most are foreign originated. According to the State Council's Information Office, over one million Chinese IP addresses were controlled and 42,000 websites were hijacked by foreign hackers in 2009.

In the January 2011 meeting of the Intergovernmental Group of Experts of the UN Crime Prevention and Criminal Justice Program, the Chinese delegation noted that in 2010, over 90% of network sites' servers used to commit frauds such as phishing, pornography,

and Internet gambling against Chinese targets were located outside China. The delegation also stated that over 70% of botnet control sites were in foreign countries. According to China's Computer Emergency Response Team's (CNCERT) Internet Security Perception Report, 8.9 million Chinese computers were attacked by 47,000 foreign IP addresses in 2011 and China was the world's biggest cyber-victim. The report further noted that foreign hackers had attacked 1,116 Chinese websites and accounted for 96% of phishing attacks targeting Chinese banks.⁶

Chinese government officials have also complained about U.S. government agencies' lack of cooperation and interest in fighting cybercrimes. Gu noted that China had received no response to its requests for cooperation from the U.S. on 13 cybercrime cases involving issues such as fake bank websites and child pornography and in other cases it took up to six months to receive replies.¹

Cyberattacks Associated with China: Victimization versus Origination

Data proxies and indicators from a number of sources across a long time period indicate substantial cyberattacks originate in China. First, let us look at foreign and domestic origins of malware infecting Chinese computers. One such indicator concerns the malware infection rate per 1,000 computers (MIR) based on the telemetry data collected by Microsoft from users of its security products opting in. The telemetry data indicated China was among the countries with the lowest infection rates, only behind Japan and Finland. Another measure is Sophos' threat exposure rate (TER), which measures the percentage of PCs experiencing malware attacks. China was the second most malware infected country only behind Chile in the third quarter of 2011 with a TER of 45.⁸

The explanation regarding the differences in the two studies is that they differ in ability to detect Chinese and foreign malware. While TER captures all types of malware, telemetry data only detects globally prevalent malware. A Microsoft report concluded that the low infection rate as detected by telemetry can be attributed to a unique Chinese malware landscape that tends to be dominated by Chinese-language threats not found elsewhere.⁵

It is important to triangulate this evidence with that coming from other sources. In 2005 and 2009 China ranked #2—behind the U.S.—in top countries for originating cyberattacks.² According to the Anti-Phishing Working Group (APWG), 70% the world's maliciously registered domain names were established by the Chinese to attack domestic businesses. In 2011, Chinese perpetrators established 11,192 unique domain names and 3,629 .cc subdomains for such attacks, the ma-

jority of which attacked Chinese companies and 80% targeted Taobao.com. Likewise, according to APWG, China had the world's highest malware infection rate of 54.1% in early 2012.

China's Capability, Willingness, and Resources to Control Cybercrimes

While fighting foreign-originated cybercrimes is an understandably challenging problem, it would be relevant and meaningful to examine China's capability, willingness, and resources to control domestically originated cybercrimes. In order to understand this complex phenomenon, let us first illustrate Brazil's experience. A computer crime bill has been pending in the Brazilian Congress since 2005 that has been unpopular with lawmakers due to a concern that it may facilitate government spying on citizens. While the Chinese government does not face similar constraints such as Brazil, due to unique institutional and economic characteristics, it faces challenges of different types.

Although about 40 governments control their online environments, China's unique approach and perspective to cybersecurity is reflected in its international engagement and domestic politics.³ Despite a broad agreement with the West on cybersecurity, China and its allies (Russia, Tajikistan, and Uzbekistan) diverge in important respects. One such difference is their preference to tackle the broader problem of information security rather than cybersecurity. In 2008, the Shanghai Cooperation Organization (SCO) Agreement in International Information Security expressed concerns about the West's monopolization in information and communications technology (ICT). The SCO economies like

China's senior government officials have commonly attributed cyberattacks targeting the country to foreigners.

to control information that is likely to provoke what they call the three "evils" (terrorism, extremism, separatism). They also consider it important to prevent other nations from disrupting their economic, social, and political stability. In September 2011, the SCO economies submitted a draft International Code of Conduct for Information Security before the 66th UN General Assembly. In the document, they show concerns about threats to domestic stability of free flow of information and highlight the dominant role of the state.

On the domestic front, the Chinese government has emphasized a healthy and harmonious Internet environment. A healthy cyberspace that is "porn-free" and "crime-free" and "harmonious" means it does not threaten to destabilize the state's social and political order. Despite the tremendous difficulties associated with regulating and controlling the Internet, the Chinese government's cyber-control measures have been successful in some senses.⁹ However, it has encountered a host of problems and difficulties while attempting to achieve these goals, as illustrated in the following examples.

First, consider the Green Dam Youth Escort firewall software program launched in 2008. The Chinese government had announced a plan to make it mandatory to have the Green Dam installed on all new PCs in the country. The stated goal of the mandate was to protect children from violent and pornographic content.

The first problem the Green Dam faced was that while addressing one cybersecurity issue, it created side effects that raised another. For instance, while it successfully blocked politically sensitive contents, many believed the software would represent significant risks to users as a single flaw in the Green Dam system would expose the entire Chinese population to cybercriminals.

A second problem stemmed from the fact that it increased PC manufacturers' costs, which led to an additional financial burden on consumers. While the Green Dam would be free to users, manufacturers needed to pay license fees to the Ministry of Industry and Information Technology (MIIT) to install the software. The vendor of the Green Dam, Beijing Dazheng Language and Knowledge Processing Research Cen-

ter (BDLKPRC) had received \$6 million from the MIIT to develop the software.

A third related problem had to do with strong opposition from computer manufacturers and the public. Even Lenovo, which is 57% government-owned, opposed it. Internet users, who are increasingly acting on a bottom-up approach, participated in collective resistance efforts to abort the Green Dam.

Given the national security and economic risks and a strong resistance, the Green Dam program was indefinitely delayed after being installed in 20 million PCs. The unsustainable business model led to the closure of BDLKPRC in the 2010 and the company was near bankruptcy.

As another example, consider the 2011 regulation that required microbloggers to register using their real names. The Nasdaq-listed Chinese online media company Sina warned that the requirement would negatively affect user activity and threaten its popular microblogging service, Sina Weibo. Even well after the March 16, 2012 deadline, Sina Weibo continued to allow users, who had not registered their real names to post and use its services.

Regarding the private sector's role, India would provide a particularly appropriate country for comparison. The active and influential roles of the National Association of Software and Service Companies (NASSCOM) have strengthened India's cybersecurity orientation. The Internet Society of China (ISC), a counterpart of the NASSCOM, has engaged in roles of different nature. In 2001, the ISC asked Internet companies to sign a voluntary pledge that required them not to disseminate information "that might threaten state security or social stability." In 2009, the ISC awarded China's biggest search engine company, Baidu, and 19 other companies the "China Internet Self-Discipline Award" for fostering and supporting "harmonious and healthy Internet development."⁴

Conclusion

As most economies, while China undoubtedly has suffered from foreign-originated cyberattacks, data triangulation from multiple sources indicates domestically originated attacks are no less severe. China, like many parts of the world, has a large number of hack-

China's unique approach and perspective to cybersecurity is reflected in its international engagement and domestic politics.

ers with diverse motivations, backgrounds, skills, and interests to engage in cyberattacks.

The base of regime legitimacy in China has shifted from Marx-Leninism to economic growth and prosperity.¹⁰ China thus would like to achieve the goal of its cyberspace governance initiatives without jeopardizing its economic development. In this regard, the government's cost/benefit calculus associated with cybercontrol measures may change over time. For instance, if the perceived risks of state insecurity or social instability increase with microblogging activities, the government may demand stricter enforcement.

Allegations and counter-allegations, which have been persistent themes in dialogues and discourses in the U.S.-China relationships in cybersecurity, can be linked to the lack of an extensive cooperation. For instance, if one country needs the help of the other country in investigating a cybercrime, a request for assistance takes place through an exchange of letters. It was reported that in 2010, the FBI office in Beijing forwarded 10 letters through the Ministry of Foreign Affairs and received responses to two. This is in sharp contrast to the deeper and stronger collaborations and partnerships between the U.S. and European Union (EU) countries. For instance, the Italy-based European Electronic Crimes Task Force, which has dedicated personnel from the countries involved to investigate and prosecute cybercrimes, provides a forum for law enforcement agencies, the private sector, and academia from the U.S. and EU nations.

The resource constraint has necessitated a partial reliance upon the private sector and semi-private institutions to enforce cybersecurity regulations and policies. In order to minimize investment risk or the risk of losing customers, some private sector enterprises have chosen not to enforce the regulations and policies. The largely unsuccessful experiences with the implementations of the Green Dam and real name registration in microblogging suggest a decline in the state's institutional capacity to regulate cyberspace.

Responses of technology companies such as Sina indicate some degree of noncompliance with regulations. This is a significant deviation from the past practices of Chinese companies. Conformance to the existing institutions has been at the expense of technical and functional efficiency, which has acted as a force of institutional changes. This type of contradiction can be described as "legitimacy that undermines functional inefficiency."⁷ Sina has tilted the balance toward efficiency and productivity at the expense of political legitimacy. **C**

References

1. *China Daily*. 2010 Internet policing hinges on transnational cybercrime. (Nov. 10, 2010); http://www.china.org.cn/business/2010-11/10/content_21310523.htm.
2. Kim, S.H., Wang, Q., and Ullrich, J.B. A comparative study of cyberattacks. *Commun. ACM* 55, 3 (Mar. 2012), 66–73.
3. Kshetri, N. Les activités d'espionnage électronique et de contrôle d'Internet à l'ère de l'infonuagique: Le cas de la Chine. *Télescope* 18, 1–2 (2012), 169–187.
4. MacKinnon, R. Inside China's censorship machine (Jan. 29, 2012); <http://fullcomment.nationalpost.com/2012/01/29/rebecca-mackinnon-inside-chinas-censorship-machine/>.
5. Microsoft. Microsoft Security Intelligence Report, 2011; http://www.microsoft.com/security/sir/keyfindings/default.aspx#section_4_1_d.
6. Pauli, D. China is the "world's biggest cybercrime victim." (Mar. 22, 2012); <http://www.scmagazine.com.au/News/294653.china-is-the-worlds-biggest-cybercrime-victim.aspx>.
7. Seo, M.G. and Creed, W.E.D. Institutional contradictions, praxis, and institutional change: A dialectical perspective. *Academy of Management Review* 27, 2 (2002), 222–247.
8. sophos.com. Security Threat Report, 2012; <http://www.sophos.com/medialibrary/PDFs/other/SophosSecurityThreatReport2012.ashx>.
9. Wu, G. In the name of good governance: E-government, Internet pornography and political censorship in China. In *China's Information and Communications Technology Revolution: Social Changes and State Responses*. Zhang, X. and Yongnian Zheng, Y., Eds. (2009), 69–83.
10. Zhao, S. Chinese nationalism and its international orientations. *Political Science Quarterly* 115, 1 (2000), 1–33.

Nir Kshetri (nbkshetri@uncg.edu) is an associate professor in the Bryan School of Business and Economics at the University of North Carolina, Greensboro.

Copyright held by author.

Viewpoint

The Problem with Hands-Free Dashboard Cellphones

Lawmakers misunderstand user experience of technology interface.

A 16-SECOND TELEVISION advertisement for the Hyundai Veloster features the car pulling into the screen, stopping, and sitting motionless in the middle of the road, the driver apparently talking to himself. Two police officers approach, each on a motorcycle, and they stop on either side of the vehicle. After a moment, the two officers pull away without incident and an announcer explains, “There’s lots of reasons to love Veloster’s voice text messaging. Here’s two.” The point of the short television ad is to promote the hands-free text messaging system built into the Veloster’s dashboard, a novel feature available in many new car models.

The claim implicit in these developments—and made almost directly in the Hyundai Veloster commercial—is: Even though texting with a handheld phone is understood to be so dangerous that it is increasingly outlawed across the globe, hands-free texting while driving is conversely so safe that it should be actively encouraged. In this Viewpoint, I challenge this line of thinking.

The problem with the assumption that hands-free phones should not be distracting to drivers is that a multitude of studies have demonstrated otherwise. A serious problem is now emerging as the automotive industry increasingly builds hands-free calling, texting, and even Facebooking into the dashboards of new cars.

**The Science and Policy Context**

Multiple countries across the globe have enacted laws against talking and texting on a handheld phone while behind the wheel (see http://www.cellular-news.com/car_bans). In the U.S., 39 states outlaw texting on a handheld phone while driving, and 10 states maintain laws against driving while talking on a handheld phone (<http://www.distraction.gov/content/get-the-facts/state-laws.html>). But only a very small minority of countries bans hands-free texting or hands-free phone conversation, and no states in the U.S. ban hands-free phone use for all drivers. The implied understanding in such laws—insofar as only handheld and

not hands-free devices are banned—is that it must be the visual and manual interface with the device that causes the driver distraction. However, the preponderance of scientific evidence reveals both handheld and hands-free phone usage to be associated with the same precipitous drop in driving performance.^a These findings point to a different understanding of cellphone-related driving impairment than what is implied by the existing traffic laws: it

^a For a large-scale meta-analysis of these studies, see A.T. McCart, L.A. Hellinga, and K.A. Bratiman, “Cell phones and driving: Review of research.” *Traffic Injury Preventions* 7, 2 (2006), 89–106.

is the conversation that is the source of the distraction. That is, to explain why the large preponderance of evidence shows both handheld and hands-free phones to impair drivers to the same degree, the answer must lie in the mental distraction that comes with talking and listening to someone on the phone.

Just how dangerous is talking on the phone while driving? Research on phone records and accident data indicates a fourfold increase in crash risk for drivers using a handheld or hands-free phone.⁵ Cellphone-induced driving impairment has even been compared to drunk driving.¹⁰ And it is not the case that simply any conversation causes this distraction; evidence is emerging that passenger conversations do not result in the same driving impairment as phone usage. This is because passengers appear to be more aware of driving conditions than are interlocutors on the other end of the phone. Passengers thus modulate conversation when driving conditions change, and even participate in the task of monitoring the road. As Frank A. Drews and his colleagues explain, this difference in driving performance “stems in large part from the changes in the difference in the structure of cellphone and passenger conversation and the degree to which the conversing dyad shares attention.”²

The disagreement between the science and the policy over the issue of hands-free phones is exemplified by recent discord between the U.S. government bodies tasked with addressing threats to traffic safety. In December 2011, the National Transportation Safety Board (NTSB) released an official, though in no way binding, recommendation for a ban on all cellphone usage while driving—including hands-free devices.⁶ The response has been mixed. For example, while Secretary of Transportation Ray LaHood has made combating distracted driving a central project of his tenure, he has distanced himself from the NTSB’s recommendation, claiming “The problem is not hands-free... That is not the big problem in America.” He adds, “Anybody that wants to join the chorus against distracted driving, welcome aboard. If other people want to work on hands-free, so be it.”⁹

On the one hand, I sympathize with the apparent pragmatism of LaHood’s

Cellphone-induced driver distraction results from a human’s inherent cognitive limitations.

strategy. Undeniable legislative progress has been made by aggressively addressing the distraction of handheld phones, especially on the issue of texting while driving. And this progress has come largely without raising the ire of a public and a business community that can be resistant to the regulation of hands-free devices. On the other hand, there are problems with this strategy.

First, when only handheld—and not hands-free—devices are subject to regulation, a message is inadvertently sent that hands-free phone usage while driving is safe. But as I noted earlier, the research shows this to be untrue. Second, the gap between the policy and the science is quickly being filled by new hands-free modes of cellular and Internet communication while driving.

The Source of Cellphone-Induced Driving Impairment

Describing the mental distraction of cellphone usage is tricky, especially in the context of a larger discussion in which some participants only acknowledge the visual distraction caused by looking away from the road and the manual distraction of taking a hand off the steering wheel. There are two general ways in which the mental distraction of cellphones has been conceived: inherent cognitive limitations, and long-developed habits of perception.

While researchers agree that cellphone usage results in dangerous driver distraction, there is no explicit consensus explanation of driver distraction in the empirical literature. Still, it is possible to abstract a general theory from the terminology through which these data are often cast: cellphone-induced driver distraction results from a human’s inherent cogni-

tive limitations. That is, using a phone and driving a car are understood to be two different tasks, each requiring some of our brain’s limited stock of cognitive resources. In this view, the explanation for why cellphone usage results in driving impairment is because the brain does not possess the resources necessary to safely perform these two cognitively demanding tasks at the same time. For example, Tova Rosenbloom summarizes the findings in this way: “the results are in line with the theory of inherent limited capacity of human attention...which predicts that the attentional resources allocated to one task (talking) come at the expense of the other (driving).”⁸

In my own work, I have suggested an alternative reading of the same data.⁷ Building on a philosophical tradition called phenomenology, which specializes in the deep description of human experience, I have developed an account of what it is like to use the phone and also what it is like to sit in the driver’s seat and operate a vehicle. My contention is that users maintain strong habitual relationships with these technologies. For example, responsible driving requires a driver to have an almost automatic relationship with the car; if a driver must stop the car suddenly, she or he must stomp on the brake pedal at the moment the decision is made. A driver cannot first make the decision to brake, then recall that braking is something that involves pressing a pedal, and then press the brake. Safe driving demands that, through training, the driver has developed responses so automatic that she or he can instead actively focus on the road, on the mirrors, on the movements of other cars, on signs and lights, and such. The habits of the phone lead a user to focus on different things. The phone inclines a user to direct attention to the content of the conversation and to the presence of the person on the other end of the phone. The inclination is to become engrossed in conversation and to have the discussion stand forward within one’s overall awareness. My suggestion is that the phone inclines a driver to become absorbed by the phone conversation through a pull much like that of a bad habit. Even if a driver intends to stay

focused on the road while talking on the phone, the long-developed habits of the phone may slither in and draw attention toward the conversation.

Whether you prefer the cognitive scientists' explanation that inherent cognitive limitations are to blame for cellphone-related driving impairment, or the alternative suggestion that impairment results from long-developed habitual inclinations to get absorbed by phone conversation, the implications are the same: despite a driver's intentions to drive safely, a dangerous level of distraction is caused by the phone conversation itself—not by the manual or visual interface with the device.

Hands-Free Dashboard Technologies

The new developments enabling hands-free cellular communication while driving come in two forms. The first are newly emerging voice interface smartphone applications. These are programs that enable users—including drivers—to operate a number of a smartphone's functions through voice command. These include placing a call, dictating text messages, and having incoming text messages read aloud by the computer. The most influential of these is the iPhone's Siri application that offers a discussion-style interface with many of the smartphone's features.

The second form of hands-free communication available to drivers is cellular phone and Internet systems built into a car's dashboard. These new features enable drivers to call and text through voice command. Additionally, the devices may be engaged through buttons and scrolling thumbwheels affixed to the steering wheel or dashboard console, and information may be displayed on screens incorporated into the dashboard. One example is Ford's Sync system, which enables drivers to place hands-free calls, listen to text messages translated into an audio format, and even to reply to texts by sending one of a number of preset responses, such as "Stuck in traffic," and "Can you give me a call?" In an effort to compete with this and other companies offering similar features, General Motors is working to modify its OnStar system to facilitate calls

Despite a driver's intentions to drive safely, a dangerous level of distraction is caused by the phone conversation itself.

and texting, and also to provide drivers a hands-free method for reading and entering posts on Facebook.³

In its 2012 guide to new cars, *Consumers Digest* begins its review of these new dashboard technologies with a brief mention of the NTSB recommendation for the nationwide ban on all in-cab electronics—a ban which, if enacted, would place prohibitions on many of the new technologies the article is about to celebrate. With regard to the ban, the author surmises that, "in any case, we expect that automakers and phone companies will reject the idea as unworkable."⁴ This seems like an understatement. Carroll Lachnit, an editor at Edmunds.com, makes a sharper observation, "It's a little bit of an arms race... There is a sense among carmakers that if they don't start presenting these kinds of vehicle systems, they will be left in the dust."¹

With the development of dashboard-integrated cellular, Internet, and dictation technologies as an exploding area of innovation in the automotive industry, challenges and opportunities are afforded to engineers and computer scientists. But how should these opportunities be pursued? In light of the scientific findings on cellphone-induced driving impairment, practitioners of computer science and engineering ought to develop creative ways to mitigate the dangers of these technologies as they advance. These projects could include, for example, devising more sophisticated options for drivers to preprogram different automated responses tailored to different potential incoming calls, or crafting ways to alert callers that the person on the other end of the phone conversation is behind the wheel.^b

Conclusion

Despite the danger science has shown hands-free devices to pose to drivers, the integration of these technologies into dashboards has become a key area of competition for the automotive industry. The Hyundai advertisement mentioned at the beginning of this Viewpoint is just one example of the way hands-free in-cab devices have become the centerpieces of marketing campaigns. And with the failure of the law to move on this issue, responsibility for the safety of pedestrians and the roadways is left exclusively in the hands of drivers.

Using a hands-free cellphone while driving is still legal in most countries, and it is easier than ever as hands-free devices are added to dashboards. This implies it is safe to use a hands-free phone while driving, and encourages you to do it. But it's not, and you shouldn't. 

^b See, for example J. Lindqvist and J. Hong. Undistracted driving: A mobile phone that doesn't distract. In *Proceedings of HotMobile 2011: 12th Workshop on Mobile Computing Systems and Applications* (Phoenix, AZ, Mar. 1–2, 2011).

References

1. Boudreau, J. Coming soon to freeways: Drivers tweeting at 70 miles per hour. *Mercury News*. (e-edition, updated 2/21/12); www.mercurynews.com/business/ci_19981113?IADID
2. Drews, F.A., Pasupathi, M. and Strayer, D.L. Passenger and cell phone conversations in simulated driving. *Journal of Experimental Psychology: Applied* 14 (2008), 392–400.
3. Halsey III, A. Cars to read text messages out loud. *Washington Post*. (Oct. 25, 2011), A02.
4. McCormick, J. New dimensions in auto technology. *Consumers Digest*. Special Edition: New Car Guide. (Feb. 5–8, 2012).
5. McEvoy, S.P. et al. Role of mobile phones in motor vehicle crashes resulting in hospital attendance: A case-crossover study. *BMJ* 331 (2005), 428–432.
6. National Transportation Safety Board. No call, no text, no update behind the wheel: NTSB calls for nationwide ban on PEDs while driving. Press Release, Dec. 12, 2011; www.nts.gov/news/2011/111213.html.
7. Rosenberger, R. Embodied technology and the dangers of using the phone while driving. *Phenomenology & the Cognitive Sciences* 11 (2012), 79–94.
8. Rosenbloom, T. Driving performance while using cell phones: An observational study. *Journal of Safety Research* 37 (2006), 207–212.
9. Shephardson, D. LaHood won't back NTSB push to ban hands-free calls. *The Detroit News* (Dec. 21, 2011).
10. Strayer, D.L., Drews, F.A., and Crouch, D.J. A comparison of the cell phone driver and the drunk driver. *Human Factors* 48, 2 (2006), 381–391.

Robert Rosenberger (robert.rosenberger@pubpolicy.gatech.edu) is an assistant professor of philosophy in the School of Public Policy at the Georgia Institute of Technology.

Copyright held by author.

ACM Inroads

The magazine for computing educators worldwide



<http://inroads.acm.org>

Paving the way toward excellence in computing education

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

Building websites that perform well on mobile devices remains a challenge.

BY NICHOLAS C. ZAKAS

The Evolution of Web Development for Mobile Devices

THE BIGGEST CHANGE in Web development over the past few years has been the remarkable rise of mobile computing. Mobile phones used to be extremely limited devices that were best used for making phone calls and sending short text messages. Today's mobile phones are more powerful than the computers that took Apollo 11 to the moon,¹⁰ with the ability to send data to and from nearly anywhere. Combine that with 3G and 4G networks for the data transfer, and now using the Internet while on the go is faster than my first Internet connection, which featured AOL and a 14Kbps dial-up modem.

Yet despite these powerful advances in mobile computing, the experience of Web browsing on a

mobile device is often frustrating. The iPhone opened up the “real” Internet to smartphone users. This was important because developers no longer had to write mobile-specific interfaces in custom languages such as Wireless Application Protocol (WAP). Instead, all existing websites and applications worked perfectly on the iPhone. At least that was the idea.

With the fast iPhone and a 3G connection, one would expect a mobile Internet experience to be pretty snappy. However, the Web had developed during a period when the bandwidth available to desktops increased each year. That meant websites and applications started to get larger, using more resources such as Cascading Style Sheets





(CSS), JavaScript, images, and video. All of this was to provide a better experience on the only Internet that many people had: a wired connection going into the home.

By using mobile devices to access that same Internet, however, users once again experienced a slower Web. Although cellular connections have continued to improve over the years, they are still nowhere near as fast as wired connections. Further, although today's smartphones are quite powerful, they still pale in comparison with the average desktop computer. Therefore, making the Internet fast for mobile devices is a strange problem. On the one hand, it is a lot like Web development in 1996 when everyone had

slow connections. On the other hand, mobile devices today are much more powerful than computers were in 1996.

The Latency Problem

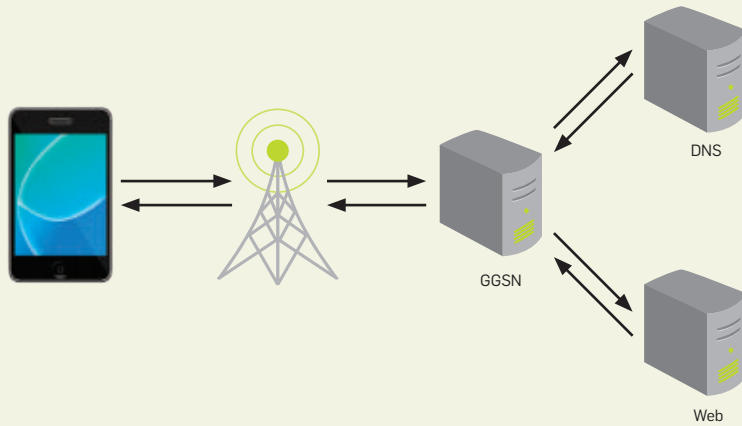
One of the biggest issues for mobile Web performance is *latency*—the delay experienced between request and response. Any given Internet connection is capable of transferring a certain amount of data within a specified amount of time, which is called bandwidth. Latency is what prevents users from receiving that optimal bandwidth even though their connections are theoretically capable of handling it.

Wired latency. Every Internet connection has some sort of latency associated with it. Wired connections have

much lower latency because there is less to get in the way of the requested data. Wired connections allow data to travel directly between points, so it is received fairly quickly. The biggest cause of latency here is the electrical resistance of the wire material. That is usually negligible unless the wire has been damaged. Otherwise, the latency of a wired connection remains fairly stable over time.

When the latency of a wired network changes unexpectedly, the source could be network congestion. If you have ever arrived home in the evening and found your Internet connection to be slower than it was in the morning, it is probably because everyone in your neighborhood is hopping on the Inter-

Figure 1. An HTTP request from a smartphone.



net at the same time. It could also be that several people in your household are on the Internet at the same time using a lot of bandwidth (streaming Netflix, surfing the Web, or using FaceTime). Network congestion is always a consideration when latency is high regardless of the network type.

Wireless latency. Wireless Internet connections are quite different from their wired counterparts. Whether the connection is 3G, 4G, or Wi-Fi, sending

and receiving data through the air introduces a variable amount of latency. The air itself not only causes resistance, but also provides an open space for other sources of interference. Radios, microwaves, walls, and any number of other physical or electromagnetic barriers can adversely impact the effective bandwidth.

Tom Hughes-Croucher ran an experiment to determine the degree to which latency affects the throughput of a connection.² By introducing just 50 ms of latency, he found that the number of requests that could be completed in 300 seconds was cut by nearly 67%. At 300 ms of latency, the number of requests was decreased by almost 90%. What did he use to affect the latency in his experiment? A simple mi-

crowave oven. Now imagine all of the interference produced by the electronics that surround you every day.

The number of requests completed is very important because a typical Web page makes dozens of requests while loading. Visiting a Web site for the first time triggers two requests in sequence right away. The first is a DNS (Domain Name System) request to look up the domain name the user entered. The response to that request contains the IP address for the domain. Then an HTTP request is sent to that IP address to get the HTML for the page. Of course, the page will typically instruct the browser to download more resources, which means more DNS requests and HTTP requests before the page is fully usable. That can happen fairly quickly with a wired connection, but a wireless connection such as that on a smartphone introduces a lot of latency.

The request first has to go from the phone to the nearest cellular tower. That request travels through the air where it is subject to a large degree of interference. Once arriving at the cell tower, the request is routed to a mobile company server that uses a GPRS (General Packet Radio Service). For 3G, this is a GGSN (Gateway GPRS Support Node) that acts as an intermediary between the user and the Internet (see Figure 1). The GGSN assigns IP addresses, filters packets, and generally acts as a gateway to the real Internet. The GGSN then sends the request to the appropriate location (DNS, HTTP, or other), and the response has to come all the way back from the Internet to the GGSN to the cell tower and finally to the phone. All of that back and forth creates a lot of latency in the system.

Making matters worse, mobile networks have only a small number of GGSNs; thus, a user's proximity to a GGSN has a measurable impact on the latency he or she experiences. For example, developer Israel Nir noted that making a request via a mobile phone from Las Vegas to a resource also located in Las Vegas actually results in the request being routed to California first before finally arriving back at the device.⁹ Because GGSNs tend to be centrally located instead of distributed, this is very common.

Latency is always going to be a factor for wireless communications, so devel-

Figure 2. A button generated in CSS.



Figure 3. CSS code.

```

.button {
  border-top: 1px solid #96d1f8;
  padding: 20px 40px;
  color: white;
  font-size: 24px;
  font-family: Georgia, serif;
  text-decoration: none;
  vertical-align: middle;

  /* create a gradient for the background */
  background: linear-gradient(top, #3e779d, #65a9d7);

  /* round those corners */
  border-radius: 40px;

  /* drop shadow around the whole thing */
  box-shadow: rgba(0,0,0,1) 0 1px 0;

  /* drop shadow just for the text */
  text-shadow: rgba(0,0,0,.4) 0 1px 0;
}
  
```

opers need to plan for it when working on mobile projects. The best way to combat latency is to use as few HTTP requests as possible for a website or application. The overhead of creating a new request on a high-latency connection is quite high, so the fewer requests made to the Internet, the faster a page will load. Fortunately, today many more tools are available for reducing requests than in 1996 when the entire Internet was slow.

Improving Web Performance

In *High Performance Web Sites*, published in 2007, Steve Souders wrote the first exhaustive reference about Web performance.¹¹ Many of the best practices in the industry can be traced back to this important book. Although the book was released before mobile Web development existed in its current form, a great deal of advice still applies.

Reducing HTTP requests. The first rule in *High Performance Web Sites* is to reduce HTTP requests. This can be done by concatenating external JavaScript and CSS files. Many sites include hundreds of kilobytes of JavaScript and CSS to create richer experiences. Whenever possible, multiple files on the server should be combined into a single file downloaded to the browser. The ideal setup is to have no more than two references to external JavaScript files and two references to external CSS files per page load (additional resources can be downloaded after page load is completed).

Traditionally, concatenation processes occurred at build time. These days, it is more common for concatenation to happen at runtime using a CDN (content delivery network). Google even released an Apache module called `mod_concat`³ that makes it easy to concatenate files dynamically at runtime. The module works by using a special URL format to download multiple files using a single request. For example, suppose you want to include the following files in your page:

```
http://www.example.com/assets/js/main.js
http://www.example.com/assets/js/
utils.js
http://www.example.com/assets/js/
lang.js
```



The best way to combat latency is to use as few HTTP requests as possible for a website or application. The overhead of creating a new request on a high-latency connection is quite high, so the fewer requests made to the Internet, the faster a page will load.



Instead of referencing each of these files separately, `mod_concat` allows them to be combined into one request using the following URL:

```
http://www.example.com/assets/
js??main.js,utils.js,lang.js
```

This URL concatenates `main.js`, `utils.js`, and `lang.js` into a single response in the order specified. Note the double question marks, which indicate to the server that this URL should use the concatenation behavior. Setting up `mod_concat` on a server and then using the server as an origin behind a CDN provides better edge caching for the resulting file.

Eliminate images. Images are one of the largest Web components on the Internet. According to the HTTP Archive (which monitors performance characteristics of the top million sites on the Internet), images account for an average of 793KB per page (as of January 2013).¹ The next closest component is JavaScript at 207KB. Clearly, the fastest way to reduce the total size of the page is to reduce the number of images being used.

CSS3, the latest version of CSS, provides numerous ways to eliminate images. Many visual effects that previously required images can now be done declaratively directly in CSS. For example, creating a button that has rounded corners, a drop shadow, and a gradient background once required several images, as well as a graphic designer to create them, but today just a few lines of CSS can achieve the same results.

The button pictured in Figure 2 is generated using the CSS and a regular `<button>` element shown in Figure 3.

The key parts of the CSS that replace what would have been images are:

- * `background: linear-gradient (top, #3e779d, #65a9d7)`. This creates a CSS gradient for the background. The most recent versions of all major browsers no longer require a vendor prefix. This line says to create a linear gradient starting from the top beginning with the color #3e779d and ending with the color #65a9d7.

- * `border-radius: 40px`. This rounds the corners of the button to have a radius of 40 pixels. The unprefixed version is supported in the most

recent version of all major browsers.

* `box-shadow: rgba(0,0,0,1) 0 1px 0`. This creates a drop shadow around the entire button. A box shadow⁴ can be used in a variety of ways, but in this example, it is used as a one-pixel offset at the bottom of the button. The numbers after the color are the x-offset, y-offset, and blur radius.

* `text-shadow: rgba(0,0,0,.4) 0 1px 0`. This creates a drop shadow that applies to just the text. A text shadow⁵ has the same syntax as a box shadow.

Thus, just four lines of CSS code can replace multiple images that might have been needed for this button. Additionally, creating this effect requires many fewer bytes than would be necessary using images. Replacing images with CSS is a good idea whenever possible. It reduces the number of HTTP requests and minimizes the total number of bytes necessary for the visual design.

Avoid redirects. Rule 11 in *High Performance Web Sites* is to avoid redirects. A redirect works similarly to call forwarding on a phone. Instead of returning actual content, the server returns a response with a `Location` header indicating the URL the browser should contact to get the content it was expecting. This can go on for quite a long time as one redirect leads to another. Every redirect brings with it the overhead of a full request and all of its latency. On a desktop, the consequence may not be immediately apparent, but on a mobile device a redirect can be painfully slow.

Many websites and applications adopted the convention of using `www.example.com` for their desktop sites and `m.example.com` for their mobile sites. Their mistaken assumption was that users would enter the full domain name for the site based on the version they wanted. In reality, people tend to type in just the hostname, such as `example.com`, meaning that the server needs to figure out what to do with that request. Frequently, the first step is to redirect to the `www` version of the domain, which is the server that is running the Web application. Then the application looks at the user agent string and determines that the device is a mobile device, prompting a second redirect to the `m` version of the domain.



Avoiding redirects and being able to serve the entire experience from the domain that received the request is an absolute performance victory for a site's users.



Bing does this very thing—with some terrible results.

The screenshot from the Web Inspector window in Figure 4 shows two redirects: the first is from `bing.com` to `www.bing.com`; the second is from `www.bing.com` to `m.bing.com`. The latency values in the Web Inspector refer to the time when the browser is waiting to receive a response. Note that each redirect still has latency associated with it, so the actual page does not begin to download until 1,448 ms after the first request was made. That is a whole second and a half of added time to get the user experience up and running without actually doing anything.

Avoiding redirects is absolutely vital in mobile Web development. A redirect has all the overhead of any HTTP request without actually returning any useful information. That is why Web applications are starting to serve both the mobile and desktop versions from the same domain based purely on the user agent string of the request. Whether a domain begins with `www` or `m` or anything else should not matter; avoiding redirects and being able to serve the entire experience from the domain that received the request is an absolute performance victory for a site's users.

Mobile Device Limitations

Until fairly recently, Web developers did not have to worry too much about the device that people were using to access their application. Developers could assume that if a computer was capable of running a Web browser, then it was probably capable of accessing their applications. However, mobile devices are very different. They all have different performance characteristics, but they have one thing in common: they are not as capable as desktops or laptops. Because of that, developers must consider not just who is accessing the application but what device they are using to do it.

Slow and expensive JavaScript. Even though mobile device browsers are pretty good, the performance of their JavaScript engines is an order of magnitude slower than what is on desktop computers. Adding to the problem—at least in iOS—is that someone may visit an application using Safari or an

embedded WebView in another application. While Safari has a reasonably fast JavaScript engine, the embedded WebView does not. So with the result is two different JavaScript performance characteristics in iOS, depending on whether or not the user is using Safari. The graph in Figure 5 shows the SunSpider benchmark results for several popular browsers.¹²

Notice that the performance of embedded WebViews in iOS is actually worse than that of Internet Explorer 8. Even for the better-performing browsers, however, there is still a vast difference between JavaScript engine performance on the desktop and on a mobile device.

Another aspect of JavaScript on mobile devices is the associated performance cost. Unlike desktop computers, mobile devices have batteries that can get drained by radios (cellular, Wi-Fi, Bluetooth), network access, and executing code such as JavaScript. Any time code is executed, the CPU uses power; therefore, more time spent executing code means more power used. Running JavaScript drains batteries more quickly.

These aspects of JavaScript on mobile devices mean developers need to be careful about JavaScript usage. As much as possible, it is best to avoid using JavaScript. For example, using CSS animations⁶ or CSS transitions⁸ to create animations is much more efficient for the device than using JavaScript for that task. JavaScript-based animations run a lot of code at frequent intervals in order to create the appearance of animation. The declarative CSS animations and transitions allow the browser to determine the optimal way to create those effects, which may mean bypassing the CPU altogether.

JavaScript should be kept small both in size and execution time on mobile devices. The JavaScript environments on these devices is much more limited than on a desktop computer, so a good rule of thumb is to use only as much JavaScript as is absolutely necessary to accomplish the goal at hand.

Less memory. Another important limitation of mobile devices is memory capacity. Whereas desktop and laptop computers tend to have many gigabytes of memory, mobile devices have much less. Only recently have mobile

devices reached 1GB of memory, which is present on both the iPhone 5 and Samsung Galaxy S III. Older devices have less memory, so it needs to be a consideration for mobile Web development—especially considering the browser does not actually have access to all of the memory on the device.

Web developers are not used to worrying about memory because it is so plentiful on desktop and laptop computers. The small amount of memory on mobile devices and the way in which

it is used in browsers, however, means it is easy to create a memory problem without knowing it. Even ordinary operations, such as adding new nodes into the Document Object Model (DOM), can cause memory problems if not done properly. When a memory problem gets too large, the browser becomes slow or unresponsive and eventually crashes.

Images are one of the biggest areas of concern regarding memory. Images that are loaded in the DOM, whether

Figure 4. Latency from redirection.

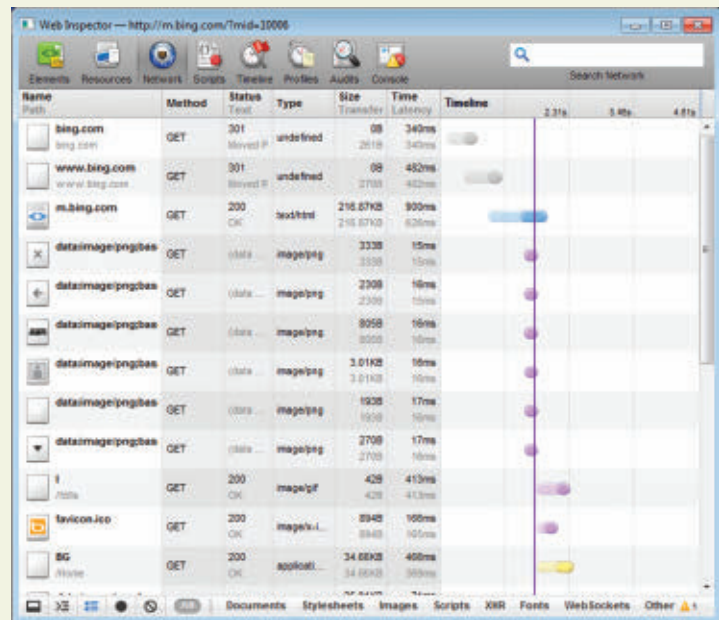
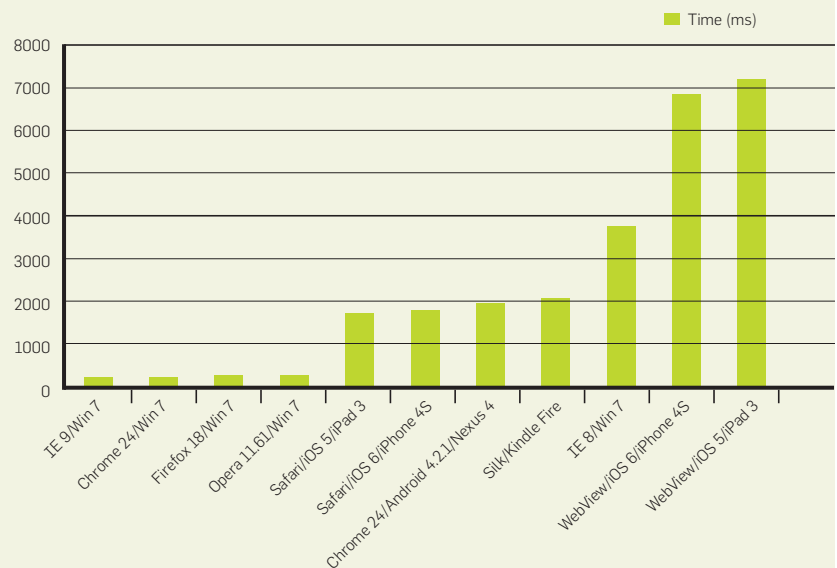


Figure 5. JavaScript performance on various platforms.



or not they are actually visible on the screen, take up memory. Developers who have developed photo-based Web applications for mobile devices have often run into problems causing browsers to crash. The photo-sharing site Flickr had a problem during its first attempt at creating a slideshow in iOS. Whenever it had loaded around 20 images, the browser would crash. Flickr engineer Stephen Woods explained that the only way to prevent this from happening was to periodically remove elements from the DOM as they were no longer needed.¹⁴ Essentially, Flickr decided to keep only a few photos at a time in the DOM and always remove one when another one had to be added.

Part of the Flickr's problem was caused by hardware-accelerated graphics, which use the GPU to calculate what needs to be drawn on the screen. The GPU is much faster than the CPU, so the result is a faster refresh of the display. CSS animations and transitions are hardware accelerated wherever possible by mobile devices (always in iOS and frequently in Android 3+). While this creates a smoother experience, it also requires more memory.

For the GPU to work, parts of the screen must be composited. Composited elements are stored as images in memory and require (width × height × 4) bytes to store. So an image that is 100 × 100 actually takes 40,000 bytes (or about 39KB) in memory. The more composited elements on a page, the more memory will be used and the more likely the browser will crash.

Images are not the only elements that get composited in browsers. DOM elements can also be composited because of certain CSS rules. Early on in mobile Web development, developers noticed hardware-accelerated graphics were much faster, and they tried to find ways to force hardware acceleration even when animations were not necessary. Many blog posts¹³ encourage the use of certain CSS properties to force elements to be hardware accelerated. In general, any time a 3D transformation is applied using CSS, that element gets translated into an image that is then composited just like any other image. For example, some recommend using

code such as this to trigger hardware acceleration:

```
.box {
  transform: translateX(0);
}
```

The transform property contains a 3D transform to translate the element's position. The element does not actually move because the translation is 0, but it still triggers hardware acceleration.

Overzealous developers started adding 3D transforms like this everywhere, thinking it would speed up the mobile Web experience. Unfortunately, it had the unintended side effect of crashing the browser because of memory overuse. Even in cases where the browser did not crash, the experience would get slow as memory was being used up.

Hardware acceleration is a useful feature for Web pages, but it has to be used responsibly. Enabling hardware acceleration on the entire page, for example, is bound to cause memory problems and, potentially, crashes. Developers should not overuse hardware acceleration, applying it only where it makes sense, preferably on small parts of the page, and leaving the rest as normal graphics.

Conclusion

Web development for mobile devices is the unique wrinkle in what has traditionally been a fairly straightforward endeavor. Mobile devices have a lot of power compared with the desktop computer of 10 years ago, but they also have severe limitations that do not have to be dealt with when developing websites solely for the desktop. The latency of over-the-air data transmission automatically means slower download times and necessitates vigilance in keeping the total number of requests on any given page to a minimum. The slower JavaScript engine and less memory means that the same Web page that runs quickly and smoothly on a desktop might be quite slow on a mobile device.

In short, mobile devices force Web developers to think about things they never had to think about before. Web applications now must take into account the type of device being used to determine the best experience for the

user. Mobile devices with high-latency connections, slower CPUs, and less memory must be catered to just as much as desktops with wired connections, fast CPUs, and almost endless memory. Web developers now more than ever need to pay close attention to how they craft interfaces, given these constraints. Byte counts, request counts, memory usage, and execution time all need to be considerations as Web development for mobile devices continues to evolve. **C**

Related articles on queue.acm.org

Making the Mobile Web Faster

Kate Matsudaira

<http://queue.acm.org/detail.cfm?id=2434256>

Mobile Media: Making It a Reality

Fred Kitson

<http://queue.acm.org/detail.cfm?id=1066066>

Mobile Devices in the Enterprise: CTO Roundtable Overview

Mache Creeger

<http://queue.acm.org/detail.cfm?id=2019556>

References

1. HTTP Archive; <http://httparchive.org/>.
2. Hughes-Croucher, T. An engineer's guide to bandwidth; http://developer.yahoo.com/blogs/ydn/posts/2009/10/a_engineers_guide/.
3. modconcat; <http://code.google.com/p/modconcat/>.
4. Mozilla Developer Network. Box-shadow, 2012; <https://developer.mozilla.org/en-US/docs/CSS/box-shadow>.
5. Mozilla Developer Network. Text-shadow, 2012; <https://developer.mozilla.org/en-US/docs/CSS/text-shadow>.
6. Mozilla Developer Network. Using CSS animations, 2012; https://developer.mozilla.org/en-US/docs/CSS/Tutorials/Using_CSS_animations.
7. Mozilla Developer Network. Using CSS gradients, 2013; https://developer.mozilla.org/en-US/docs/CSS/Using_CSS_gradients.
8. Mozilla Developer Network. Using CSS transitions, 2013; https://developer.mozilla.org/en-US/docs/CSS/Tutorials/Using_CSS_transitions.
9. Nir, I. Latency in mobile networks—the missing link; <http://calendar.perfplanet.com/2012/latency-in-mobile-networks-the-missing-link/>.
10. Robertson, G. How powerful was the Apollo 11 computer? <http://downloadsquad.switched.com/2009/07/20/how-powerful-was-the-apollo-11-computer/>.
11. Souders, S. *High Performance Web Sites: Essential Knowledge for Front-end Engineers*. O'Reilly Media, 2007.
12. SunSpider JavaScript Benchmark; <http://www.webkit.org/perf/sunspider/sunspider.html>.
13. Walsh, D. Force hardware acceleration in WebKit with translate3d, 2012; <http://davidwalsh.name/translate3d>.
14. Woods, S. Lessons learned from the Flickr Touch Lightbox. code.flickr.net/2011/07/20/lessons-learned-from-the-flickr-touch-lightbox/.

Nicholas C. Zakas is a Web technologist, author, and speaker. He currently works at Box, and previously worked at Yahoo!, where he was front-end tech lead for the company's homepage and a contributor to the YUI library. He is a strong advocate for development best practices including progressive enhancement, accessibility, performance, scalability, and maintainability. He blogs at <http://www.nczonline.net/> and can be found on Twitter via @slickeat.

© 2013 ACM 0001-0782/13/04



Website performance data has never been more readily available.

BY PATRICK MEENAN

How Fast is Your Website?

THE OVERWHELMING EVIDENCE indicates a website's performance (speed) correlates directly to its success, across industries and business metrics. With such a clear correlation (and even proven causation), it is important to monitor how your website performs. So, how fast is your website?

First, it is important to understand that no single number will answer that question. Even if you have defined exactly what you are trying to measure on your website, performance will vary widely across your user base and across the different pages on your site.

Here, I discuss active testing techniques that have been traditionally used and then explain newer technologies that permit the browser to report back to the server accurate timing data.

Traditionally, monitoring tools are used in active testing to measure the

performance of a website, and the results end up being plotted on a time-series chart. You may choose specific pages to measure and geographic locations from which to measure, and then the test machines load the pages periodically from the various locations and the performance gets reported. The results are usually quite consistent and provide a great baseline for identifying variations, but the measurements are not representative of actual users.

The main limiting factors in using

active testing for understanding Web performance are:

- **Page Selection.** Active testing usually tests only a small subset of the pages that users visit.

- **Browser Cache.** The test machines usually operate with a clear browser cache and do not show the impact of cached content. This includes content that not only is shared across pages but also is widely distributed (such as JavaScript for social-sharing widgets, shared code libraries, and code used by advertising systems).

- **Machine Configuration.** The test machines tend to be clean configurations without viruses, adware, browser

toolbars, or antivirus and all sorts of other software that mess with the performance of the user's machine and browsing in the real world.

- **Browsers.** The browser wars have never been more alive, with various versions of Internet Explorer, Chrome, Firefox, Safari, and Opera all maintaining significant market share. Active testing is usually limited to a small number of browsers, so you must try to select one that is representative of your user base (assuming you even have a choice).

- **Connectivity.** Users have all sorts of Internet connections with huge differences in both latency and bandwidth,

from dial-up to 3G mobile to cable, DSL, and fiber-broadband connections. With active testing you usually must choose a few specific configurations for testing or perform tests directly from a data center.

The impact of connectivity cannot be understated. As illustrated in Figure 1, bandwidth can have a significant impact on Web performance, particularly at the slower speeds (less than 2Mbps), and latency has a near-linear correlation with Web performance.

Testing from within a data center skews heavily with effectively unlimited bandwidth and no last-mile latency (which for actual users can range anywhere from 20 to 200-plus milliseconds depending on the type of connection). The connectivity skew also is exaggerated by CDNs (content distribution networks) because these usually have edge nodes collocated in the same data centers as the test machines for many monitoring providers, reducing latency to close to zero.

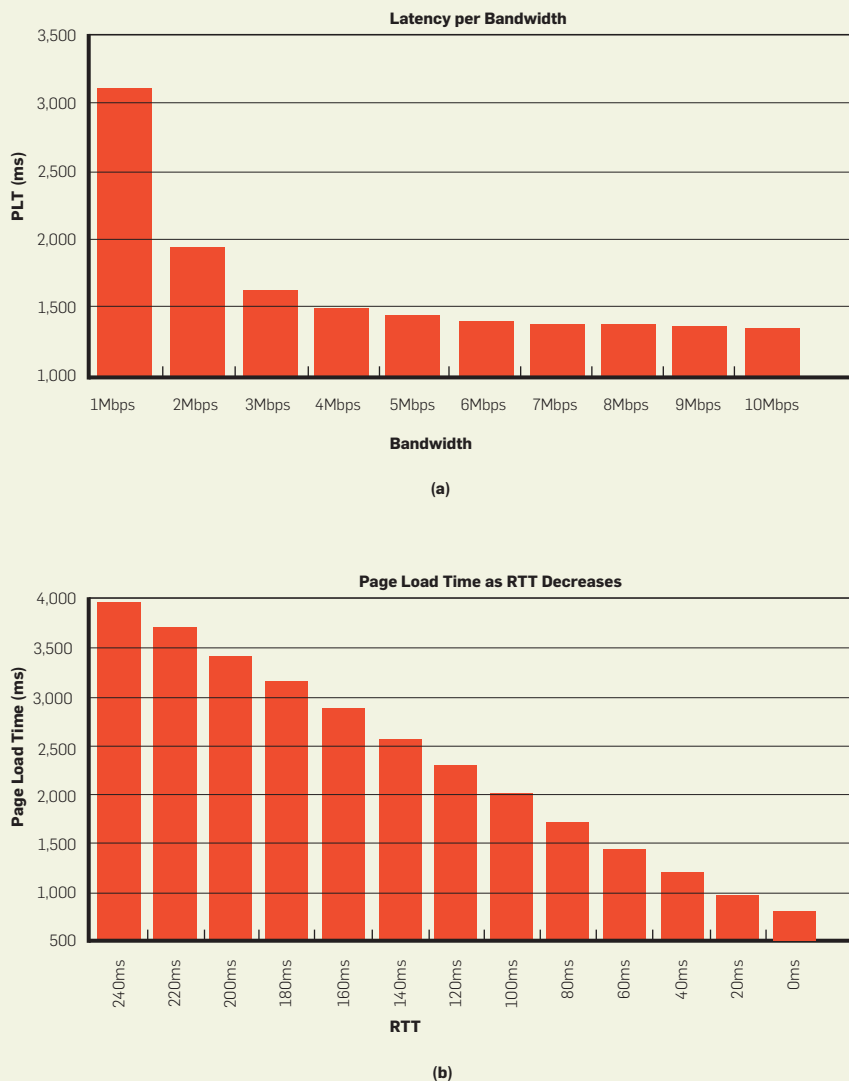
Passive Performance Reporting

Given the constraints of active testing in measuring performance, a lot of work has focused on reporting the actual performance experienced by end users as they browse a site. Historically, sites have implemented their own solutions for measuring and reporting performance, although there have always been caveats because JavaScript on the page did not have the ability to report on the full experience. Reporting on the performance from actual user sessions is generally referred to as RUM (real user measurement).

When loading a page, the browser generally:

- Does a DNS lookup to map the host name to an IP address.
- Establishes a TCP connection to the server.
- Negotiates an encrypted connection (for HTTPS pages).
- Sends a request to the server for the page HTML.
- If the server responds with a redirect, repeats all of the above steps for the new location.
- Downloads the HTML response from the server.
- Parses the HTML, downloads all of the referenced content, and executes the page code.

Figure 1. Bandwidth and latency.



Source: <http://www.belshe.com/2010/05/24/more-bandwidth-doesnt-matter-much/>

The last step is very complex and constitutes the majority of time (usually over 80%^{4,6}) consumed by most sites and is the only part of the experience that you have been able to directly measure from JavaScript. When a page is instrumented using JavaScript to measure performance, JavaScript's first chance to execute is at the point where the HTML has been downloaded from the server and the browser has begun executing the code. That leaves the approximately 20% of the page-load time outside of the measurement capabilities of in-page code. Several clever workarounds have been implemented over the years, but getting a reliable start time for measurements from the real world has been the biggest barrier to using RUM. Over the years browsers implemented proprietary metrics that they would expose that provided a page start time (largely because the browser vendors also ran large websites and needed better instrumentation themselves), but the implementations were not consistent with each other and browser coverage was not very good.

In 2010, the browser vendors got together under the W3C banner and formed the Web Performance Working Group¹⁵ to standardize the interfaces and work toward improving the state of Web-performance measurement and APIs in the browsers. (As of this writing, browser support for the various timing and performance information is somewhat varied, as shown in the accompanying table.)

In late 2010 the group released the Navigation Timing specification,⁹ which has since been implemented in Internet Explorer (9+ desktop and mobile), Chrome (15+ all platforms), Firefox (7+), and Android (4+).

The largest benefit navigation timing provides is that it exposes a lot of timings that lead up to the HTML loading, as shown in Figure 2. In addition to providing a good start time, it exposes information about any redirects, DNS lookup times, time to connect to the server, and how long it takes the Web server to respond to the request—for every user and for every page the user visits.

The measurement points are exposed to the Document Object Model (DOM) through the performance object and make it trivial to calculate load times (or arbitrary intervals, really) from JavaScript.

Browser support for performance information.

	IE (Desktop and Mobile)	Chrome (Desktop and Android)	Firefox (Desktop and Android)	Safari (Desktop and Mobile)
Navigation	9+	6+	7+	
Resource Timing	10+	26+		
requestAnimationFrame	10+	10+	4+	6+
High Resolution Time	10+	21+	15+	

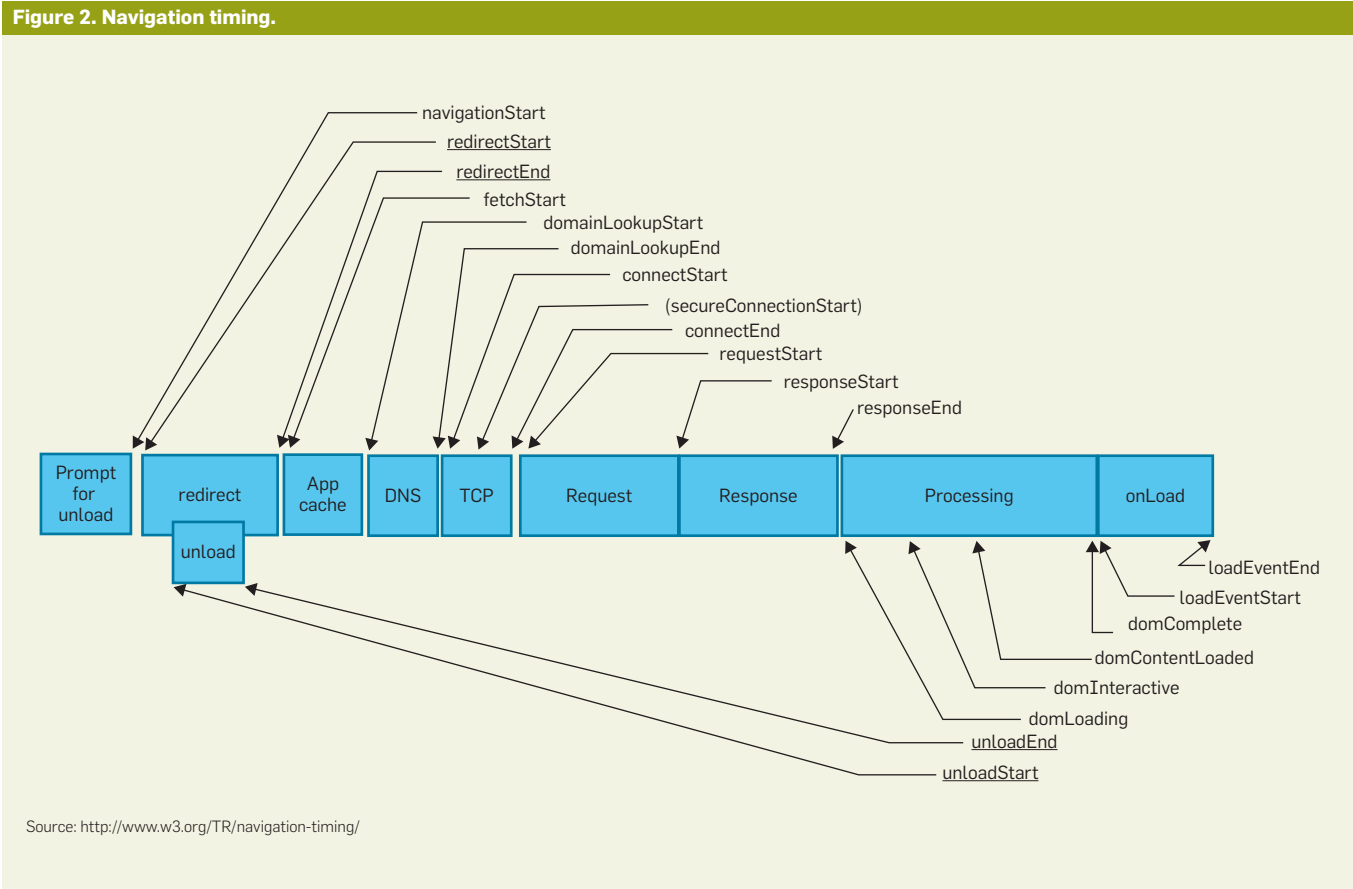


Figure 3. Distribution of page load times.

The following code calculates the page load time:

```
var loadTime = performance.
  timing.loadEventStart - performance.
  timing.navigationStart;
```

Just make sure to record measurements after the point you are measuring (loadEventStart will not be set until the actual load event happens, so attaching a listener to the load event is a good way of knowing it is available for measurement).

The times are reported as regular JavaScript times and can be compared with arbitrary points in time:

```
var now = new Date().getTime();
var elapsed = now - performance.
  loadEventStart;
```

What is interesting about users' clocks is that they don't always move forward and are not always linear, so error checking and data filtering are necessary to make sure only reasonable values are being reported. For example, if a user's computer connects to a time server and adjusts the clock, it can jump forward or backward in between measurements, resulting in negative times. If a user switches apps on a phone, then the page can be paused midload until the user opens the browser again, leading to load times that span days.

The W3C Web Performance Group recognized the clock issue, as well as the need for granularity beyond one-millisecond resolution, and also introduced the High Resolution Time specification.⁷ This is a time measurement as accurate as the platform it is running on supports (at least one-millisecond resolution but significantly better than that on most platforms), and it is guaranteed always to increase and not be skewed by clock changes. This makes it significantly better for measuring elapsed time intervals. It is exposed on the DOM to JavaScript as `performance.now()` and is relative to the page navigation start time.

Sites can now report on the actual performance for all of their users, and monitoring and analytics providers are also providing ways to mine the data (Google Analytics reports the navigation timing along with business metrics, for example²), so websites may already be getting real user performance-measurement data. Google Analytics also makes an interface available for reporting arbitrary performance data so website operators can add timings for anything else they would like to measure.³

Unlike the results from active testing, the data from real users is noisy, and averages tend not to work well at all. Performance data turns into regular analytics like the rest of a company's business metrics, and managers have to do a lot of similar analysis such as looking at users from specific regions, looking at percentiles instead of averages, and looking at the distribution of the results.

Figure 4. Different page loads with the same load event time.

For example, Figure 3 shows the distribution of page-load times for recent visitors to a website (approximately two million data points). In this case U.S. and European traffic skews toward faster load times, while Asia and Africa skew slower with a significant number of users experiencing page-load times of more than 11 seconds. Demographic issues drive some of the skew (particularly around Internet connectivity in different regions), but it is also a result of the CDN used to distribute static files. The CDN does not have nodes in Eastern Asia or Africa, so one experiment would be to try a different CDN for a portion of the traffic and watch for a positive impact on the page-load times from real users.

When is Timing Done?

One thing to consider when measuring the performance of a site is when to stop the timing. Just about every measurement tool or service will default to reporting the time until the browser's load event fires. This is largely because it is the only well-defined point that is consistent across sites and reasonably consistent across browsers.⁸ In the world of completely static pages, the load event is the point where the browser has finished loading all of the content referred to by the HTML (including all style sheets and images, among others).

All of the browsers implement the basic load event consistently as long as scripts on the page do not change the content. The HTML5 specification clarified the behavior for when scripts modify the DOM (adding images, other scripts, and so on), and the load event now also includes all of the resources that are added to the DOM dynamically. The notable exception to this behavior is Internet Explorer prior to version 10, which did not block the load event for dynamically inserted content (since it predated the spec that made the behavior clear).

The time until the load event is an excellent technical measurement, but it may not convey how fast or slow a page was for the user. The main issues are that it measures the time until every request completes, even those that are not visible to the user



One thing to consider when measuring the performance of a site is when to stop the timing. Just about every measurement tool or service will default to reporting the time until the browser's load event fires.



(content below the currently displayed page, invisible tracking pixels, among others), and it does not tell you anything about how quickly the content was displayed to the user. As illustrated in Figure 4, a page that displays a completely blank screen right up until the load event and a page that displays all visible content significantly earlier can both have the same time as reported by measuring the load event.

It is also quite easy to optimize for the load-event metric by loading all of the content through JavaScript after the page itself has loaded. That will make the technical load event fire very fast, but the user experience will suffer as a result (and will no longer be measured).

Several attempts have been made to find a generic measurement that can accurately reflect the user experience:

- *Time to first paint.* If implemented correctly, this can tell you the first point in time when the user sees something other than a blank white screen. It doesn't necessarily mean the user sees anything useful (as in the previous example, it is just the logo and menu bar), but it is important feedback telling the user that the page is loading. There is no consistent way to measure this from the field. Internet Explorer exposes a custom `msFirstPaint` time on the `performance.timing` interface, and Chrome exposes a time under `chrome.loadTimes().firstPaintTime`; in both cases, however, it is possible that the first thing the browser painted was still a blank white screen (though this is better than nothing and good for trending).

- *Above-the-fold time.*¹ This measures the point in time when the last visual change is made to the visible part of the page. It was an attempt at a lab-based measurement that captured the user experience better. It can be tuned to work reasonably well to reveal when the last bit of visible content is displayed, but it doesn't distinguish between a tiny social button being added to the page and the entire page loading late (in the example, the first page adds a social button at the end, and both pages would have the same above-the-fold measurement time). The above-the-fold time

Figure 5. Visual progress display.

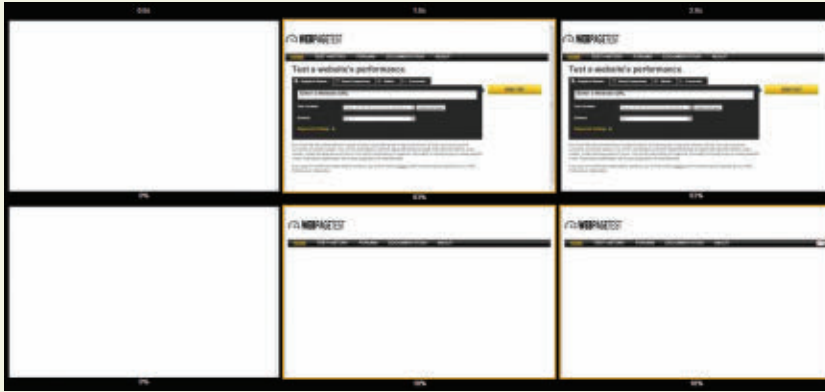
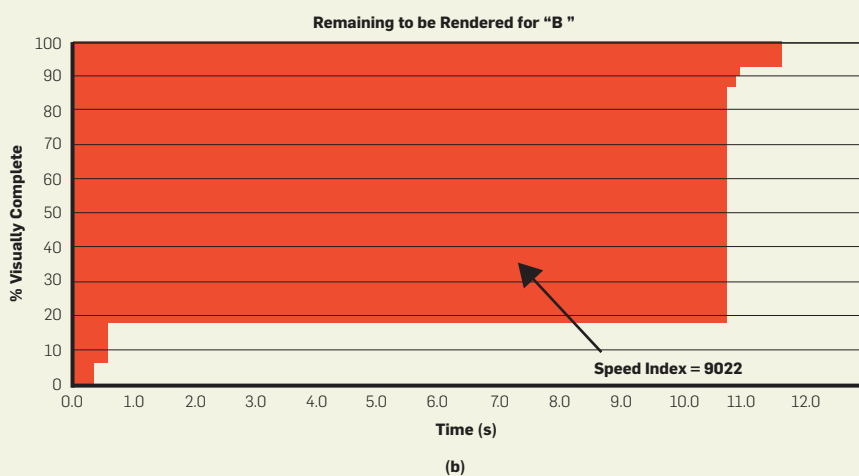
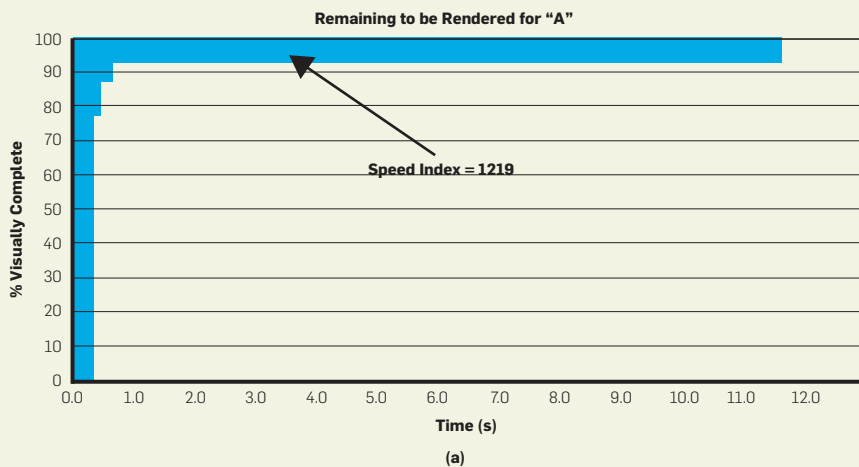


Figure 6. Speed index value.



is available only in lab environments where the visual progress of a page loading can be recorded.

Speed index.¹⁴ After experimenting with several point-in-time measurements, we evolved our visible measurement frameworks to capture the visual progress of displaying a page over time. This current best effort for representing the user experience in a single number takes the visual progress as a page is loading and calculates a metric based on how quickly content is painted to the screen. Figure 5 shows the example from Figure 4 but with visual progress included each step of the way.

Figure 6 shows how to plot the progress and calculate the unrendered area over time to boil the overall experience down to a single number. The speed index is an improvement in the ability to capture the user experience, but it is only measurable in a lab. It still requires that you determine what the endpoint of the measurement is, and it does not deal well with pages that have large visual changes as they load.

► *Custom measurements.* Nothing beats application-specific knowledge and measurements. While there is no perfect solution for measuring the user experience generically, each application owner can instrument specific performance points.

As browsers go through a page and build the DOM, they also execute inline scripts when they get to them. They cannot execute the scripts until all previous scripts have been executed. If you want to know when the browser made it to the main article, then you can just add a tiny inline script that records the current time right after the HTML for the main article. When you report the performance metrics, you can gather the different custom measurements and include them as well.

The W3C Web Performance Group considered this use case as well and created the User Timing specification.¹³ It provides a simple way of marking points in time using `performance.mark("label")` and a standard way of querying all of the custom measurements later. This is more for standardization and ease of use than anything else since you

could always store the timings in your own JavaScript variables, but by using the standard interfaces, you also make it possible for third-party reporting tools to report on your custom metrics.

This still isn't perfect because there is no guarantee the browser actually painted the content to the screen, but it is a good proxy for it. To be even more accurate with the actual painting event, the animation timing interface can request a callback when the browser is ready to paint through `requestAnimationFrame()`.¹²

Per-Resource Timings

One specification the community has been eager to see implemented is the resource timing interface.¹¹ It exposes timing information about every network request the browser had to make to load a page and what triggered the resource to be requested (whether stylesheet, script, or image).

Some security provisions are in place to limit the information that is provided across domains, but in all cases the list (and count) of requests that were made on the network are available, as are the start and end times for each of them. For requests where you have permission to see the granular timing information, you get full visibility into the individual component timings: DNS lookup, time to connect, redirects, SSL (Secure Sockets Layer) negotiation time, server response time, and time to download.

By default the granular timing is visible for all of the resources served by the same domain as the current page, but cross-domain visibility can also be enabled by including a "Timing-Allow-Origin" response header for requests served by other domains (the most common case being a separate domain from which static content can be served).

You can query the list of resources through:

```
performance.getEntriesByType:
var resourceList
= performance.getEntriesBy
Type("resource");
for (i = 0; i < resourceList.length; i++)
...
```

The possibilities for diagnosing issues in the field with this interface are huge:

- ▶ Detect third-party scripts or beacons that intermittently perform poorly.
- ▶ Detect performance issues with different providers in different regions.
- ▶ Detect the most likely cause of a slow page for an individual user.
- ▶ Report on the effectiveness of your static resources being cached.
- ▶ And a whole lot more...

The default buffer configuration will store up to 150 requests, but there are interfaces available to:

- ▶ Adjust the buffer size (`performance.setResourceTimingBufferSize`).
- ▶ Clear it (`performance.clearResourceTimings`).
- ▶ Get notified when it is full (`onresourcetimingbufferfull`).

Theoretically, you could report all of the timings for all of your users and be able to diagnose any session after the fact, but that would involve a significant amount of data. More likely than not you will want to identify specific things to measure or diagnose in the field and report on the results only of the analysis that is done in the client (and this being the Web, you can change and refine this as needed when you need more information about specific issues).

Join the Effort

If you run a website, make sure you are regularly looking at its performance. The data has never been more readily available, and it is often quite surprising.

The W3C Web Performance Working Group has made great progress over the past two years in defining interfaces that will be useful for getting performance information from real users and for driving the implementation in the actual browsers. The working group is quite active and plans to continue to improve the ability to measure performance, as well as standardize on solutions for common performance issues (most notably, a way to fire tracking beacons without holding up a page).⁵ If you have thoughts or ideas, I encourage you to join the mailing list¹⁰ and share them with the working group.

Related articles on queue.acm.org

High Performance Web Sites

Steve Souders

<http://queue.acm.org/detail.cfm?id=1466450>

Building Scalable Web Services

Tom Killalea

<http://queue.acm.org/detail.cfm?id=1466447>

Improving Performance on the Internet

Tom Leighton

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

References

1. Brutlag, J., Abrams, Z. and Meenan, P. Above-the-fold time: measuring Web page performance visually (2011); http://cdn.oreillystatic.com/en/assets/1/event/62/Above%20the%20Fold%20Time_%20Measuring%20Web%20Page%20Performance%20Visually%20Presentation.pdf.
2. Google Analytics (2012); Measure your website's performance with improved Site Speed reports; <http://analytics.blogspot.com/2012/03/measure-your-websites-performance-with.html>.
3. Google Developers (2013); User timings – Web tracking (ga.js); <https://developers.google.com/analytics/devguides/collection/gajs/gaTrackingTiming>.
4. Hallock, A. Some interesting performance statistics; <http://torbit.com/blog/2012/09/19/some-interesting-performance-statistics/>.
5. Mann, J. W3C Web performance: continuing performance investments. IEBlog; <http://blogs.msdn.com/b/ie/archive/2012/11/27/w3c-web-performance-continuing-performance-investments.aspx>.
6. Souders, S. The performance golden rule; <http://www.stevesouders.com/blog/2012/02/10/the-performance-golden-rule/>.
7. W3C. High Resolution Timing (2012); <http://www.w3.org/TR/hr-time/>.
8. W3C. HTML5 (2012); <http://www.w3.org/TR/2012/CR-html5-20121217/webappapis.html#handler-window-onload>.
9. W3C. Navigation timing (2012); <http://www.w3.org/TR/navigation-timing/>.
10. W3C. Public-web-perf@w3.org mail archives (2012); <http://lists.w3.org/Archives/Public/public-web-perf/>.
11. W3C. Resource timing (2012); <http://www.w3.org/TR/resource-timing/>.
12. W3C. Timing control for script-based animations (2012); <http://www.w3.org/TR/animation-timing/>.
13. W3C. User timing (2012); <http://www.w3.org/TR/user-timing/>.
14. WebPagetest documentation; <https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index>.
15. Web Performance Working Group. W3C; <http://www.w3.org/2010/webperf/>.

Patrick Meenan has worked on Web performance in one form or another for the last 12 years and currently works at Google to make the Web faster. He created the popular open source WebPagetest Web performance measurement tool, runs the free instance of it at <http://www.webpagetest.org/>, and can frequently be found in the forums helping site owners understand and improve their website performance. He also helped found the WPO foundation, a non-profit organization focused on Web Performance Optimization.

The programmability of FPGAs must improve if they are to be part of mainstream computing.**BY DAVID F. BACON, RODRIC RABBAH, AND SUNIL SHUKLA**

FPGA Programming for the Masses

WHEN LOOKING AT how hardware influences computing performance, we have general-purpose processors (GPPs) on one end of the spectrum and application-specific integrated circuits (ASICs) on the other. Processors are highly programmable but often inefficient in terms of power and performance. ASICs implement a dedicated and fixed function and provide the best power and performance characteristics, but any functional change requires a complete (and extremely expensive) re-spinning of the circuits.

Fortunately, several architectures exist between these two extremes. Programmable logic devices (PLDs) are one such example, providing the best of both worlds. They are closer to the hardware and can be reprogrammed.

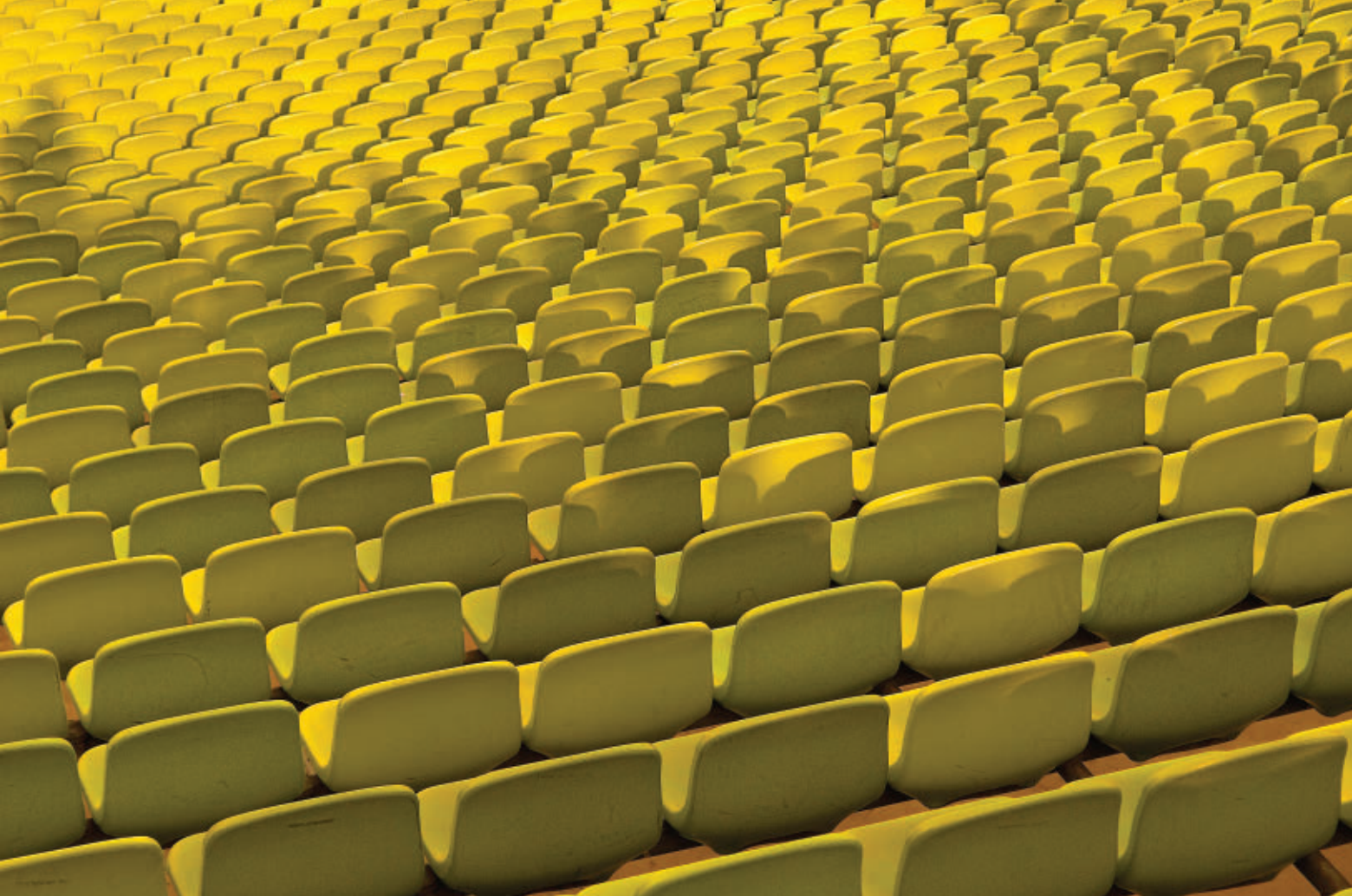
The most prominent example of a PLD is a field programmable gate array (FPGA). It consists of look-up tables (LUTs), which are used to implement combinational logic; and flip-flops (FFs), which are

used to implement sequential logic. Apart from the homogeneous array of logic cells, an FPGA also contains discrete components such as BRAMs (block RAMs), digital signal processing (DSP) slices, processor cores, and various communication cores (for example, Ethernet MAC and PCIe).

BRAMs, which are specialized memory structures distributed throughout the FPGA fabric in columns, are of particular importance. Each BRAM can hold up to 36Kbits of data. BRAMs can be used in various form factors and can be cascaded to form a larger logical memory structure. Because of the distributed organization of BRAMs, they can provide terabytes of bandwidth for memory bandwidth-intensive applications. Xilinx and Altera dominate the PLD market, collectively holding more than an 85% share.

FPGAs were long considered low-volume, low-density ASIC replacements. Following Moore's Law, however, FPGAs are getting denser and faster. Modern-day FPGAs can have up to two million logic cells, 68Mbits of BRAM, more than 3,000 DSP slices, and up to 96 transceivers for implementing multigigabit communication channels.²³ The latest FPGA families from Xilinx and Altera are more like a system-on-chip (SoC), mixing dual-core ARM processors with programmable logic on the same fabric. Coupled with higher device density and performance, FPGAs are quickly replacing ASICs and application-specific standard products (ASSPs) for implementing fixed function logic. Analysts expect the programmable integrated circuit (IC) market to reach the \$10 billion mark by 2016.²⁰

The most perplexing fact is how an FPGA running at a clock frequency that is an order of magnitude lower than CPUs and GPUs (graphics processing units) is able to outperform them. In several classes of applications, especially floating-point-based ones, GPU performance is either slightly better or very close to that of an FPGA. When it comes to power efficiency (perfor-



mance per watt), however, both CPUs and GPUs significantly lag behind FPGAs, as shown in the available literature comparing the performance of CPUs, GPUs, and FPGAs for different classes of applications.^{6,18,19}

The contrast in performance between processors and FPGAs lies in the architecture itself. Processors rely on the Von Neumann paradigm where an application is compiled and stored in instruction and data memory. They typically work on an instruction and data fetch-decode-execute-store pipeline. This means both instructions and data have to be fetched from an external memory into the processor pipeline. Although caches are used to alleviate the cost of expensive fetch operations from external memory, each cache miss incurs a severe penalty. The bandwidth between processor and memory is often the critical factor in determining the overall performance. The phenomenon is also known as “hitting the memory wall.”²¹

FPGAs have programmable logic cells that could be used to implement an arbitrary logic function both spatially and temporally. FPGA designs

implement the data and control path, thereby getting rid of the fetch and decode pipeline. The distributed on-chip memory provides much-needed bandwidth to satisfy the demands of concurrent logic. The inherent fine-grained architecture of FPGAs is very well suited for exploiting various forms of parallelism present in the application, ranging from bit-level to task-level parallelism. Apart from the conventional reconfiguration capability where the entire FPGA fabric is programmed with an image before execution, FPGAs are also capable of undergoing partial dynamic reconfiguration. This means part of the FPGA can be loaded with a new image while the rest of the FPGA is functional. This is similar to the concept of paging and virtual memory in the processor taxonomy.

Various kinds of FPGA-based systems are available today. They range from heterogeneous systems targeted at high-performance computing that tightly couple FPGAs with conventional CPUs (for example, Convey Computers), to midrange commercial-off-the-shelf workstations that use

PCIe-attached FPGAs, to low-end embedded systems that integrate embedded processors directly into the FPGA fabric or on the same chip.

Programming Challenges

Despite the advantages offered by FPGAs and their rapid growth, use of FPGA technology is restricted to a narrow segment of hardware programmers. The larger community of software programmers has stayed away from this technology, largely because of the challenges experienced by beginners trying to learn and use FPGAs.

Abstraction. FPGAs are predominantly programmed using hardware description languages (HDLs) such as Verilog and VHDL. These languages, which date back to the 1980s and have seen few revisions, are very low level in terms of the abstraction offered to the user. A hardware designer thinks about the design in terms of low-level building blocks such as gates, registers, and multiplexors. VHDL and Verilog are well suited for describing a design at that level of abstraction. Writing an application at a behavioral level and leaving its destiny in the hands of a synthe-

sis tool is commonly considered a bad design practice.

This is in complete contrast with the software programming languages, which have evolved over the past 60 years. Programming for CPUs enjoys the benefits of well-established ISAs (instruction set architectures) and advanced compilers that offer a much simpler programming experience. Object-oriented and polymorphic programming concepts, as well as automatic memory management (garbage collection) are no longer seen just as desirable features but as necessities. A higher level of abstraction drastically increases a programmer's productivity and reduces the likelihood of bugs, resulting in a faster time to market.

Synthesizability. Another language-related issue is that only a narrow subset of VHDL and Verilog is synthesizable. To make matters worse, there is no standardization of the features supported by different EDA (elec-

tronic design automation) tools. It is largely up to the EDA tool vendors to decide which language features they want to support.

While support for data types such as char, int, long, float, and double is integral to all software languages, VHDL and Verilog have long supported only bit, array of bits, and integer. A recent revision to VHDL standard (IEEE 1076-2008) introduced fixed-point and floating-point types, but most, if not all, synthesis tools do not support these abstractions yet.

Figure 1 is a C program that converts Celsius to Fahrenheit. The language comes with standard libraries for a number of useful functions. To implement the same program in VHDL or Verilog for FPGA implementation, the user must generate a netlist (pre-compiled code) using FPGA vendor-specific IP (intellectual property) core-generator tools (for example, Coregen for Xilinx and MegaWizard for Altera).

Then the generated netlist is connected in a structural manner. Figure 2 shows the top-level wrapper consisting of structural instantiation of leaf-level modules; namely, a double-precision floating-point multiplier (fp mult) and adder (fp add). (For brevity, the leaf-level modules are not shown here.)

Verification. Design and verification go hand in hand in the hardware world. Verification is easily the largest chunk of the overall design-cycle time. It requires a TB (test bench) to be created either in HDL or by using a high-level language such as C, C++, or SystemC that is connected to the design under test (DUT) using the FLI (foreign language interface) extension provided by VHDL and Verilog. The DUT and TB are simulated using event-based simulators (for example, Mentor Graphics ModelSim and Cadence Incisive Unified Simulator). Each signal transaction is recorded and displayed as a waveform. Waveform-based debugging works for small designs but can quickly become tedious as design size grows. Also, the simulation speed decreases very quickly with the increasing design size. In general, simulations are many orders of magnitudes slower than the real design running in hardware. Often, large-system designers have to turn to prototyping (emulation) using FPGAs to speed up design simulation. This is in complete contrast with the way software verification works. From simple printf statements to advanced static and dynamic analysis tools, a software programmer has numerous tools and techniques available to debug and verify programs in a much simpler yet powerful manner.

Design and tool flow. Apart from the language-related challenges, programming for FPGAs requires the use of complex and time-consuming EDA tools. Synthesis time—the time taken to generate bitstreams (used to program the FPGA) from source code—can vary anywhere from a few minutes to a few days depending on the code size and complexity, target FPGA, synthesis tool options, and design constraints.

Figure 3 shows the design flow for Xilinx FPGAs.²² After functional verification, the source code—consisting of VHDL, Verilog, and netlists—is fed to the synthesis tool. The synthesis phase compiles the code and

Figure 1. Celsius to Fahrenheit conversion in C.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 double main (int argc, char* argv[]) {
5     double celsius = atof(argv[1]);
6     return (9*celsius/5 + 32);
7 }
```

Figure 2. Celsius to Fahrenheit conversion in Verilog.

```
1 module c2f (
2     input clk,
3     input rst,
4     input [63:0] celsius,
5     input input_valid,
6     output [63:0] fahrenheit,
7     output result_valid);
8
9     localparam double_9div5 = 64'h3FFCCCCCCCCCD; // 1.8 (=9/5)
10    localparam double_32 = 64'h4040000000000000; // 32.0
11    wire [63:0] temp;
12
13    // temp = 1.8 * celsius
14    fp_mult mult_9div5_i (.clk(clk), .rst(rst),
15        .in1(celsius), .in2(double_9div5), .in_valid(input_valid),
16        .out(temp), .out_valid(temp_valid));
17
18    // fahrenheit = temp + 32
19    fp_add add_32_i (.clk(clk), .rst(rst),
20        .in1(temp), .in2(double_32), .in_valid(temp_valid),
21        .out(fahrenheit), .out_valid(result_valid));
22
23 endmodule
```

produces a netlist (.ngc). The *translate phase* merges all the synthesized netlists and the physical and timing constraints to produce a native generic database (NGD) file. The *map phase* groups logical symbols from the netlists into physical components such as LUTs, FFs, BRAMs, DSP slices, IOBs (input/output blocks), among others. The output is stored in native circuit description (NCD) format and contains information about switching delays. The map phase results in an error if the design exceeds the available resources or the user-specified timing constraints are violated by the switching delay itself. The *PAR (place-and-route) phase* performs the placement and routing of the mapped symbols on the actual FPGA device. For some FPGA devices, placement is done in the map phase. PAR stores the output in NCD format and notifies the user of timing errors if the total delay (switching+routing) exceeds the user-specified timing constraints.

Following PAR, timing analysis is performed to generate a detailed timing report consisting of all delays for different paths. Users can refer to the timing analysis report to determine which critical paths fail to meet the timing requirements and then fix them in the source code. Most of the synthesis errors mean the user must go back to the source code and repeat the whole process. Once the design meets the resource and timing constraints, a bitstream can be generated. The bitgen tool takes in a completely routed design and generates a configuration bitstream that can be used to program the FPGA. Finally, the design can be downloaded to the FPGA using the Impact tool.

The design flow described in Figure 3 varies from one FPGA vendor to another. Moreover, to make good use of the synthesis tool, the user must have good knowledge of the target FPGA architecture because several synthesis options are tied to the architecture. The way these options are tuned ultimately decides whether the design will meet resource and timing constraints.

Design portability is another big challenge because porting a design from one FPGA to another often requires re-creation of all the FPGA device-specific IP cores. Migrating designs

from one FPGA vendor to another is an even bigger challenge.

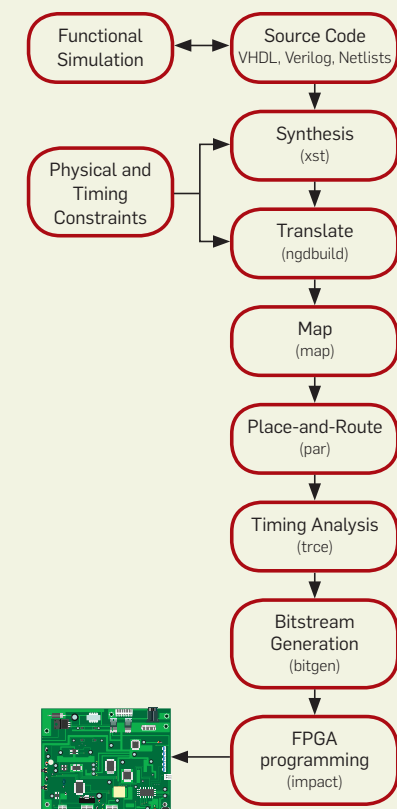
High-Level Synthesis

High-level synthesis (HLS) refers to the conversion of an algorithmic description of an application into either a low-level register-transfer level (RTL) description or a digital circuit. Over the past decade or so, several languages and compiler frameworks were proposed to ease the pain of programming FPGAs. They all aim to raise the level of abstraction in which the user must write a program to compile it down to VHDL or Verilog. Most of these frameworks can be classified into five categories: HDL-like languages; C-based frameworks; Compute Unified Device Architecture (CUDA)/Open Computing Language (OpenCL)-based frameworks; high-level language-based frameworks; and model-based frameworks.

HDL-like languages. Although only minute incremental changes have been made to VHDL and Verilog since they were proposed, SystemVerilog represents a big jump. It has some semantic similarity to Verilog, but it is more dissimilar than similar. SystemVerilog consists of two components: the synthesizable component that extends and adds several features to the Verilog-2005 standard; and the verification component that heavily uses an object-oriented model and is more akin to object-oriented languages such as Java than it is to Verilog. SystemVerilog substantially improves on the type system and parametrizability of Verilog. Unfortunately, EDA tools offer limited support for the language today: the Xilinx synthesis tool does not support SystemVerilog, and the Altera synthesis tool supports a small subset.

Another notable mention in this category is BSV (Bluespec SystemVerilog), which incorporates concepts such as modules, module instances, and module hierarchies just like Verilog and VHDL, and follows the SystemVerilog model of interfaces to communicate between modules. In Bluespec, module behavior is expressed using guarded atomic actions that are basically concurrent finite state machines (FSMs), a concept well known to hardware programmers. A program written in Bluespec is passed on to the Bluespec compiler, which generates an efficient RTL de-

Figure 3. Design flow for Xilinx FPGAs.



scription. A 2011 article in *acmqueue* provides a comprehensive description of the language and compiler.¹⁴

C-based frameworks. Most of the early work in the field of HLS was done on C-based languages such as C, C++, and SystemC. Most of these frameworks restrict the programmer to only a small subset of the parent language. Features such as pointers, recursive functions, and dynamic memory allocations are prohibited. Some of these frameworks require users to annotate the program extensively before compilation. A comprehensive list of C-based frameworks can be found in João M.P. Cardosa and Pedro C. Diniz's book, *Compilation Techniques for Reconfigurable Architectures*.⁷

Interestingly, all the EDA vendors have developed or acquired tool suites to enable high-level synthesis. Their HLS frameworks (for example, Cadence C-to-Silicon Compiler, Synopsys Symphony C Compiler, Mentor Graphics Catapult C, and Xilinx Vivado), with the exception of the one proposed by Altera, are based on C-like languages. The following section looks at the Xilinx Vivado HLS tool.

Xilinx Vivado (previously autopilot). In 2011 Xilinx acquired the AutoPilot framework^{5,8} developed by AutoESL and is now offering it as a part of the Vivado Design Suite. It supports compilation from a behavioral description written in C/C++/SystemC to RTL. The behavioral synthesis consists of four phases: compilation and elaboration; advanced code transformation; core behavioral and communication synthesis; and microarchitecture generation.

The behavioral code in C/C++/SystemC is parsed and compiled by a GNU Compiler Collection (GCC)-compatible compiler front end. Several hardware-oriented features are added to the front end such as bit-width optimization for data types to use exactly the right number of bits required to represent a data type. For SystemC designs, an elaboration phase extracts processes, ports, and other interconnect information, and it forms a synthesis data model.

The synthesis data model undergoes several compiler optimizations (for example, constant propagation, dead-code elimination, and bit-width analysis) in the advanced code transformation phase.

In the core behavioral and communication synthesis phase, Vivado takes into account the user-specified constraints (for example, frequency, area, throughput, latency, and physical location constraints) and the target FPGA device architecture to do scheduling and resource binding.

In the microarchitecture generation phase, the compiler backend produces VHDL/Verilog RTL together with the synthesis constraints, which could be passed as an input to the synthesis tool. The tool also generates an equivalent description in SystemC, which could be used to do verification and equivalence checking of the generated RTL code.

CUDA/OpenCL-based frameworks. The popularity of GPUs for general-purpose computing soared with the release of the CUDA framework by Nvidia in 2006. This framework consists of language and runtime extensions to the C programming language and driver-level APIs for communication between the host and GPU device. With CUDA, researchers and developers can exploit massive SIMD (single instruction, multiple data)-style architectural parallel-



Despite the advantages offered by FPGAs and their rapid growth, use of FPGA technology is restricted to a narrow segment of hardware programmers.



ism for running computationally intensive tasks. In 2008, OpenCL was made public as an open standard for parallel programming on heterogeneous systems. OpenCL supports execution on targets such as CPUs and GPUs. To leverage the popularity of CUDA and OpenCL, a number of compiler frameworks have been proposed to convert CUDA¹⁶ and OpenCL^{3,13,15} code to VHDL/Verilog. One of these products is the Altera OpenCL framework.

Altera OpenCL-to-FPGA framework. Altera proposed a framework for synthesizing bitstreams from OpenCL.³ Figure 4 shows the Altera OpenCL-to-FPGA framework for compiling an OpenCL program to Verilog for co-execution on a host and Altera FPGA. The framework consists of a kernel compiler, host library, and system integration component. The kernel compiler is based on the open source LLVM compiler infrastructure to synthesize OpenCL kernel code into hardware. The host library provides APIs and bindings for establishing communication between the host part of the application running on the processor and the kernel part of the application running on the FPGA. The system integration component wraps the kernel code with memory controllers and a communication interface (such as PCIe).

The user application consists of a host component written in C/C++ and a set of kernels written in OpenCL. The OpenCL kernels are compiled using the open source LLVM infrastructure. The parser completes the first step in the compilation, producing an intermediate representation (IR) consisting of basic blocks connected by control-flow edges.

A live variable analysis determines the input and output of each basic block. A basic block consists of three types of nodes: the merge node is responsible for aggregating data from previous nodes; the operational node represents load, store, and execute instructions; and the branch node selects one thread successor among many basic blocks.

After live variable analysis, the IR is optimized to generate a control-data flow graph (CDFG). RTL for each basic block is generated from the CDFG. An execution schedule is calculated for each node in the CDFG using the system

of difference constraints (SDC) scheduling algorithm. The scheduler uses linear equations to schedule instructions while minimizing the cost function.

The hardware logic generated by the kernel compiler for the nodes in the CDFG is added with stall, valid, and data signals that implement the control edges of the CDFG. The resultant Verilog code is wrapped with the communication (for example, PCIe) and memory infrastructure to interact with the host and the off-chip memory, respectively. A template-based design methodology is used where the application-independent communication and memory infrastructure is locked down as a static part of the system and only the application-dependent kernel part is synthesized. This hierarchical, partition-based approach reduces synthesis time. The design is then synthesized to generate a bitstream.

Altera also provides a host library consisting of APIs to bind the host function calls to the OpenCL kernels implemented in the FPGA. The host-side code written in C/C++ is linked to these libraries, enabling runtime communication and synchronization between the application code running on the host processor and the OpenCL kernels running on the FPGA. The host library also consists of the Auto-Discovery module, which allows the host program to query and detect the types of kernels running on the FPGA.

High-level language-based frameworks. Because of the popularity of modern programming languages, several recently proposed frameworks use high-level languages as starting points for generating hardware circuits. The languages in this category are highly abstract, usually object-oriented and offer high-level features such as polymorphism and automatic memory management. Some of the noteworthy languages and frameworks are Esterel,²³ Kiwi,¹² Chisel,⁴ and IBM's Liquid Metal.¹

Liquid Metal. The Liquid Metal project at IBM aims to provide not only high-level synthesis for FPGAs,¹ but also, more broadly, a single and unified language for programming heterogeneous architectures. The project offers a language called Lime,² which can be used to program hardware (FPGAs), as well as software running

on conventional CPUs and more exotic architectures such as GPUs.¹⁰ The Lime language was founded on principles that eschew the complicated program analysis that plagues C-based frameworks, while also offering powerful programming models that espouse the benefits of functional and stream-oriented programming.

Notable features of the Lime language include the ability to express bit literals and computation at the granularity of individual bits—just as in HDLs—but with the power of high-level abstractions and object-oriented programming. These permit, for example, the definition of generic classes that are parameterized by bit width, polymorphic methods and overloaded operators.

Lime is Java compatible and hence strongly typed. The language adds features to aid the compiler in deriving important properties for efficient behavioral synthesis into HDL. These include instantiation-based generics, bounded arrays, immutable types, and localized side effects that are guaranteed by the type-checker and exploited by the compiler for the purpose of generating efficient and pipelined circuits.

Another important feature of the Lime language is a task-based data-flow programming model that allows for program partitioning at task granularity, often resembling a block-level diagram of an architecture or FPGA circuit.

A task in Lime can be strongly isolated such that it cannot access a mutable

global program state (because it is prohibitively expensive in an FPGA), and no external references to the task exist so that its state cannot be mutated except from within the task, if at all. Task graphs result from connecting tasks together (using a first-class connect operator), so that the output of one becomes the input to another.

Figure 5 shows the Liquid Metal compilation and runtime architecture. The front-end compiler first translates the Lime program into an intermediate representation that describes the task graph. Each of the compiler backends is a vertically integrated tool chain that not only compiles the code for the intended architecture, but also generates an executable artifact suitable for the architecture. For example, the Verilog backend, which synthesizes HDL from Lime, will also automatically invoke the EDA tools for the target FPGA to perform logic synthesis and generate a bit file that can be loaded onto the FPGA. Each backend provides a set of exclusion rules that restrict the Lime language to a synthesizable subset. Some notable exclusions are the use of unbounded arrays and non-final classes. Dynamic memory allocation is restricted to final fields of objects, and recursive method calls may have no more than one argument. These rules are expected to change over time as the compilation technology matures.

Lime code is always executable on at least one architecture: the JVM (Java Virtual Machine). As such, func-

Figure 4. Altera OpenCL-to-FPGA framework.³

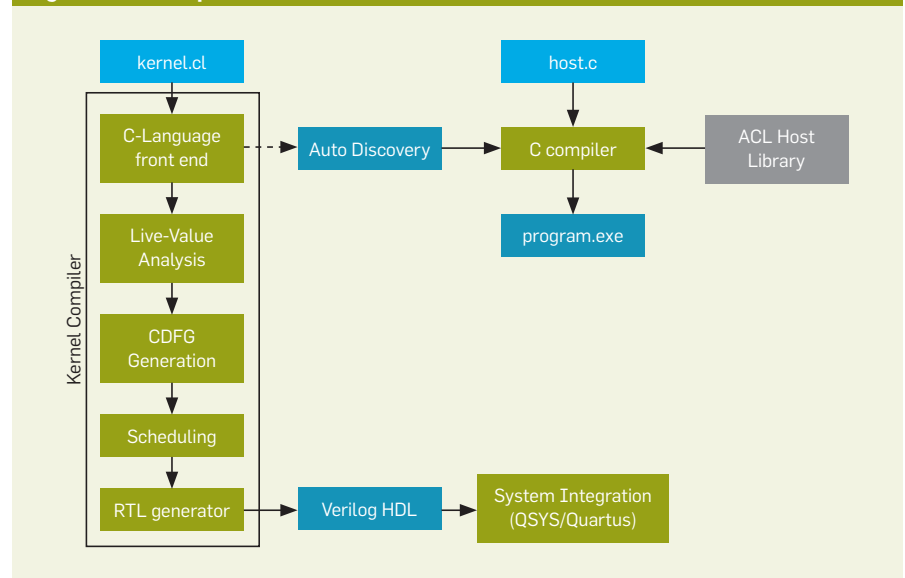
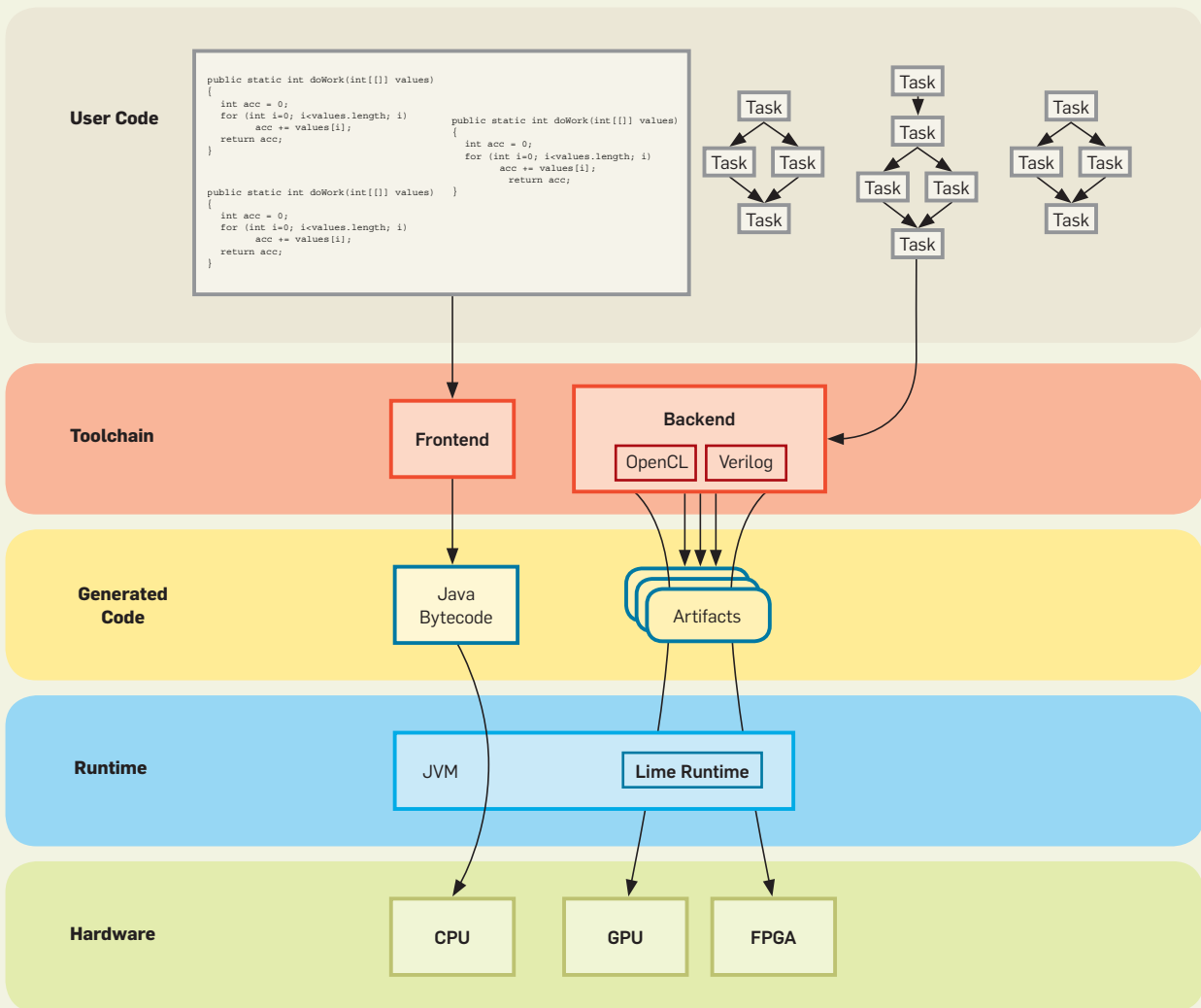


Figure 5. Liquid Metal compilation and runtime architecture.



tional verification of Lime programs may be carried out entirely in software using high-level debugging tools, including the Eclipse-based Lime IDE. Tasks that are mapped to the FPGA may co-execute with tasks mapped to other parts of the system (for example, GPU or JVM), and the communication throughout the system is orchestrated by the Lime runtime.

The language allows programmers to seamlessly integrate native HDL code into their Lime code via the Lime Native Interface. This allows the programmer to implement timing- and performance-critical components of the application in HDL and the rest of the application in Lime in order to meet performance specifica-

tions that may not be readily achievable through the current behavioral synthesis technology.

Model-based frameworks provide an abstract way of designing complex control- and signal-processing systems. They improve design quality and accelerate design and verification tasks by using an executable specification. The executable specification facilitates hardware-software partitioning, verification, and rapid design iterations. NI (National Instruments) LabView⁹ and Matlab HDL Coder¹⁷ are two Model-based frameworks. They are especially useful for designers with strong domain expertise but little experience with software/hardware programming languages.

NI LabView is a graphical programming and design language. Using functional blocks and interconnects, designers can create a graphical program that looks like their whiteboard designs. LabView provides an integrated environment that consists of a graphical editor for development, debugger, compilation framework, and middleware for execution on diverse targets such as desktop processors, PowerPC, ARM processors, and a number of FPGAs. With LabView, the user can start development and testing on one platform and then incrementally migrate to another platform.

Matlab HDL coder a high-level language and interactive environment for numerical and scientific computa-

tion and system modeling. It provides a number of tools and function-rich libraries that could be used for quickly implementing algorithms and model systems for a wide range of application domains. Simulink, a part of the Matlab product suite, provides a block-diagram-based graphical environment for system simulation and Model-based design. Simulink is basically a collection of libraries (toolboxes) for different application domains. HDL Coder is one such toolbox that generates synthesizable VHDL and Verilog code from Matlab functions and Simulink models. Users can model their systems in Simulink using library blocks and Matlab functions and then easily test the models against the functional specifications by creating a test bed using the same modular approach. After verification, users can generate bit-optimized and cycle-accurate synthesizable RTL code.

Future Directions

The era of frequency scaling in micro-processor design is largely believed to be over, and programmers are increasingly turning to heterogeneous architectures in search of better performance.

We have already started to see the integration of general-purpose processors and GPUs on the same die. The diversity is expected to increase in the future to include ASSPs, and even programmable logic (for example, FPGAs). The key is tighter integration of this diverse set of compute elements. The time is not far off when CPUs, GPUs, FPGAs, and other ASSPs will be integrated on the same chip. There are several standards and specifications for co-programming CPUs and GPUs, but FPGAs have been isolated so far, primarily because of the lack of support for device drivers, programming languages, and tools.

There is a serious need to improve the programming aspect of FPGAs if they are to join mainstream heterogeneous computing. Programming technology for FPGAs has lagged behind the advancements in semiconductor technology. As discussed here, HDLs are too low level. Significant design-cycle time must be spent in coding and verification. Design iterations are

quite expensive and prohibit thorough design-space exploration.

HLS presents a promising direction, however even after a few decades of research in the area, FPGA programming practices continue to be fragmented and challenging, placing a high-engineering burden on researchers and developers. Some problems arise because too many efforts are going on simultaneously without any standardization, yet none provide a comprehensive ecosystem to foster adoption and growth. Many of the ongoing endeavors are centered on C-based languages. Only time will tell if C is the right abstraction for the future since it was designed for a sequential execution model rooted in Von Neumann architectures and existing design trends suggest increasingly heterogeneous and parallel architectures.

We also need to revisit the approach of compiling a high-level language to VHDL/Verilog and then using vendor-synthesis tools to generate bitstreams. If FPGA vendors open up FPGA architecture details, third parties could develop new tools that compile a high-level description directly to a bitstream without going through the intermediate step of generating VHDL/Verilog. This is attractive because current synthesis times are too long to be acceptable in the mainstream. C

Related articles on queue.acm.org

Computing without Processors

Satnam Singh

<http://queue.acm.org/detail.cfm?id=2000516>

Of Processors and Processing

Gene Frantz, Ray Simar

<http://queue.acm.org/detail.cfm?id=984478>

Abstraction in Hardware System Design

Rishiyur S. Nikhil

<http://queue.acm.org/detail.cfm?id=2020861>

References

1. Auerbach, J., Bacon, D.F., Burcea, I., Cheng, P., Fink, S.J., Rabbah, R. and Shukla, S. A compiler and runtime for heterogeneous computing. In *Proceedings of the 49th ACM/EDAC/IEEE Design Automation Conference* (2012), 271–276.
2. Auerbach, J., Bacon, D.F., Cheng, P. and Rabbah, R. Lime: A Java-compatible and synthesizable language for heterogeneous architectures. In *Proceedings of the ACM International Conference on Object-oriented Programming Systems Languages and Applications* (2010), 89–108.
3. Aydonat, U., Denisenko, D., Freeman, J., Kinsner, M., Neto, D., Wong, J., Yiannacouras, P. and Singh, D.P. From OpenCL to high-performance hardware on FPGAs. In *Proceedings of the 22nd International*

Conference on Field Programmable Logic and Applications (2012), 531–534.

4. Bachrach, J., Richards, B., Vo, H., Lee, Y., Waterman, A., Avidienis, R., Wawrzyniec, J. and Asanovic, K. Chisel: Constructing hardware in a Scala-embedded language. In *Proceedings of the 49th ACM/EDAC/IEEE Design Automation Conference* (2012), 1212–1221.
5. Berkeley Design Technology. An independent evaluation of the AutoESL AutoPilot high-level synthesis tool. Technical Report, 2010.
6. Brodtkorb, A.R., Dyken, C., Hagen, T.R., Hjelmervik, J.M. and Storaasli, O.O. State-of-the-art in heterogeneous computing. *Scientific Programming* 18, 1 (2010).
7. Cardoso, J. and Diniz, P. *Compilation Techniques for Reconfigurable Architectures*. Springer, 2009.
8. Coussey, P. and Morawiec, A. *High-level Synthesis: From Algorithm to Digital Circuit*. Springer, 2008.
9. Dase, C., Falcon, J. S. and MacCleery, B. Motorcycle control prototyping using an FPGA-based embedded control system. *IEEE Control Systems* 26, 5 (2006), 17–21.
10. Dubach, C., Cheng, P., Rabbah, R., Bacon, D.F., Fink, S.J. Compiling a high-level language for GPUs: (via language support for architectures and compilers). In *33rd SIGPLAN Symposium for Programming Design and Implementation* (2012), 1–12.
11. Edwards, S.A. High-Level Synthesis from the Synchronous Language Esterel. In *IEEE/ACM International Workshop on Logic & Synthesis* (2002), 401–406.
12. Greaves, D. and Singh, S. Designing application-specific circuits with concurrent C# programs. In *Proceedings of the 8th ACM/IEEE International Conference on Formal Methods and Models for Code Design* (2010).
13. Jaaskelainen, P.O., de La Loma, C.S., Huerta, P., Takala, J.H. OpenCL-based design methodology for application-specific processors. *Embedded Computer Systems* (2010), 223–230.
14. Nikhil, R.S. Abstraction in hardware system design. *ACM Queue* 9, 8 (2011); <http://queue.acm.org/detail.cfm?id=2020861>.
15. Owaid, M., Bellas, N., Daloukas, K. and Antonopoulos, C. Synthesis of platform architectures from OpenCL programs. In *Field-programmable Custom Computing Machines* (2012), 186–193.
16. Papakonstantinou, A., Karthik, G., Stratton, J. A., Chen, D., Cong, J. and Hwu, W.-M.W. 2009. FCUDA: Enabling efficient compilation of CUDA kernels onto FPGAs. In *Application Specific Processors* (2009), 35–42.
17. Sharma, S. and Chen, W. Using Model-based design to accelerate FPGA development for automotive applications. The MathWorks, 2009.
18. Sirowy, S. and Forin, A. Where's the beef? Why FPGAs are so fast. Microsoft Research Technical Report MSR-TR-2008-130, 2008.
19. Thomas, D.B., Howes, L., Luk, W. A comparison of CPUs, GPUs, FPGAs, and massively parallel processor arrays for random number generation. In *ACM/SIGDA International Symposium on Field programmable Gate Arrays* (2009), 22–24.
20. WinterGreen Research Inc. Programmable logic IC market shares and forecasts, worldwide, 2010 to 2016. Technical report, 2010.
21. Wulf, W.A. and McKee, S.A. Hitting the memory wall: Implications of the obvious. *SIGARCH Computer Architecture News* 23, 1 (1995), 20–24.
22. Xilinx. Command line tools user guide. Technical Report UG628 (14.3), 2012.
23. Xilinx. 7 series FPGAs overview. Technical Report DS180 (1.13), 2012.

David Bacon is a research staff member at the IBM T.J. Watson Research Center. His research interests are in programming-language design and implementation, and he is currently working on the Liquid Metal project.

Rodric Rabbah is a research staff member at the IBM T.J. Watson Research Center. He is interested in programming languages, compilers, and architectures for heterogeneous computing. He has worked in these areas since the founding of the Liquid Metal project at IBM in 2007.

Sunil Shukla is a research staff member at the IBM T.J. Watson Research Center. He is currently working on the Liquid Metal project, which provides a single-language solution to programming heterogeneous architectures (CPU, GPU, FPGA).

DOI:10.1145/2436256.2436272

**Start with talent and skills driven
by curiosity and hormones, constrained
only by moral values and judgment.**

BY ZHENGCHUAN XU, QING HU, AND CHENGHONG ZHANG

Why Computer Talents Become Computer Hackers

THE EPIDEMIC OF computer hacking is a direct result of advances in computer-networking technologies like the Internet and the widespread use of computers throughout society, from personal entertainment to business transactions, social networking to scientific discovery, and managing personal lives to multinational organizations. Related illicit and often illegal activities have cost organizations and individuals billions of dollars directly and indirectly worldwide,¹⁷ with one estimate of \$5.5 million per organization in 2011.¹⁸ An interesting but troubling aspect of the epidemic

is that so much of it is committed by college-age young people.²⁸ In a 2006 study of college students in three U.S. universities, Cronan et al.⁶ reported that 34% of their respondents admitted to committing some form of software misuse or piracy and 22% to committing data misuse. How and why would talented, computer-savvy young people who might otherwise aspire to productive careers in the computer and IT professions evolve into computer hackers, even into criminals? Rigorous academic studies of hackers, especially those involving empirical evidence, are scarce in the literature, despite some notable exceptions (such as Bachmann,² Holt,^{11,12} Turgeman-Goldschmidt,²⁶ and Young et al.²⁹)

There are no consistent, widely accepted theories or theoretical frameworks in the literature as to why hackers emerge and evolve, and therefore no clear, effective guidance on what to do to prevent talented computer-savvy young people from becoming hackers or criminals. Here, we discuss our own study of six computer-hackers in China, addressing two main questions: How do hackers get started? and How and why do they evolve from innocent behavior (such as curious exploration of school computer systems) to criminal acts (such as stealing intellectual property)? Answers will help schools, universities, and society develop better policies and programs for addressing the phenomenon.

» key insights

- **Computer hackers start out not as delinquents or as social outcasts but often as talented students, curious, exploratory, respected, and, most important, fascinated by computers.**
- **Porous security, tolerance by teachers and school administrators, and association with like-minded individuals make for fertile ground in transforming young talents into hackers.**
- **Moral values and judgment appear to be the only reliable differentiator between gray hats and black hats.**

ILLUSTRATION BY ALICIA KUBISTA / ANDRTJ BORYS ASSOCIATES



Table 1. Primary criminological theories in hacker research.

Theory	Main Propositions	Main References
Rational Choice	- Individuals try to maximize expected value based on some utility function or scale when making decisions involving multiple options; - Individuals are able to rank-order the available options; preference orders are transitive; and - Individuals' decisions, rules, tastes, and preferences are relatively stable over time and similar among all people.	Green and Shapiro ⁹
Deterrence	- Individuals are fundamentally rational in their behavior, choosing crime only when it pays; and - Individuals are less likely to commit criminal acts if the perceived certainty, severity, and celerity of sanctions against the acts outweigh the expected gains.	Gibbs ⁷
Self-Control	- The primary difference between criminals and noncriminals is self-control; - An individual's self-control is established early in life, remaining relatively stable through the lifetime; and - Individuals with weak self-control tend to respond to tangible stimuli in the immediate environment and are more likely to be seduced by the thrill and excitement of criminal acts.	Gottfredson and Hirschi ⁸
Social Learning	- Criminal behavior is learned through an individual's association with criminals in personal and social groups; - Learning happens in social and nonsocial situations and includes techniques for committing crimes, as well as motives, drives, rationalization, attitudes, and reinforcements; and - The likelihood of an individual engaging in criminal behavior increases when choosing to differentially associate with criminals and imitate their behavior, when exposed to definitions (attitudes, norms, orientations) that justify or rationalize such behavior, and when, in the past, they had received differential reinforcement rewarding similar behavior.	Akers ¹ , Sutherland ²²
Neutralization	- Delinquent individuals are at least partly committed to the dominant social order in that they frequently exhibit guilt or shame when violating its proscriptions, accord approval to certain figures, and distinguish between appropriate and inappropriate targets for their deviance; - Delinquency is more likely by individuals who justify deviant acts seen as valid by the delinquent but not by the legal system or society at large, thus neutralizing or deflecting disapproval from internalized norms and conforming others in the social environment and rendering social controls inoperative; and - Five common neutralizations techniques are denial of responsibility, denial of injury, denial of victimhood, condemnation of the condemner, and appeal to higher loyalties.	Sykes and Matza ²³

Relevant Theories

Computer hacking is as old as digital technology but has not always had the negative connotations we see today. The term “hacker” was meant to describe a creative person who could alter computer programs and systems to do things beyond their inherent or intended design.²⁸ However, once the potentially destructive power of computer hacking was unleashed, there was no turning back. Computer hackers gradually separated into two camps—white hats and black hats—depending on motivation and objective. White hats are on a quest for knowledge, discovering and alerting security weaknesses in organizational systems and developing better, more secure computer systems; black hats go for revenge, sabotage, or outright criminal gain (such as to steal money, products, or services).¹⁹ In between are gray hats who hack for curiosity, fun, notoriety, or self-fulfillment but usually do not intend to harm their targets. Here, we focus on the gray and black hats, investigating their evolution.

Most academic research on computer hackers understandably takes a criminal view, using criminological

theories as the lens of analysis. Citing the research literature, Yar²⁸ in 2005 attributed two primary causes to the “youth problem” in hacking, as hackers tend to be young males and school dropouts in their mid-20s. The first is adolescence as a period of inevitable psychological turmoil, helping account for youthful participation in various forms of “delinquent” and “antisocial” behavior. The second is the apparent “ethical deficit” among adolescents disposing them toward law- and rule-breaking behavior. This argument is consistent with developmental psychology theory,¹⁰ which says juveniles, moving from childhood to adulthood, pass through stages of moral learning before “maturity” when they are finally able to fully appreciate and apply moral principles to regulate their own and others' behavior; juveniles are thus more likely to act on their hedonistic impulses with limited regard for their effect on others.

While the youth-moral-delinquency perspective attributes the root causes of hacker behavior among young people to social development and transition from adolescence to

adulthood, it is inadequate for explaining why only a small percentage of the youth population is involved in hacking and related deviant behavior. Scholars have begun to examine the role of personal character in and propensity toward deviant behavior, as in computer hacking and abuse.^{2,13,14} Holt et al.¹³ and Bossler and Burruss⁴ focused on testing the applicability of a widely accepted criminological theory—self-control, by Gottfredson and Hirschi⁸—to explain and predict hacking behavior. Scholars also use social learning theory,¹ another widely supported theory in the criminological literature, to study hacking behavior alone^{14,20} or in conjunction with self-control theory.^{4,14} In addition to self-control theory and social-learning theory, various theoretical lenses have also been used in the study of computer hacking. Table 1 outlines five primary criminological theories used in hacker studies; Table 2 highlights representative studies on hacking and hacker behavior.

While the studies have produced compelling evidence of and insight into computer hackers and their behavior, they are constrained by the

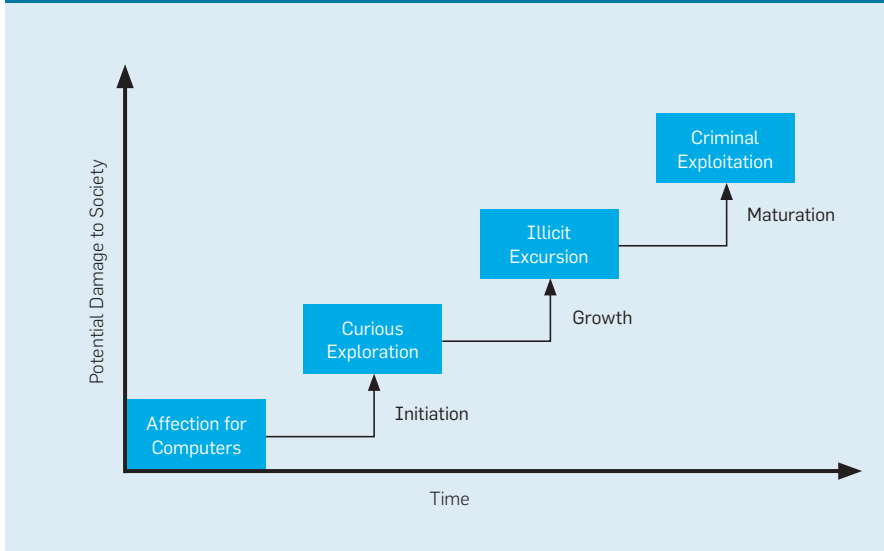
theoretical lenses and the perspectives used to examine the hacking phenomenon. As a result, some significant theoretical gaps persist in the literature regarding hacking. To better understand the gaps, we outline an evolutionary path taken by computer hackers (see Figure 1) based on findings in the literature, our own knowledge of computer hacking, and the evidence we gathered from our six research subjects.

Published studies focus primarily on the middle stage—growth—of the evolutionary path of computer hackers, in which hackers organize into loosely connected groups and virtual or real communities, acquire technical skills through mentoring and sharing, and establish social orders, group norms, and individual and social identities. We thus have a fairly good understanding of who hackers are and what they do and why they do

it. However, little research has targeted the first and last stages—initiation and maturation—of the evolutionary path, leaving many questions unanswered or with no clear answers, including: How and why do certain talented young people evolve into pathological computer hackers? and How and why do certain computer hackers become computer criminals? A comprehensive understanding of all stages of the evolutionary

Table 2. Representative academic studies on hacking behavior.

Study	Theoretical Lens	Methodology and Sample	Main Findings
Social learning and computer crime ²⁰	Social learning theory	▶ Quantitative ▶ Linear regression (OLS) ▶ Survey of college students	▶ Computer crimes (such as software piracy, password guessing, and illegal access to files) are prevalent among college students; and ▶ The four basic elements in social learning theory—differential association, imitation, definition, and reinforcement—are strong predictors of computer crimes.
Sociology of hackers ¹⁵	Imagined community	▶ Qualitative ▶ Literature review	▶ Hackers operate in an “imagined community” characterized by comfort with technology, ambivalence to secrecy, complex interplay between anonymity and identity, fluid membership and associations, male dominance, and non-malicious motivations; and ▶ The boundaries between the hacker community and the computer-security community are not defined by what they do but by the meanings of their actions.
Minds of hackers ²⁹	Moral disengagement and deterrence theories	▶ Quantitative ▶ Survey of hackers and non-hacker groups ▶ Analysis of Variance (ANOVA)	▶ Hackers have a significantly higher level of moral disengagement than other groups, perceiving hacking as acceptable as long as it does no harm; ▶ Hackers perceive a significantly lower level of informal sanctions against hacking and significantly less likelihood of being caught; and ▶ Hackers engage in hacking because perceived gains exceed perceived costs, expecting significantly higher utility value from hacking.
Structure of the hacker community ¹¹	Social organization theory	▶ Qualitative ▶ Interviews with hackers ▶ Forum postings ▶ Observations	▶ Hackers have personal and social relationships, though ties are neither deep nor strong; ▶ Most hackers act alone, with limited evidence of teamwork and division of labor; and ▶ With no evidence of groups with extensive history, hackers act as colleagues, rather than as peers, teams, or formal organizations.
Hacker attack dynamics ¹²	Behavior motivated by politics and religion	▶ Qualitative ▶ Interviews with hackers ▶ Forum postings	▶ Hackers gain knowledge and skills through individual learning and peer mentoring in hacker communities; ▶ Hackers are mission driven, with attacks against targets often based on religious and national beliefs.
Justification of hacker behavior ²⁶	Neutralization theory	▶ Qualitative ▶ Interviews with hackers	▶ Hackers imagine they are heroes, viewing themselves as positive deviants; ▶ Hackers justify their actions through neutralization techniques: denial of injury, denial of victimhood, condemnation of the condemners, appeal to higher loyalties, and a sense of self-fulfillment; and ▶ Two external neutralization techniques notably not used by hackers are denial of responsibility and the sad tale, differentiating hackers from other deviants in society.
How personality affects hacking ²	Rational choice and self-control theories	▶ Quantitative ▶ Survey of hackers ▶ Linear regression (OLS)	▶ Hackers with higher risk propensity and rationality tend to engage in more attack activity; and ▶ Lower risk propensity and higher rationality are associated with a higher level of attack success.
How social learning affects cyber-deviance ¹³	Social learning theory	▶ Quantitative ▶ Survey of college of students ▶ Structural equation model (SEM)	▶ Social learning is modeled as a second-order construct with four first-order components: differential association, definition, reinforcement, and imitation; ▶ Social learning mediates the effects of race, gender, and computer skill on cyber-deviance; and ▶ Those less likely to engage in deviant social learning process are less likely to commit deviant computer acts.
Self-control and computer hacking ⁴	Self-control and social learning theory	▶ Quantitative ▶ Survey of college of students ▶ Structural equation model (SEM)	▶ Weak self-control and social learning are modeled as single second-order constructs; ▶ Social learning is a stronger predictor of hacking behavior than weak self-control; and ▶ Weak self-control contributes to hacking through social learning; those with weak self-control are more likely to participate in the hacker social-learning process.

Figure 1. Evolutionary path taken by hackers.

process is critical for effectively managing the hacking epidemic and how it can cause significant harm to individuals and organizations worldwide. Targeted remedies for the initiation stage could prevent young computer talents from becoming illicit hackers, and effective intervention in the maturation stage could redirect hackers to more productive use of their knowledge and skills.

Here, we report the findings of an exploratory case study we conducted from December 2009 to March 2010 involving six young hackers in China, hoping to shed light on the evolutionary paths hackers generally take. Our findings provide insight into how to guide and shape young, talented, yet highly malleable, individuals toward productive careers in computing and IT, instead of a treacherous path toward computer hacking and criminal behavior.

Case Study

In attempting to develop a better framework for understanding and managing hacker behavior among young people, we faced two significant challenges: On the one hand, there is a rich body of qualitative discussions about the technical, sociological, psychology, and cultural origins of computer hacking from various perspectives;^{15,16,19,24,25} on the other, the extant quantitative studies seem to have produced findings that are more diverse than congruent due to their differing theoretical

perspectives.^{2,4,13,14,20,29} These challenges motivated us to conduct our own exploratory case study of computer hackers to address some of the critical elements not previously addressed. Our basic approach was to be informed by the extant literature but not constrained by the frameworks or theories. The only limit we adhered to was established methods of case-study research (see the online Appendix). We describe and discuss the most significant findings next, using pseudonyms for our six subjects to protect their identities.

Early Interest in Computers

In all but one case, our subjects developed an interest in computers early in life, some as early as elementary school, usually from ordinary circumstances (such as curiosity about how computers work and playing computer games), as with many other teenagers. “Adam” said his hacking began when he was in high school and took courses in computer programming. Though not very interested in computer games, he liked to disassemble and reassemble computers at home. “Eric” said he became interested in computers in third grade when his family bought its first computer. His primary interest then was playing computer games. Likewise, when describing how he began hacking, “Frank” said, “I got my first personal computer in the 1990s when I was in middle school. I bought the computer with antivirus software. I was curious

how the software worked. So, in my first year of middle school, I was able to break into the software and understand how it worked.”

Innocent Motives

Our interviews found these students typically started hacking due to innocent motives (such as wanting to know more about computers and going online with school computers). “Chris” said his first hacking followed his interest in a female student as a freshman in high school. He was too shy to ask where she lived, thinking the information must be stored in the school’s registration system. He then learned hacking techniques from computer magazines and multiple sources on the Internet. After a semester of trial and error, he gained access to the school registration system and quickly found the information he was looking for.

Chris’s story is typical of our subjects. “Brian” said his earliest hacking experience was in middle school when he wanted to continue playing computer games on the Internet and the teacher in charge of the computer lab cut access to the router to control access time. That motivated him to learn how to turn on access remotely so he could continue playing when the teacher would leave. Eric began hacking his own computer and computer games when he, too, was in middle school, modifying the computer and the games to install new games and give himself a better game experience, sometimes altering the balance in his online gaming accounts to be able to play without paying.

Minds Not Challenged

It appeared that all our subjects were exceptionally bright compared to their student peers and could have chosen to be the kind of A-students their teachers and parents expected and hoped for. Interestingly, as students with great academic potential, all appeared *uninterested* in being A-students, preferring to spend their time learning hacking skills instead of doing their coursework. Adam said, “Among my friends in college, none of them have good grades. While we were the smartest kids in high school, there is not much difference among

the classmates in terms of intelligence at the top colleges. Some students want to devote most of their time to academic studies; others like to spend time on more interesting things. I spend a lot of time in labs on complex computer networks, which have little relationship with my major. Many of the courses in my college curriculum are not very meaningful to me, so I don't have any motivation to achieve A's in these courses. Compared to other students, students of this type have stronger technical skills, spend less time on courses, and have more time to kill. We want to be different, have an interesting life, and develop unique characters. I use the time to hack and develop hacking tools, while among my friends, there is a variety of other interests. Some participate in student clubs (such as the debate club)."

Frank said his childhood dream was time travel and studying high-energy physics but ended up as an electrical engineering major in college. He was not interested in topics like magnetic fields taught in class, spending most of his time studying computer programming languages and learning computer hacking skills. He said, "I read through all of the books about computer hacking on the second floor of the college's library, as well as books on computer programming, during my first year in college. I learned every computer programming language I had access to."

Porous Security

The convergence of computers and networks in homes, schools, and organizations, along with connectivity provided by the Internet, poor-quality security mechanisms in major operating systems and application software, and Web servers based on the TCP/IP protocols was fertile ground for the talented and the curious to explore and exploit. Our six subjects demonstrated that with some fundamental understanding of computer programming and network protocols, along with tips and techniques from computer magazines and the Internet, they could penetrate almost any computer system, viewing and downloading documents at will, while still in high school.



The only thing separating him from being a computer hacker (gray hat) and being a computer criminal (black hat) is his moral values and judgment regarding such behavior.



They quickly discovered the situation was no better in college. Chris said, "When I arrived at this university, I went into the computer labs and tried to figure out if there were any security holes in the systems. Unfortunately, I found a lot. Unlike others, my interest was not in individual computers but in servers. Most of the individual computers had virtually no protection. I liked to be challenged with technical issues, so I targeted only servers and tried various approaches to penetrate them."

Similar discoveries were made by the others as well. Adam said, "In my first year at the university, I discovered there was a system for admissions. I was curious about whether a girl in my high school was admitted into the university, so I attempted to hack the system. I found the security of the system was very weak; a simple SQL injection allowed me to break in. I could have easily changed the admission records or the registration records."

Tolerated by Schools

The study subjects' participation in regional and national computer programming competitions brought accolades to their schools. They thus received special treatment and respect from their teachers and school officials. Asked whether they feared being caught and how it might affect admission to the top universities they sought, Adam said, "The teacher who was managing the systems was like a brother to us. Even if we were caught, he wouldn't punish us. We were winning prizes for him and the school, and he couldn't be more thankful to us."

Although clear that not all school computer administrators are indifferent to hacking, our evidence shows our student hackers were usually able to mend the relationship to avoid punishment after their hacking was exposed. Chris said, "There are two types of attitude toward student hackers in my school: One group thought we were troublemakers, and the other was more accommodating and admiring what we could do because they had some interest as well. Once you break into a system, the system administrators usually dislike you because you have made them look bad, because their job is to protect the sys-

tem. But I eventually helped them redesign the security of the system, and we were safe after that.”

Associated with Other Hackers

Our subjects said at some point in their hacking histories, they connected with others with like interests, significantly accelerating development of their skills and scope. Hackers and potential hackers seek each other through the Internet and online communities (such as QQ, a popular instant-messaging and online community platform in China) and college bulletin-board system sites, forming their own cliques and communities, sharing experiences, tools, and skills, and occasionally bragging about their accomplishments. Adam said his connections with peer hackers developed when he was accepted to a university and students from all over the country joined the QQ group to exchange ideas, learn skills, and even organize coordinated attacks on targets.

In some cases, student hackers would openly organize themselves into student clubs or special interest groups. Frank said, “When I was in college in 2006, I started a network security club consisting of only students, many of whom were victims of computer hacking (such as some who had their QQ passwords stolen or their computers invaded by Trojan malware). Some were just curious; others were interested in knowing how to steal QQ passwords. Gradually, some of the students started to steal exam files from professors’ computers. Students in the club taught each other hacking skills. After I graduated, I went back to give lectures to the club, sometimes with 300–400 in attendance, covering topics ranging from how to identify and take advantage of security holes to how to defend against security attacks. We even set up a mobile server and let students compete to see who could get maximum control over it and grant themselves the most permission on the server.”

Shifting Moral Values

Our subjects indicated that many college students were involved in computer hacking, though only a small

They are aided early on by tolerance and even reinforcement by parents, teachers, and school administrators, and later by sophisticated social networks and cliques.

number ever become hackers who commit crimes using their skills, in college or after graduation. Most will find jobs in top-tier IT companies and information-security firms. Frank said, “Many of the students in the hacking club went on to work for top IT companies like Baidu, Tencent, and Symantec. These students came from all majors, including management, foreign languages, and transportation engineering, and knew the information-security industry offered higher pay.”

There is no guarantee our subjects, as students or as future employees, would not continue to use their increasingly sophisticated hacking skills to do harm. The primary constraining factor seems to be their moral values and judgment about hacking. All insisted they had drawn a line they would not cross—do no harm to others. Brian described an episode in which he gained remote access to a teacher’s desktop computer and found a document with his family’s credit-card and bank-account information. He said he felt badly for the teacher for his poor awareness of computer security but did not take money from his accounts. Asked why not, he said, “My education from a very young age has been that it is shameful to take something without working for it. Fundamentally, I believe I can tell right from wrong. I did this because I just wanted to practice as a case study of what I read from a [hacking] guide. I have always been mindful of my moral bottom line.” However, a few of our subjects acknowledged they might cross such a line under certain circumstances (such as for survival and for justice, not a very high bar in today’s material world). Asked whether he had ever used his hacking skills to make a profit, Frank said, “I will not deny that I have sold the security holes I identified for money. But my basic principle is I will absolutely not sell the holes to individuals.”

Likewise, Chris said he learned when he was young it is morally acceptable to benefit oneself as long as he did no harm to others. He said, “I often see in QQ groups that individuals are selling botnets (zombie net-

works with hundreds or thousands of computers infected with Trojan-horse malware enabling control of the computers by a single perpetrator) they controlled for money. I think that's a problem. This is like you are attacking an individual who has no means to protect himself. I feel that a true hacker should have some moral principles, that is, do not attack individuals' computers. I enjoy getting data from servers, but I won't alter or destroy data on the servers." However, when asked whether he would deviate from his principles if he were unemployed and needed money to pay rent or buy food and someone was offering to buy control over botnets, he said, "I feel that is entirely possible."

Discussion and Insight

The evidence we found that a perspective involving moral delinquency among young people does not adequately explain how our subjects became who they are today and why they do what they do. None were delinquent in many aspects of their adolescence nor did they appear to struggle with moral confusion or disengage-

ment. On the contrary, all were viewed as outstanding students and treated with respect by their teachers. Their successful admissions into China's top universities represented the strongest manifestation of their academic success in high school.

Evidence of low self-control is mixed. While two subjects said they did not think much about long-term goals, the other four seemed thoughtful and goal-oriented. Their pursuit of increasingly complex hacking skills and control over what to target based on their own moral values appears inconsistent with the predictions of self-control theory.

On the other hand, we found many consistencies between what we learned and what is described in routine activity theory⁵ (RAT), social learning theory¹ (SLT), and situational action theory²⁷ (SAT). We submit that these theories together capture the essence of our main findings: how computer hacking emerges in young people; why talented computer students become hackers; and how gray hats become black hats. Here, we discuss these findings in light of the theories,

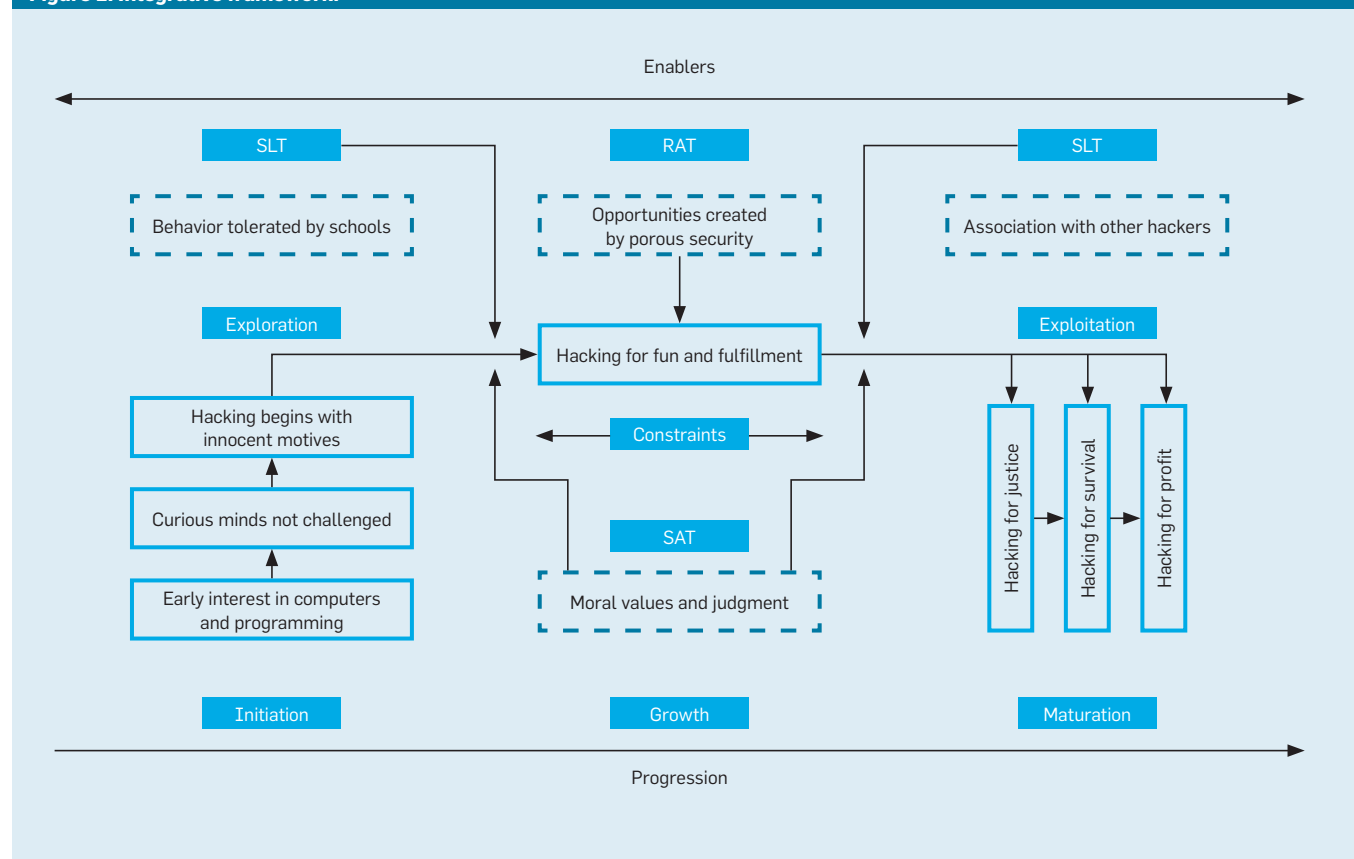
proposing our own framework based on our findings.

How Does Hacking Emerge?

RAT informs us that for a crime to occur, three essential elements must converge in time and space—motivated offender; appropriate target; and absence of able guardians⁵—offering an adequate explanation of how computer hacking began among our subjects. The emergence of a middle class in China following significant economic development over the past three decades, coupled with the dramatic price decrease of personal computers and penetration of Internet connectivity, has enabled many millions of families to buy computers for their children and to provide their (usually) only children their own rooms. This socioeconomic environment created conditions similar to the emerging suburbanization around major U.S. cities in the 1950s, from which Cohen and Felson⁵ developed RAT of crime.

Although our subjects were not necessarily committing criminal offenses when breaking into their schools' computer labs or library systems, a logi-

Figure 2. Integrative framework.



cal analogy can be made between the behavior of middle and high school students and home invasions and robberies committed by street criminals—the convergence in time and space of the three essential elements of computer hacking: a motivated young adult with the ability (talent and skills driven by curiosity and hormones); an attractive target (computer systems storing information they want or providing access to the Internet or computer games they want); and the absence of able guardians (innate security holes in many operating systems and application software, weak security protection provided by their owners, and basic tolerance by school authorities to hacking).

Hacker Evolution

Although RAT provides a reasonable explanation for how computer hacking emerged in our subjects, it cannot explain why innocent, curiosity-driven computer hacking evolves into more sophisticated, mission-oriented hacking. For hacking to occur, all three elements identified in RAT must be present, but they are only the necessary conditions. Imagine a young male college student wants to know which classes a particular female student has registered for, a common scenario described in our interviews. He is aware of the registration system where such information is located and has the skills to break into the system and access it. However, he also has other alternatives for getting the information (such as asking the student directly, following her around, or asking her friends), which may take more or less effort and be more or less risky than hacking into the registration system. Why he chooses one over the others is beyond RAT.

This is where SAT becomes salient, informing us that people are moved to action by how they view their options when confronting a particular situation; what they see and what they choose depend on their knowledge and skill, experience, morality, opportunity, and moral context.²⁷ In the scenario of a boy following a girl our subject would most likely choose the hacking option for four reasons: his knowledge and skill concerning the computer system; his past success

hacking with little or no adverse consequence; hacking as the easiest option compared to the alternatives; and his moral judgment telling him it is not wrong if he intends to only *peek* at the records.

Imagine this young man graduates and finds a job that pays a good salary. He needs more money to support his increasingly demanding lifestyle, but the salary alone is not sufficient. He now sees offers on an underground website to buy controls for botnets, with the price dependent on the number of computers in a particular botnet. He has the skills to quickly infect thousands of computers, he might have done it in the past just to see if he could, and he has access to the population of computers that are easy targets for such an operation. The only thing now separating him from being a computer hacker (gray hat) and being a computer criminal (black hat) is his moral values and judgment regarding such behavior. SAT rightfully focuses on the moral judgment and context of the particular setting as the lens through which the subject views his options.

Aiding the Transition

While RAT and SAT are insightful theories to help explain the emergence and transition of computer hackers, their focus is mainly on individuals, in this case the lone hacker, in terms of what they see, how they evaluate their situations, and why they take certain actions. However, as suggested by our case evidence and the literature, though hackers may act alone most of the time, their evolutionary processes almost never happen in isolation but are fostered and sustained by salient elements in their social environments. They are aided early on by tolerance and even reinforcement by parents, teachers, and school administrators, and later by the sophisticated social networks and cliques from which they learn and share techniques, brag of accomplishments, form social orders and identities, and perpetuate the hacking subculture.^{11,12,15,26} This is where SLT offers a salient explanation. Developed as a general theory to explain criminal behavior, SLT maintains that the probability an individual engages in crimi-

nal behavior will increase when the individual chooses to associate with others who commit criminal behavior and imitate their actions, is exposed to definitions (attitudes, norms, and orientations) that justify or rationalize the behavior, and has previously received differential reinforcement rewarding similar behavior.¹

Imagine a young hacker beginning to explore the weaknesses in school computers and networks in order to look at student records or extend the time he is able to access the Internet. Had teachers and school administrators not tolerated such behavior, enforced disciplinary measures, and provided venues to challenge his mind and instill the right ethical and moral values, he would have likely become a talented computer science or engineering college student like many others. Now he is in college armed with computer skills, motivated by curiosity, and emboldened by past experience, finding himself surrounded by like-minded and equally or even more talented peers, along with social groups and cliques, social networking tools that diminish physical distance and amplify the sense of community, and seminars and clubs that appeal to every interest. Immersed in this environment, he faces the challenge of differential association, or the individuals or groups he will identify with. Such groups or cliques will have a significant influence on what he does and how he views the world and computer hacking. The six subjects in our study and those studied in the literature^{11,12,15,26} decided to associate themselves with hacker groups and communities for acceptance, learning, support, and identity. Their evolutionary paths from talents to hackers illuminate the hallmarks of SLT.

A Framework for Understanding Hacking

While routine RAT, SLT, and SAT offer insight into most of what we observed about computer hackers, an integrative view of the evolutionary process of hacker motivation, skill, experience, moral values, and behavior is clearly in order. As the evidence shows, our six subjects began hacking due to essentially innocent

motivations. However, they gradually transitioned into more serious ones when they went to college and fell in with other hackers.

Likewise, their knowledge and skills improved in college due to social networking tools and student clubs connecting talented, like-minded students, enabling them to learn from one another, share their experiences, and exchange ideas. Perhaps most important, their moral values were also evolving. Hacking high school computer systems was mostly for fun, spurred by curiosity. But when hacking university registration and admission systems, professors' computers, and foreign government and military systems, the motivation was much less innocent and the consequences much more serious. In such circumstances, their moral values and judgment played a significant role in regulating their behavior.

Since they were rarely caught and disciplined, they formed the moral value that as long as they do no harm to others, it is not wrong to benefit themselves. On the rare occasions they were caught, they were enlisted to help the schools identify weaknesses in their systems, reinforcing the value that it is not only okay to hack computer systems, it may be justified if it helps the targeted organization improve its security. However, the moral bottom line—do no harm—may be less constraining than our subjects reported. When a situation is viewed as doing justice or taking revenge (such as in the Chinese-American cyber war in 2001^{3,21}), the line was readily crossed by at least one of our subjects, with others indicating they, too, would have participated without hesitation. Likewise, when survival is at stake, the guideline of do no harm would impose little moral constraint on our subjects, based on what they said to us.

In order to incorporate the adaptive and evolutionary nature of hacker motivation, skill, moral values, and behavior into RAT, SLT, and SAT, we propose an integrative process framework to explain the evolution of computer hacking among young people (see Figure 2). In it, the three stages of hacker evolution in Figure 1 are further supported through specific ac-



Since they were rarely caught and disciplined, they formed the moral value that as long as they do no harm to others, it is not wrong to benefit themselves.



tivities, enablers, and constraints. We also marked the locations where each of the three main theories—RAT, SLT, and SAT—is most salient in explaining the dynamics of hacker evolution.

We submit that the transition from innocent young talent to exploitive hacker begins with benign motivations (such as interest in computers, curiosity about people, and inner drive for knowledge and skill). Aided by three external enablers—bountiful opportunity due to porous security in computer systems and applications, tolerance of hacking by schools, and association with other hackers—and constrained primarily by moral values and judgment about computer hacking, such young talents gradually transition from curious exploration to purposeful exploitation. It must be pointed out that the constraining effect of moral values and judgment can be weak or strong and transitory, depending on the individual and his shifting values, as suggested by our case evidence.

The value of this framework is twofold: improve our understanding of how talented young people become hackers from an evolutionary perspective, and, perhaps more important, provide guidance as to how the hacking epidemic among talented young people could be better managed. Schools and universities can do little about initial motivation (mostly legitimate and normal) and behavior (usually too late to change). Likewise, schools and universities can do little about opportunities due to porous computer security, as well as the ability to associate with and learn from other hackers in today's social networking environment.

However, schools, universities, and society in general can manage two critical enablers—tolerance and shifting moral values—to tame hacking. The moral values and judgment involved in computer hacking are shaped in part by the attitudes and actions of schools and universities toward hacking, as shown by our case evidence. So if schools and universities adopt an attitude of zero tolerance toward hacking, along with early intervention to address identified hacking activity (such as offering courses in computer ethics, organizing competitions in-

volving defense of computer security, and setting up computer security services for organizations), these students might develop stronger moral values against illicit hacking, significantly influencing their later behavior in college. Eliminating tolerance and strengthening moral-value constraint appear to be the only manageable options in resisting hacking today. SAT and SLT would provide insight for policymakers at multiple levels.

Conclusion

We investigated how and why talented, computer-savvy young people become computer hackers through a case-study approach based on interviews with six known computer hackers in China. While RAT helps explain how hacking begins, SAT explains why talented young people take the road toward computer hacking, even when presented with many alternatives, and SLT calls for attention to environments that sustain hacking behavior and subculture. However, none of these theories explains the evolution of certain critical elements in the hacker process: motivation, knowledge and skill, opportunity, moral values and judgment, and the environment.

Based on our case evidence and the literature, we developed a framework for understanding and managing hackers and hacking behavior from an evolutionary perspective. The framework's most significant contribution is its explication of the enablers and constraints influencing hackers, providing guidance for managing the hacking epidemic by schools, universities, and throughout society. This framework calls for zero tolerance for hacking in schools and early intervention (such as through courses in computer ethics in middle and high schools, supervised competitions in defending computer security, and organizing computer security services for organizations) to strengthen the moral values of students against hacking and channel their interest in computers in a positive direction.

We also note a few caveats that may limit the generalizability of our findings and recommendations. The hacker subjects in our study were all from China. While this fills a sig-

nificant gap in the hacking literature, some factors may be unique to the cultural and economic context; for instance, tolerance of student hacking by teachers and school administrators that appeared prevalent in our cases may not be the same in other countries; and the fact that all our subjects were the only children in their families due to China's birth-control policies may have some influence on their desire, ability, and ways of socializing with their peers and social groups, as well as on their character development related to self-control, moral values, and other relevant personality traits. Overall, however, their profiles, activities, and evolutionary paths are fairly congruent with what has been reported in the hacker research literature based primarily on Western cultures and countries.

Acknowledgment

This research is supported in part by grants from the National Natural Science Foundation of China (71272076 and 70972048).

References

1. Akers, R.L. *Social Learning and Social Structure: A General Theory of Crime and Deviance*. Northeastern University Press, Boston, 1998.
2. Bachmann, M. The risk propensity and rationality of computer hackers. *International Journal of Cyber Criminology* 4, 1–2 (combined issue, Jan.–July and July–Dec. 2010), 643–656.
3. Becker, E. F.B.I. warns that Chinese may disrupt U.S. Web sites. *The New York Times* (Apr. 28, 2001); <http://www.nytimes.com/2001/04/28/world/fbi-warns-that-chinese-may-disrupt-us-web-sites.html?src=pm>
4. Bossler, A.M. and Burruss, G.W. The general theory of crime and computer hacking: Low self-control hackers? In *Corporate Hacking and Technology-Driven Crime: Social Dynamics and Implications*, T.J. Holt and B. H. Schell, Eds. Information Science Reference, Hershey, PA, 2011, 38–67.
5. Cohen, L.E. and Felson, M. Social change and crime rate trends: A routine activity approach. *American Sociological Review* 44, 4 (Aug. 1979), 588–608.
6. Cronan, T.P., Foltz, C.B., and Jones, T.W. Piracy, computer crime, and IS misuse at the university. *Commun. ACM* 49, 6 (June 2006), 84–90.
7. Gibbs, J.P. *Crime, Punishment, and Deterrence*. Elsevier, New York, 1975.
8. Gottfredson, M. and Hirschi, T. *A General Theory of Crime*. Stanford University Press, Stanford, CA, 1990.
9. Green, D.P. and Shapiro, I. *Pathologies of Rational Choice Theory: A Critique of Applications in Political Science*. Yale University Press, New Haven and London, 1994.
10. Hollin, C. Criminological psychology. In *The Oxford Handbook of Criminology*, M. Maguire, R. Morgan, and R. Reiner, Eds. Oxford University Press, Oxford, U.K., 2002.
11. Holt, T.J. Lone hacker or group cracks: Examining the social organization of computer hackers. In *Crimes of the Internet*, F. Schmallenger and M. Pittaro, Eds. Pearson, Upper Saddle River, NJ, 2009, 336–355.
12. Holt, T.J. The attack dynamics of political and religiously motivated hackers. In *Cyber Infrastructure Protection*, T. Saadawi and L. Jordan, Eds. Strategic Studies Institute, New York, 2009, 161–182.
13. Holt, T.J., Bossler, A.M., and May, D.C. Low self-control, deviant peer associations, and juvenile cyberdeviance. *American Journal of Criminal Justice*

- 37, 3 (Sept. 2012), 378–395.
14. Holt, T.J., Burruss, G.W., and Bossler, A.M. Social learning and cyber-deviance: Examining the importance of a full social learning model in the virtual world. *Journal of Crime and Justice* 33, 2 (2010), 31–61.
15. Jordan, T. and Taylor, P.A. Sociology of hackers. *The Sociological Review* 46, 4 (Nov. 1998), 757–780.
16. Jordan, T. and Taylor, P. *Hactivism and Cyber Wars*. Routledge, London, 2004.
17. Mercuri, R.T. Analyzing security costs. *Commun. ACM* 46, 6 (June 2003), 15–18.
18. Ponemon Institute. *2011 Cost of Data Breach Study*. Ponemon Institute LLC, Traverse City, MI, Mar. 2012; http://www.symantec.com/content/en/us/about/media/pdfs/b-ponemon-2011-cost-of-data-breach-us-en-us.pdf?om_ext_cid=biz_socmed_twitter_facebook_marketwire_linkedln_2012Mar_worldwide_CODB_US
19. Schell, B.H., Dodge, J.L., and Moutsatsos, S.S. *The Hacking of America: Who's Doing It, Why, and How*. Quorum Books, Westport, CT, 2002.
20. Skinner, W.F. and Frearm, A.M. A social learning theory analysis of computer crime among college students. *Journal of Research in Crime and Delinquency* 34, 4 (Nov. 1997), 495–518.
21. Smith, C. The first world hacker war. *The New York Times* (May 13, 2001); <http://www.nytimes.com/2001/05/13/weekinreview/may-6-12-the-first-world-hacker-war.html>
22. Sutherland, E.H. *Principles of Criminology*. J.B. Lippincott, Philadelphia, 1947.
23. Sykes, G.M. and Matza, D. Techniques of neutralization: A theory of delinquency. *American Sociological Review* 22, 6 (Dec. 1957), 664–670.
24. Taylor, P. *Hackers: Crime in the Digital Sublime*. Routledge, London, 1999.
25. Thomas, D. *Hacker Culture*. University of Minnesota Press, Minneapolis, 2002.
26. Turgeman-Goldschmidt, O. The rhetoric of hackers' neutralizations. In *Crimes of the Internet*, F. Schmallenger and M. Pittaro, Eds. Pearson, Upper Saddle River, NJ, 2009, 317–335.
27. Wikström, P.H. Linking individual, setting, and acts of crime: Situational mechanisms and the explanation of crime. In *The Explanation of Crime: Contexts, Mechanisms, and Development*, P.H. Wikström and R.J. Sampson, Eds. Cambridge University Press, Cambridge, U.K. 2006.
28. Yar, M. Computer hacking: Just another case of juvenile delinquency? *The Howard Journal of Criminal Justice* 44, 4 (Sept. 2005), 387–399.
29. Young, R., Zhang, L., and Prybutok, V.R. Hacking into the minds of hackers. *Information Systems Management* 24, 4 (Dec. 2007), 281–287.

Zhengchuan Xu (zcxu@fudan.edu.cn) is an associate professor in the Department of Information Management and Information Systems in the School of Management at Fudan University, Shanghai, China.

Qing Hu (qinghu@iastate.edu) is Associate Dean for Graduate Programs and Union Pacific Professor in Information Systems in the College of Business at Iowa State University, Ames, IA.

Chenghong Zhang (chzhang@fudan.edu.cn) is a professor in the Department of Information Management and Information Systems in the School of Management at Fudan University, Shanghai, China.

How to run virtual machines together with physical machines, especially when sharing computational resources.

BY NISHANT THORAT, ARVIND RAGHAVENDRAN,
AND NIGEL GROVES

Offline Management in Virtualized Environments

VIRTUALIZATION IS PROLIFIC and heterogeneous, but, despite delivering unprecedented efficiency and dynamism, is also a challenge for traditional IT management tools, techniques, and processes. The problem is multifaceted; we want a common comprehensive management environment that

leverages existing tools, applications, and IT management processes in both physical and virtual environments. However, IT management processes do not always work as expected in virtualized environments due to fundamental differences in the operation of physical and virtual machines.

Though all IT management functions are affected by virtualization, op-

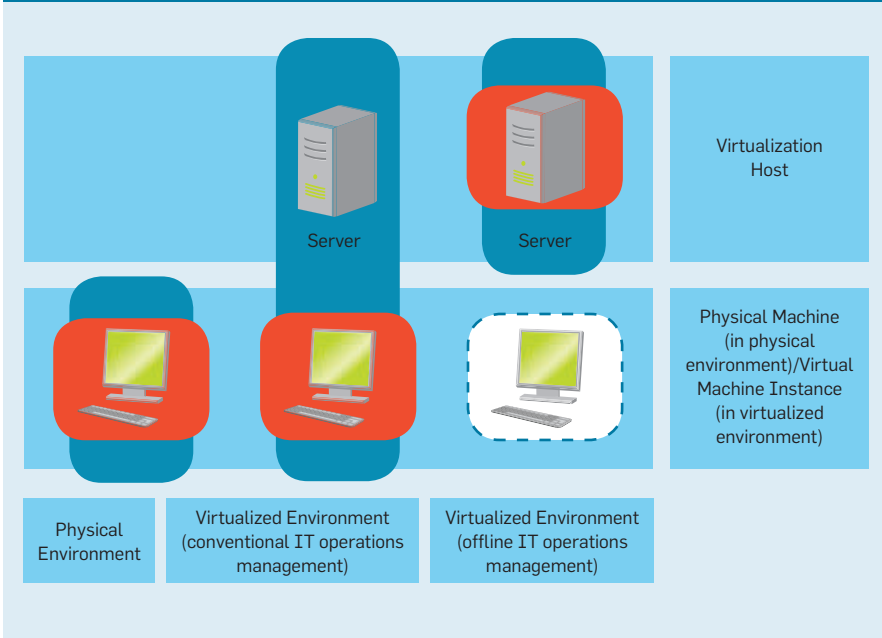
erations management deserves special attention, as it bears most of the load. Operations management must address both physical and virtual environments, and management of the virtualized environment must be as effective as management of the physical environment. IT management needs information, including device discovery, device inventory, software inventory, and operating system inventory. Together, this data defines the information perimeter, within which IT management principles, processes, and tools are applied, forming the IT management perimeter. This article examines these processes and proposes a new complementary IT management model.

Traditional physical IT management tools depend on close alignment

» key insights

- Every offline virtual machine left unpatched is a potential threat when brought online for patching.
- Virtual machines do not have to be online to be managed.
- Using offline management techniques can improve management of virtual environments significantly.

Figure 1. Information perimeter and IT operations management perimeter.



between management perimeter and information perimeter; for example, in the physical world, device inventory information (such as MAC address, RAM, and HDD configuration) resides

in the machine itself, deep in the operating system, and is typically read only through agent-based techniques that understand operating system interfaces. Remote technologies (such as Win-

dows Management Instrumentation, or WMI) still need a service to run on the target. Some technologies (such as Intel Active Management Technology) use a management stack that is part of the hardware platform. The information perimeter and IT management perimeter overlap in physical environments (see Figure 1). This way, the information perimeter acts more like a barrier; needed is an agent on the inside to get management data to the outside and manipulate machine data on the inside. That is, IT management tools must be able to run on physical machines.

Traditional IT management tools follow two implicit assumptions:

Confined to hardware. The information required for IT management is confined to the computer hardware on which it runs; and

Only when running. The information is available only when the computer is running.^a

They force IT management tools to be agent-based; even technologies purporting to be agentless require some kind of service to be running on the managed computer (such as WMI).

In typical environments such systems follow a client-server-based architecture in which a management server does back-end work (such as storing and analyzing inventory information and correlating with the services provided to enterprise IT users). Agents run on managed systems to do the real work of inventory collection, deployment of software and patches, scanning systems for viruses, and backing-up data.

The model is successful for two reasons:

Rarely available. The information collected by the agent is rarely available outside the physical boundaries of machines; and

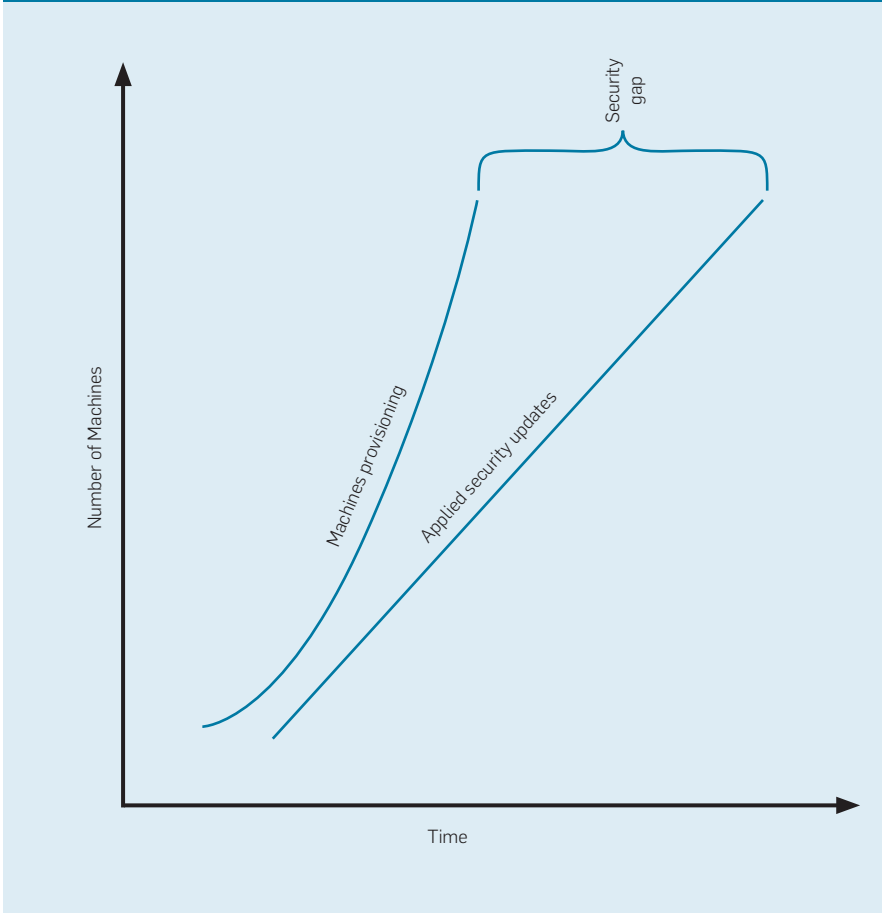
Distributed. Processing is distributed to individual machines, reducing load on servers and promoting scalability.

Applying Traditional Tools

Many IT management vendors take a sensible approach in which physical and virtual computers are managed together in a similar way, extending

^a An exception to this assumption is CA Desktop Migration Manager, providing disk-to-disk migration for data and settings directly to or from an alternate drive when the operating system of the source/target machine is not running.¹

Figure 2. Security vulnerability when applying traditional tools in virtualized environments.



their IT management products to understand and exploit virtualization while allowing their customers to apply the same or similar policies and operations to physical and virtual resources; for example, CA Client Automation extends its physical software deployment capabilities to address the virtual environment (such as managing virtual applications and virtual desktop infrastructures). This means new virtual-delivery options can be adopted while still maintaining the management paradigm already understood in the physical environment. CA Server Automation takes a similar approach regarding machine provisioning, bare-metal provisioning of physical servers, virtualization servers, and provisioning of virtual machines all under one roof.

Some enhancements are based on low-level hypervisor APIs that collect information (such as the virtual machine instance and the host machine that runs it), but for all other operations management activities (such as software asset management, security, and compliance) the traditional physical model is reused within the virtualized environment, usually with little or no modification. The middle component in Figure 1 labeled “Virtualized Environment” shows the information perimeter surrounds the virtualization server/host, though IT management tools are forced to run on the virtual machine instance. The advantage for vendors is they re-

quire no extensive effort to make their tools virtualization ready.

Not having to develop anything further represents a short-term advantage but exposes (in the long term) gaps that could be critical from the perspective of IT management; Table 1 lists aspects of operations management and their complications in virtualized environments.

Traditional patching, antivirus, and vulnerability management tools are agent-based and require managed machines to be running for proper maintenance. This can be a problem in virtualized environments where virtual machines are not expected to be online all the time. Moreover, patching a running virtual machine does little good unless the stored image of the virtual machine is also patched, which is not

always the case with current technologies (such as nonpersistent virtual machines and linked clones).

Traditional patching of offline virtual machines includes virtual machine power on and snapshot commits. The resources needed for this activity can be nontrivial, especially when a large number of virtual machines must be patched, since much of the flexibility of virtualized environments derives from deploying many lightweight virtual machines with specific purposes that run when needed, rather than fewer physical machines that run constantly.⁷

The few commercially available products include Virtual Machine Servicing Tool¹⁰ and VMware vCenter Protect Essentials Plus, formerly Shavlik NetChk Protect,¹² which services offline virtual

Figure 3. Offline management functional model.

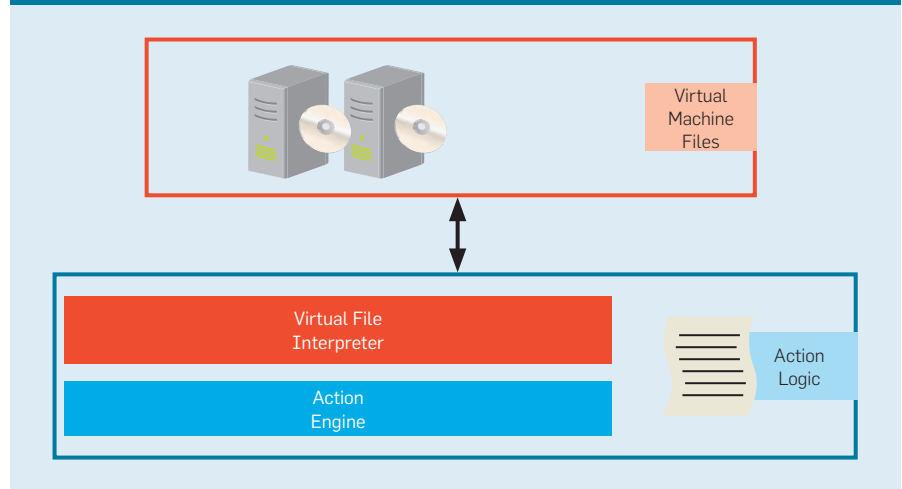


Table 1. Operations management challenges in virtual machines.

Operations Management	Attribute	Challenge
Discovery	Physicality	The assumption is that managed entities exist in physical form and can be discovered through discovery protocols. However, the existence of a virtualized entity depends on the transformation process in the virtualized layer.
Licensing and auditing	Fixed association	Problematic license types and rules in virtualized environments include: ► Device-specific software license models. The license is tied to a particular computer. Dynamic load balancing sometimes used in virtualized environments can cause noncompliance. The license may or may not allow such a move, as when, say, Microsoft imposes 90-day move restrictions in some cases. ► Processor-based software-license models. These licenses require details of the physical processor(s) to determine license compliance, but on a virtual machine the hardware details are often hidden by the hypervisor. ► Virtual machines per-host software license rules. These rules allow the software to run on a certain number of virtual machines for each physical machine with a valid license. Installed instance-based licensing requires separate license fees be paid for each installed instance of an application, regardless of whether the instance is online. Many organizations maintain offline backup copies of VMs, making impractical installed instance-based licensing.
Security	Online	Traditional management processes assume computers can be updated against vulnerabilities in a timely fashion because they run frequently or even constantly; this is less likely for virtual computers.
Change and configuration management	Uniqueness	The assumption is that one entity (such as resource, application, and service) exists only once. Since virtualization has a multiplication effect, entities may exist multiple times, with the same identity.

machine images. However, none of them works in a truly offline fashion; for example, the Microsoft Offline VM Servicing Tool wakes up an offline virtual machine in a guarded environment, applies patches, and services the virtual machine, then returns the updated image to the library. VMware's offering is quasi-offline; patch files are inserted into the offline image, while the actual patching is done when the virtual machine is powered up. In dynamic cloud environments, where virtual machines are provisioned in response to disaster recovery or load balancing, the startup delay is unacceptable.

Virtualization is a major enabler of cloud computing, and administrators must address how to update thousands of virtual machine images.^b

^b Amazon EC2 included more than 25,500 public images (as of February 2012) according to Cloud Market (<http://thecloudmarket.com/stats>). A list compiled by Amazon of 1,098 public Amazon Machine Images (<http://aws.amazon.com/amis>) (as of February 2012) showed approximately 99% of the 1,098 images had not been updated for a month.

Given the advantages and ease of deploying virtual machines, the rate of virtual machine provisioning could outpace the rate of applied security patch updates (see Figure 2). As a result, offline virtual machines may lag security updates, unless updated tools are deployed to specifically address offline virtual machines.

Virtualized Environment Is Different

Virtualized environments add architectural complexity, and virtual systems allocate and use resources differently, more dynamically and less transparently. They have different operational and performance profiles, as well as more fluid configurations. IT management must operate both virtually and physically, and when virtual machines are able to move between physical machines in response to load and resource requirements, IT management based on physical boundaries is awkward, if not impossible.

The fundamental difference between virtual and physical machines

is that while a physical computer and a virtual computer are both digital objects, the data comprising them is far less accessible and far more heterogeneous when working with a physical computer. Physical computer data is stored in volatile memory, as well as in BIOS ROMs, on peripheral cards, and on various vendors' hard drive products. Most physical computer data-configuration information (such as software, accounts, users, groups, patches, services, packages, registry keys, MD5s, and configuration files) are accessible only through code executing within the address space of the machine itself, and almost all physical computer data is available only when the computer is running, since it is buried deep inside platform-specific configuration files and structures. Meanwhile, all virtual computer data is stored in a single file or multiple linked files; in cloud environments these files are typically stored in storage fabric. Nevertheless, virtual machines are not simply data objects

Table 2. IT management use cases for offline virtual machine management.

Offline IT Management Activity	IT Management Use Case
Offline virtual machines and virtual-machine templates are scanned, antivirus/malware signatures updated, and infected files/settings cleaned.	Disaster recovery. Business applications running in virtual environments may be disrupted through hacking, system failure, malicious employee intent, data corruption, and application malfunction. In the event of such disruption, backup (or archived) virtual machines at secondary sites are activated to restore the production environment. Backup VM images must be updated and free of malware. Offline management allows antivirus scanning and signature update and cleaning, so anti-malware security profiles of offline virtual machines are updated automatically without bringing VMs online. Audit and attestation. Vulnerability management can provide evidence to auditors that even offline VMs are compliant.⁴ Security management. Offline VMs must be kept up to date with patches, AV signatures, and firewall rules.⁶ Security policies can be deployed to offline machines to stay in sync with their powered-on counterparts.
Offline virtual machines are scanned for software and associated license information.	License management. Discovered software information can be used for license compliance and to determine if starting offline machines would violate license terms.
Offline virtual machines are scanned for configuration information (such as software, accounts, users, groups, patches, services, packages, registry keys, MD5s, and configuration files).	Configuration management: ▶ The platform-neutral nature of virtual-machine formats may allow access to some information without having to first access platform details; for example, virtual-machine hardware-attribute information required for hardware-asset inventory collection (such as CPU, RAM, chipsets, and network configuration) can be located in virtual-machine files (such as the <code>ovf:VirtualHardwareSection_Type</code> section in the OVF specification). ▶ Before change-management activity is carried out on offline VMs, the discovered configuration information may be used to determine if the proposed change violates any policies. VMs in compliance with a defined set of requirements can be connected to the network.² Change management. Though virtual machines may be designed to be interoperable, the virtual hardware supported by different vendors is different, and the operating system installed on the virtual machine may likewise be different. This offline information can be used for virtual-hardware validation as well; for example, in the case of VMware ESX server, complete virtual-machine installation is not needed to discover it is not compatible, as it uses IDE virtual disks.

to manipulate but actual computers with real workloads and to a large extent the same management requirements as physical systems.

Unlike physical systems, virtualized environments change dynamically and for the most part are server-centric, or composed of digital objects residing and executing on physical servers. The information perimeter may not necessarily be tightly associated with the hardware resource on which it runs. So, for a virtualized environment (unlike a physical environment) IT management does not have to be executed within the running machine.

Offline Operations Management

Virtualization technology is file-based, with files typically available at a central location for ready access by IT management tools. One aspect of file-based technology is that if the file format is known to an IT management tool, the tool should be able to interpret, extract, and update the “data” that is the virtual machine.

Virtualization vendors agree for the most part on interoperable and/or open virtual file formats. The Distributed Management Task Force (DMTF; <http://dmf.org/>) released the Open Virtualization Format, or OVF,³ specification now being adopted by major virtualization vendors (such as Citrix, Microsoft, and VMware) and accepted in August 2010 as an American National Standards Institute (ANSI; <http://ansi.org/>) standard. In addition, other proprietary file formats (such as the Virtual Machine Disk Format, or VMDK,¹¹ and Virtual Hard Disk Image Format, or VHD)⁹ store virtual machines as monolithic files or in multiple layers. These files may be mounted and accessed by external tools and utilities that understand them.^c That is, the information perimeter once viewed as a barrier becomes a standardized enabler.

Virtual machine data does not have to be executing for its state to be managed but can instead be managed and

c VMware provides the DiskMount utility, as well as a programmable interface, to VMDK files through the Virtual Disk Development Kit. Microsoft provides a utility called VHD that mounts a virtual hard disk file (.vhd file) as a virtual disk device on the host operating system.

manipulated offline. This enables a new model for IT operations management in virtualized environments—offline IT management. As explored here, managing a computer in an offline state offers many advantages.

Offline IT Management Model

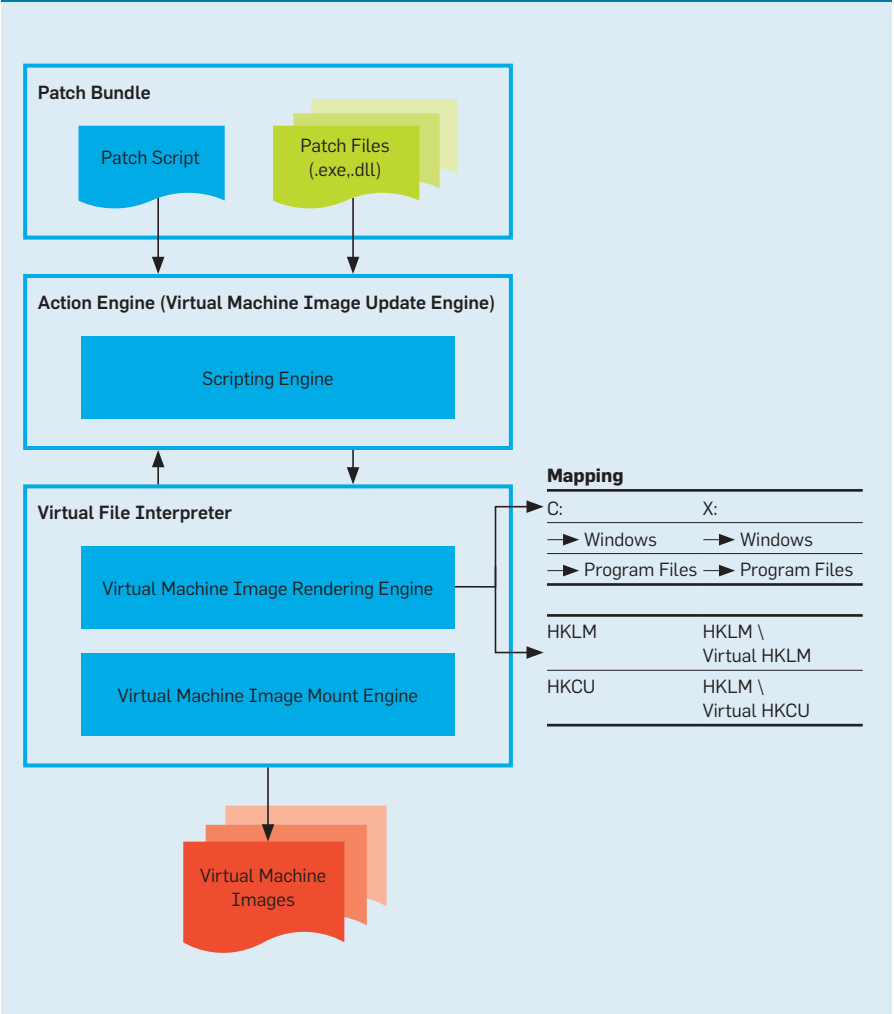
The offline model for IT management of virtualized environments is straightforward, allowing information access or retrieval from virtual machine data, in addition to IT management executed directly on virtual machine data (see Table 2). Information access can be used for IT-asset-management activities (such as hardware asset inventory, software inventory, and software-license auditing and security compliance). IT management can include, but is not limited to, security patches, virus-definition updates, and software upgrades. The following section covers a model that enables offline information access and servicing.

Offline Management Functional Model

The offline management functional model involves a functional system model for offline IT management in virtualized environments (see Figure 3). The virtual file interpreter is the core of the system, responsible for mounting the virtual machine file and providing a common, abstracted, programmatic interface to various kinds of virtual machine data (such as hardware attributes, end-user licensing information, files, and configuration registry). The interoperable file format ensures a virtual machine is accessible irrespective of host and guest platforms. The file format should be well documented and accessible on external media; for example, the OVF, VMDK, VHD, and Kernel-based Virtual Machine^d file

d KVM supports raw images, including in native Quick EMUlator (QEMU) format (qcow2) and VMware format.

Figure 4. Offline virtual machine patching framework.



formats^e can be accessed by mounting externally. Virtual machine files encrypted or password-protected must be decrypted or given proper credentials for the virtual file interpreter to be able to mount and access the virtual machines. The action engine is responsible for carrying out platform-specific actions directed by action scripts (such as remediate vulnerabilities on a Windows system and apply patches to a Debian server).

The model allows for scheduling updates in the background without having to power virtual machines, pos-

sibly resulting in significant resource savings, especially in cloud environments. One obvious overhead that could thus be avoided is generation of action logic. However, in some cases the action engine and action logic may be repurposed traditional agent technology. Techniques like those used for virtualized-application-streaming package creation can also be used to automate action logic creation.⁸

Virtual Machine Patching Framework

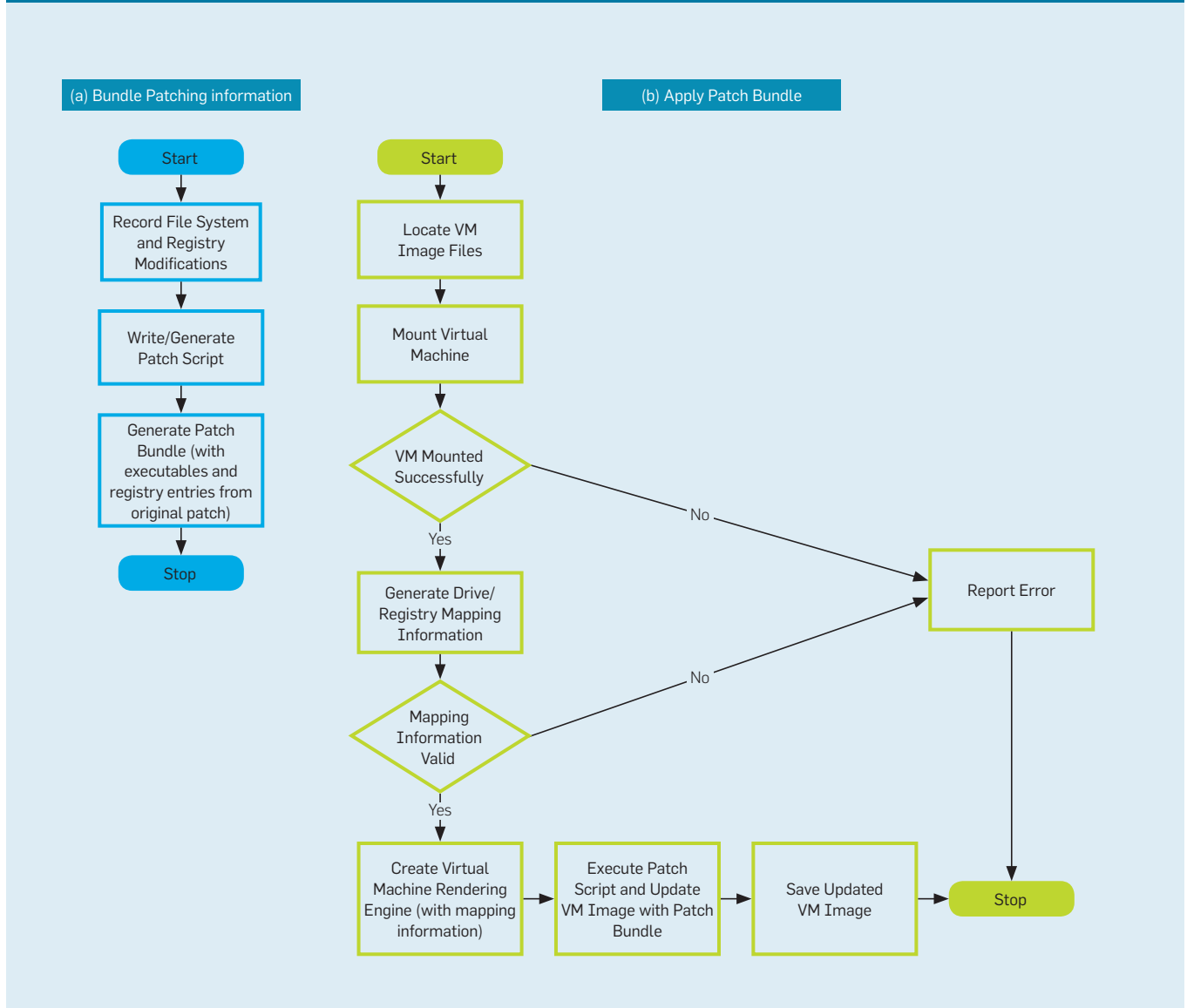
Here, we discuss a novel system for offline virtual machine patching on Windows platforms based on offline management for virtualized environments covered earlier (see Figure 4). Each virtual machine is stored as a file when offline, with its image ren-

dered to file-system data by mounting the virtual machine's virtual-hard-disk drive. The rendering engine uses the registry hive files and virtual hard disks to provide a computing platform that can be used by the offline virtual machine image update engine, in collaboration with the scripting engine. The scripting engine runs the patch script and updates the virtual machine image with patch files; the patch script records the registry and file system location to be updated, along with relevant metadata.

Virtual machine image mount engine. The virtual machine image mount engine is responsible for mounting the virtual machine image as a virtual disk device on the host operating system. It also generates drive-mapping infor-

^e We have not evaluated Solaris zones and mainframe partitioning with LPAR for offline management, though a similar concept can be applied to these technologies, perhaps with enhancements.

Figure 5. Offline patching process.



mation (such as on virtual machine X: was C: provided to the virtual machine image rendering engine).

Virtual machine image rendering engine. This engine uses drive-mapping information and mounted drive to locate registry hives and identify the operating system, system files, and data files. It loads the user hives and the system hives into the host operating system environment and maintains the registry mapping information. It uses drive-mapping information so the patch-script instructions are executed in correct context; for example, the patch script may ask to update C:\Windows\System32\avifile.dll, which, in the context of the host operating system, is X:\Windows\System32\avifile.dll. Likewise, all registry update instructions from the patch script may be redirected to correct locations (such as when the virtual machine's HKLM\Software\CA update instruction is redirected to HKLM\VirtualHKLM\Software\CA).

Offline virtual machine image-update engine. The image-update engine works closely with the scripting engine and the virtual machine image-rendering machine, executing the patch script and updating the files and registry using the patch files; Figure 5 outlines the steps required to apply the patch bundle on an offline virtual machine.

Advantage and Disadvantages of Offline Management

Why manage computers this way? Alternatively, why not manage them this way?

Advantages

- ▶ The computer does not have to be started:
 - » Less demand on resources, including physical computing resources (such as CPU and memory); power; and network bandwidth (if loaded from SAN or network);
 - ▶ More security:
 - » A noncompliant machine does not have to execute before being brought into compliance;
 - » A noncompliant machine may be inhibited from executing;
 - » The runtime external attack surface is reduced since there is less need for an internal agent; and

» A fundamental limitation of traditional host-based anti-malware systems is they run inside the very hosts they protect (called “in the box”), leaving them vulnerable to counter-detection and subversion by malware.⁵

▶ Vendors are regularly developing new virtualization layers (such as a user-persona layer, user-application layer, corporate-security layer, business-application layer, corporate-application layer, and operating-system layer), particularly for virtual desktops; the offline management model allows for interrogating and manipulating data at each one;

▶ Speedier application of changes to computers;

▶ Management code does not have to be compiled for specific platforms; there is no runtime agent so no requirement for different compilation targets and installers;

▶ Potential for being able to repurpose existing agent code (such as file scanning); and

▶ Discovery as file scanning.

Disadvantages

▶ No runtime monitoring; certain management operations need access to the managed computer during execution;

▶ The cost of code for new platforms (such as action logic); development costs are incurred since programmatic interfaces used today by agents are likely different from those available when using a virtualization API; and

▶ No real-time update; virtual machines exist in order to execute, and once they are executing the offline model become less effective; it is not possible to apply changes to a running virtual machine through access to its files, though this may change.

Conclusion

The offline model is a promising operations-management model for virtualized environments. Managing virtualized environments through agent-based tools is convenient and seamless but does not take advantage of the unique characteristics of virtual systems. The offline model augments traditional management models, promising to be more effective and less resource intensive, even though challenges complicate creation of a

seamless management environment in which both physical and virtual systems are managed identically; some aspects of runtime management may require other approaches. The model, which is scalable, is based on constructing a logical view based on interoperable virtual machine file formats and can be applied to a range of operations-management tasks. Keeping in mind that all nodes share resources in virtualized environments, moving to an offline model would help administrators manage these shared resources more efficiently. ■

References

1. CA Technologies Inc. *CA Desktop Migration Manager r12 Product Description*. CA Technologies Inc., 2012; https://supportcontent.ca.com/phpdocs/0/4920/4920_%20r121_GA%20letter32.pdf
2. Ciano, G. and Pichetti, L. *Patent No. U.S. 8,055,737 B2*.
3. *Distributed Management Task Force Open Virtualization Format Specification*, 2011; <http://www.dmtf.org/standards/ovf>
4. Henry, T. *Audit and Attestation in Virtual Environments*. Burton Research, 2009; <http://www.gartner.com/id=1405572>
5. Jiang, X., Wang, X., and Xu, D. Stealthy malware detection and monitoring through VMM-based ‘out-of-the-box’ semantic view reconstruction. *ACM Transactions on Information and System Security* 13, 2 (Feb. 2010).
6. MacDonald, N. *Securing the Next-Generation Virtualized Data Center*. Gartner, Stamford, CT, Mar. 25, 2010; http://www.gartner.com/it/content/1304200/1304218/march_25_securing_virtual_data_center_nmacdonald.pdf
7. Soundararajan, V. and Anderson, J. The impact of management operations on the virtualized data center. In *Proceedings of the 37th Annual International Symposium on Computer Architecture* (June 19–23, 2010).
8. Thorat, N. and Gupta, B. *U.S. Patent No. 20110265076*.
9. *Virtualized Hard Disk Image Format Specification*. Microsoft Inc., Redmond, WA, 2009; <http://technet.microsoft.com/en-us/library/bb676673.aspx>
10. *Virtual Machine Servicing Tool 3.0*. Microsoft Inc., Redmond, WA, Sept. 27, 2010; <http://technet.microsoft.com/en-us/library/cc501231.aspx>
11. *Virtual Machine Disk Format*. VMware Inc.; <http://www.vmware.com/technical-resources/interfaces/vmdk.html>
12. *VMware vCenter Protect Essentials Plus*. VMware Inc.; <http://www.vmware.com/products/datacenter-virtualization/vcenter-protect/overview.html>

Nishant Thorat (Nishant.Thorat@ca.com) is a principal software engineer at CA Technologies, Hyderabad, India.

Arvind Raghavendran (Arvind.Raghavendran@ca.com) is a principal software engineer at CA Technologies, Hyderabad, India.

Nigel Groves (nigel.groves@ca.com) is a senior software architect at CA Technologies, Datchet, England.

DOI:10.1145/2436256.2436274

The main applications and challenges of one of the hottest research areas in computer science.

BY RONEN FELDMAN

Techniques and Applications for Sentiment Analysis

SENTIMENT ANALYSIS (OR OPINION MINING) is defined as the task of finding the opinions of authors about specific entities. The decision-making process of people is affected by the opinions formed by thought leaders and ordinary people. When a person wants to buy a product online he or she will typically start by searching for reviews and opinions written by other people on the various offerings. Sentiment analysis is one of the hottest research areas in computer science. Over 7,000 articles have been written on the topic. Hundreds of startups are developing sentiment analysis solutions and major statistical packages such as SAS and SPSS include dedicated sentiment analysis modules. There is a huge explosion today of ‘sentiments’



available from social media including Twitter, Facebook, message boards, blogs, and user forums. These snippets of text are a gold mine for companies and individuals that want to monitor their reputation and get timely feedback about their products and actions. Sentiment analysis offers these organizations the ability to monitor the different social media sites in real time and act accordingly. Marketing managers, PR firms, campaign managers, politicians, and even equity investors and online shoppers are the direct beneficiaries of sentiment analysis technology.

It is common to classify sentences into two principal classes with regard to subjectivity: objective sentences that

>> key insights

- Sentiment analysis offers organizations the ability to monitor various social media sites in real time and act accordingly.
- Aspect-level sentiment analysis is the most fine-grained analysis of review articles and social media snippets with respect to specific objects and their aspects.
- Utilization of sentiment analysis techniques in stock picking can lead to superior returns.



contain factual information and subjective sentences that contain explicit opinions, beliefs, and views about specific entities. Here, I mostly focus on analyzing subjective sentences. However, I refer to the usage of objective sentences when describing a sentiment application for stock picking.

As an example, here is a review about a hotel in Manhattan.

"The king suite was spacious, clean, and well appointed. The reception staff, bellmen, and housekeeping were very helpful. Requests for extras from the maid were always provided. The heating and air conditioning functioned well; this was good as the weather was variable. The sofa bed was the best I've ever experienced. The king size bed was very comfortable. The building and rooms are very well soundproofed. The neighborhood is the best for shopping, restaurants, and access to subway. Only "complaint" has to do with high-speed Internet access. It's only available on floors 8–12."

Overall the review is very positive about the hotel. It refers to many different aspects of the hotel including: heating, air conditioning, staff courtesy, bed, neighborhood, and Internet

access. Sentiment analysis systems must be able to provide a sentiment score for the whole review as well as analyze the sentiment of each individual aspect of the hotel.

I present the main research problems related to sentiment analysis and some of the techniques used to solve them, then review some of the major application areas where sentiment analysis is being used today. I conclude with some of the open research problems in this field. Due to limited space, I am not able to cover the whole range of problems and techniques; but refer the reader to some of the extensive reviews written on this topic.^{20,21,27}

In this review, I will focus on five specific problems within the field of sentiment analysis:

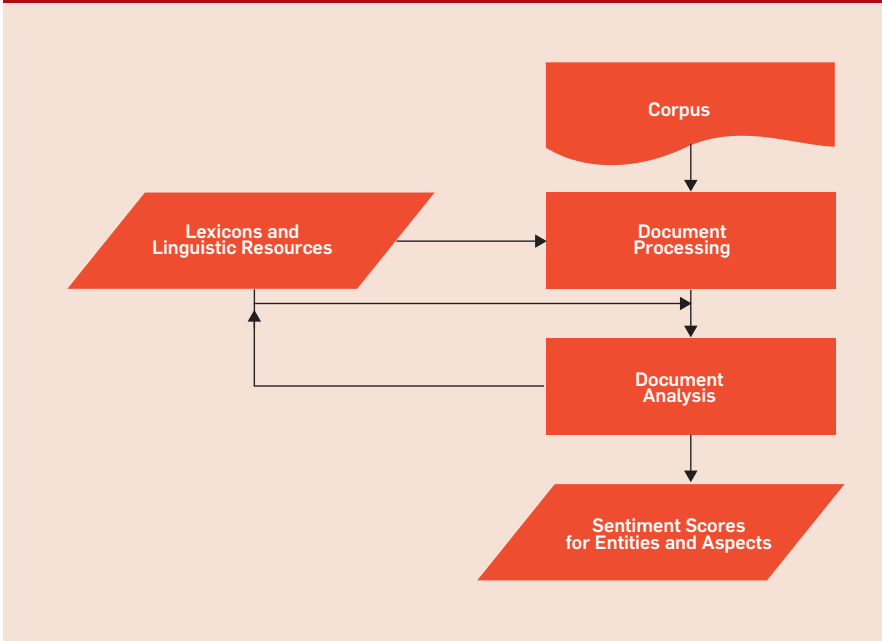
- ▶ Document-level sentiment analysis;
- ▶ Sentence-level sentiment analysis;
- ▶ Aspect-based sentiment analysis;
- ▶ Comparative sentiment analysis; and,
- ▶ Sentiment lexicon acquisition.

Before explaining each of these problems in detail, let's review a general architecture of a generic sentiment analysis system. The architecture is shown in Figure 1.

The input to the system is a corpus of documents in any format (PDF, HTML, XML, Word, among others). The documents in this corpus are converted to text and are pre-processed using a variety of linguistic tools such as stemming, tokenization, part of speech tagging, entity extraction, and relation extraction. The system may also utilize a set of lexicons and linguistic resources. The main component of the system is the document analysis module, which utilizes the linguistic resources to annotate the pre-processed documents with sentiment annotations. The annotations may be attached to whole documents (for document-based sentiment), to individual sentences (for sentence-based sentiment) or to specific aspects of entities (for aspect-based sentiment). These annotations are the output of the system and they may be presented to the user using a variety of visualization tools.

Document-Level Sentiment Analysis

This is the simplest form of sentiment analysis and it is assumed that the document contains an opinion on one

Figure 1. Architecture of a generic sentiment analysis system.

main object expressed by the author of the document. Numerous papers have been written on this topic. There are two main approaches to document-level sentiment analysis: supervised learning and unsupervised learning.

The supervised approach assumes that there is a finite set of classes into which the document should be classified and training data is available for each class. The simplest case is when there are two classes: positive and negative. Simple extensions can also add a neutral class or have some discrete numeric scale into which the document should be placed (like the five-star system used by Amazon). Given the training data, the system learns a classification model by using one of the common classification algorithms such as SVM, Naïve Bayes, Logistic Regression, or KNN. This classification is then used to tag new documents into their various sentiment classes. When a numeric value (in some finite range) is to be assigned to the document then regression can be used to predict the value to be assigned to the document (for example, in the Amazon five-star ranking system). Research²⁸ has shown that good accuracy is achieved even when each document is represented as a simple bag of words. More advanced representations utilize TFIDE, POS (Part of Speech) information, sentiment lexicons, and parse structures.

Unsupervised approaches to document-level sentiment analysis are based on determining the semantic orientation (SO) of specific phrases within the document. If the average SO of these phrases is above some predefined threshold the document is classified as positive and otherwise it is deemed negative. There are two main approaches to the selection of the phrases: a set of predefined POS patterns can be used to select these phrases³⁶ or a lexicon of sentiment words and phrases can be used.³⁴ A classic method to determine the SO of a given word or phrase is to calculate the difference between the PMI (Pointwise Mutual Information) of the phrase with two sentiment words.³⁶ $PMI(P, W)$ measures the statistical dependence between the phrase P and the word W based on their co-occurrence in a given corpus or over the Web (by utilizing Web search queries). The two words used in Turney³⁶ are 'excellent' and 'poor.' The SO measures whether P is closer in meaning to the positive word ('excellent') or the negative word ('poor').

A few researchers^{1,37} have used machine translation to perform document-level sentiment analysis in languages such as Chinese and Spanish that lack the vast linguistic resources available in English. (Their method works by translating the documents to English and then performing senti-

ment analysis on these documents using a sentiment analyzer in English.

Sentence-Level Sentiment Analysis

A single document may contain multiple opinions even about the same entities. When we want to have a more fine-grained view of the different opinions expressed in the document about the entities we must move to the sentence level.

We assume here that we know the identity of the entity discussed in the sentence. We further assume there is a single opinion in each sentence. This assumption can be relaxed by splitting the sentence into phrases where each phrase contains just one opinion. Before analyzing the polarity of the sentences we must determine if the sentences are subjective or objective. Only subjective sentences will then be further analyzed. (Some approaches also analyze objective sentences, which are more difficult.) Most methods use supervised approaches to classify the sentences into the two classes.⁴⁰ A bootstrapping approach was suggested in Hai³² in order to reduce the amount of manual labor needed when preparing a large training corpus. A unique approach based on the minimum cuts was proposed in Pang and Lee.²⁶ The main premise of their approach is that neighboring sentences should have the same subjectivity classification.

After we have zoned in on the subjective sentences we can classify these sentences into positive or negative classes. As mentioned earlier, most approaches to sentence-level sentiment analysis are either based on supervised learning¹⁷ or on unsupervised learning.⁴⁰ The latter approach is similar in nature to that of Turney,³⁶ except that it uses a modified log-likelihood ratio instead of PMI and the number of seed words that are used to find the SO of the words in the sentence is much larger.

Recent research²⁴ has shown that it is advisable to handle different types of sentences by different strategies. Sentences that need unique strategies include conditional sentences, question sentences and sarcastic sentences. Sarcasm is extremely difficult to detect and it exists mainly in political contexts. One solution for identifying sarcastic sentences is described in Tsur et al.³⁵

Aspect-Based Sentiment Analysis

The two previous approaches work well when either the whole document or each individual sentence refers to a single entity. However, in many cases people talk about entities that have many aspects (attributes) and they have a different opinion about each of the aspects. This often happens in reviews about products or in discussion forums dedicated to specific product categories (such as cars, cameras, smartphones, and even pharmaceutical drugs). As an example here is a review of Kindle Fire taken from the Amazon website:

“As a long-time Kindle fan I was eager to get my hands on a Fire. There are some great aspects; the device is quick and for the most part dead-simple to use. The screen is fantastic with good brightness and excellent color, and a very wide viewing angle. But there are some downsides too; the small bezel size makes holding it without inadvertent page-turns difficult, the lack of buttons makes controls harder, the accessible storage memory is limited to just 5GB.”

Classifying this review as either positive or negative toward the Kindle would totally miss the valuable information encapsulated in it. The author provides feedback about many aspects of the Kindle (like speed, ease of use, screen quality, bezel size, buttons, and storage memory size). Some of these aspects are reviewed positively while some of the others get a negative sentiment.

Aspect-based sentiment analysis (also called feature-based sentiment analysis) is the research problem that focuses on the recognition of all sentiment expressions within a given document and the aspects to which they refer.

The classic approach, which is used by many commercial companies, to the identification of all aspects in a corpus of product reviews is to extract all noun phrases (NPs) and then keep just the NPs whose frequency is above some experimentally determined threshold.¹² One approach is to reduce the noise in the found NPs.³⁰ The main idea is to measure for each candidate NP the PMI with phrases that are tightly related to the product category (like phones, printers, or cameras). Only those NPs that have a PMI above a learned thresh-



Aspect-based sentiment analysis is the research problem that focuses on the recognition of all sentiment expressions within a given document and the aspects to which they refer.



old are retained. For instance, for the printer category such phrases, for example, would be “printer comes with” or “printer has.”

Another approach to aspect identification is to use a phrase dependency parser that utilizes known sentiment expressions to find additional aspects (even infrequent ones).³⁹

We can also view the problem of aspect identification as an information extraction problem and then use a tagged corpus to train a sequence classifier such as a Conditional Random Field (CRF)¹⁸ to find the aspects.¹⁴

I have just discussed identification of explicit aspects, that is, aspects that are mentioned explicitly in the sentences. However, there are many aspects that are not mentioned explicitly in the sentences and can be inferred from the sentiment expressions that mention them implicitly. These aspects are called implicit aspects. Examples of such aspects are weight, which can be inferred from the fragment “this phone is too heavy,” or size, which can be inferred from “the camera is quite compact.” One way to extract such implicit aspects is suggested in Liu¹⁰ where a two-phase co-occurrence association rule mining approach is used to match implicit aspects (sentiment expressions) with explicit aspects.

With these two sets we can use a simple algorithm² that determines the polarity of each sentiment expression based on a sentiment lexicon, sentiment shifters (such as negation words), and special handling of adversative conjunctions, such as ‘but.’ The final polarity of each aspect is determined by a weighted average of the polarities of all sentiment expressions inversely weighted by the distance between the aspect and the sentiment expression.

Comparative Sentiment Analysis

In many cases users do not provide a direct opinion about one product but instead provide comparable opinions such as in these sentences taken from the user forums of Edmonds.com: “300 C Touring looks so much better than the Magnum,” “I drove the Honda Civic, it does not handle better than the TSX, not even close.” The goal of the sentiment analysis system in this

case is to identify the sentences that contain comparative opinions, and to extract the preferred entity(-ies) in each opinion.

One of the pioneering papers on comparative sentiment analysis is Jindal and Liu.¹⁵ This paper found that using a relatively small number of words we can cover 98% of all comparative opinions. These words are:

- Comparative adjectives adverbs such as: ‘more,’ ‘less,’ and words ending with –er (for example, ‘lighter’).

- Superlative adjectives and adverbs such as: ‘most,’ ‘least,’ and words ending with –est (for example, ‘finest’).

- Additional phrases such as ‘favor,’ ‘exceed,’ ‘outperform,’ ‘prefer,’ ‘than,’ ‘superior,’ ‘inferior,’ ‘number one,’ ‘up against.’

Since these words lead to a very high recall, but low precision, a naïve Bayes classifier was used to filter out sentences that do not contain comparative opinions. The classifier used sequential patterns as features. The sequential patterns were discovered by the class sequential rule (CSR) mining algorithm. A simple algorithm to identify the preferred entities based on the type of comparative used and the presence of negation is described in Ding et al.³

Sentiment Lexicon Acquisition

As we have seen in the previous discussion, the sentiment lexicon is the most crucial resource for most sentiment analysis algorithms. Here, I briefly mention a few approaches for the acquisition of the lexicon. There are three options for acquiring the sentiment lexicon: manual approaches in which people code the lexicon by hand, dictionary-based approaches in which a set of seed words is expanded by utilizing resources like WordNet,⁸ and corpus-based approaches in which a set of seed words is expanded by using a large corpus of documents from a single domain.

Clearly, the manual approach is in general not feasible as each domain requires its own lexicon and such a laborious effort is prohibitive. I will focus on the other two approaches. The dictionary-based approach starts with a small set of seed sentiment words suitable for the domain at hand. This set of words is then expanded by using

The sentiment lexicon is the most crucial resource for most sentiment analysis algorithms.

Word Net’s synonyms and antonyms. One of the elegant algorithms is proposed in Kamp et al.¹⁶ The method defines distance $d(t1, t2)$ between terms $t1$ and $t2$ as the length of the shortest path between $t1$ and $t2$ in WordNet. The orientation of t is defined as $SO(t) = (d(t, bad) - d(t, good))/d(good, bad)$. $|SO(t)|$ is the strength of the sentiment of t , $SO(t) > 0$ entails t is positive, and t is negative otherwise. The main disadvantage of any dictionary-based algorithm is that the acquired lexicon is domain independent and hence does not capture the specific peculiarities of any specific domain. More advanced dictionary-based approaches are reported in Dragut et al.⁴ and Peng and Park.²⁹

If we want to create a domain-specific sentiment lexicon we have to use one of the many corpus-based algorithms. A classic work¹¹ in this area introduced the concept of sentiment consistency that enables one to identify additional adjectives that have a consistent polarity as a set of seed adjectives. A set of linguistic connectors (AND, OR, NEITHER-NOR, EITHER-OR) was used to find adjectives that are connected to adjectives with known polarity. Consider the sentence “the phone is both powerful and light.” If we know that ‘powerful’ is a positive word, we can assume that by utilizing the connector AND the word ‘light’ is positive as well. In order to eliminate noise the algorithm created a graph of adjectives by using connections induced by the corpus and after a clustering step, positive and negative clusters are formed.

An approach called double propagation for simultaneous acquisition of a domain-specific sentiment lexicon and a set of aspects was introduced in Qiu et al.³¹ This approach used the minipar¹⁹ parser to parse the sentences in the corpus and find associated aspects and sentiment expressions. The algorithm starts with a seed set of sentiment expressions and uses a set of predefined dependency rules and the minipar parser to find aspects that are connected to the sentiment expressions. It then uses the found aspects to find more sentiment expressions that in turn find more aspects. This mutual bootstrapping process stops when no more as-

pects or sentiment expressions can be added. For example, in “Kindle Fire has an amazing display,” the adjective ‘amazing’ modifies the noun ‘display,’ so given that ‘amazing’ is a sentiment expression and we have the rule “a noun which is modified by a sentiment expression is an aspect,” we can extract ‘display’ as an aspect. Conversely, if we know ‘display’ is an aspect, then using a similar rule we can infer that ‘amazing’ is a sentiment expression. The algorithm uses several additional constraints to reduce the effect of noise.

Migrating a sentiment lexicon from one domain to another domain was studied in Du et al.⁵ An algorithm for acquiring a slightly different type of lexicon called a connotation lexicon is reported in Feng et al.⁹ A connotation lexicon contains words that express sentiment either explicitly or implicitly. For instance, award and promotion have positive connotations and cancer and war have negative connotations.

Applications

The most common application of sentiment analysis is in the area of reviews of consumer products and services. There are many websites that provide automated summaries of reviews about products and about their specific aspects. A notable example of that is “Google Product Search.”

Twitter and Facebook are a focal point of many sentiment analysis applications. The most common application is monitoring the reputation of a specific brand on Twitter and/or Facebook. One application that performs real-time analysis of tweets that contain a given term is tweetfeel (<http://www.tweetfeel.com>).

Sentiment analysis can provide substantial value to candidates running for various positions. It enables campaign managers to track how voters feel about different issues and how they relate to the speeches and actions of the candidates. An analysis of tweets related to the 2010 campaign can be found at <http://www.nytimes.com/interactive/us/politics/2010-twitter-candidates.html>.

Another important domain for sentiment analysis is the financial markets. There are numerous news items, articles, blogs, and tweets about each public company. A sentiment analysis system can use these various sources to find articles that discuss the companies and aggregate the sentiment about them as a single score that can be used by an automated trading system. One such system is The Stock Sonar (<http://www.thestocksonar.com>).⁷ This system (developed by Digital Trowel) shows graphically the daily positive and negative sentiment about each stock alongside the graph of the price of the stock. An example

of such a graph is shown in Figure 2. The sentiment for CHK is extremely negative and indeed the stock went down considerably between April 21, 2012 and May 22, 2012. The graph is interactive, so a click on any point will reveal the events and sentiment expressions behind the various increases in positive or negative sentiment, as shown in Figure 3.

StockTwits (<http://www.stocktwits.com>) is a site that shows all tweets that contain at least one stock ticker in them (A ‘\$’ sign must be before the ticker of the stock to signal it is a ticker). The following are three tweets about Google (Ticker: GOOG) from Sunday, July 29, 2012.

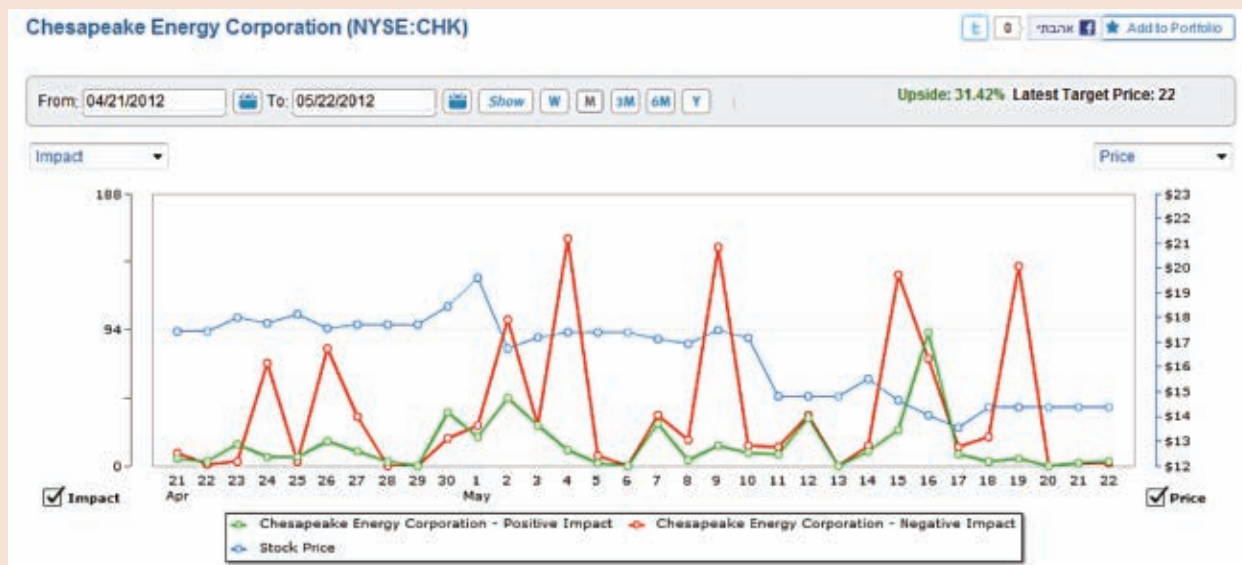
1. IMO, if market up Monday, \$PCLN \$AAPL look much better for Call options plays than \$GOOG. \$GOOG needs a little rest

2. \$GOOG Monday will probably prove to be a nice shorting opportunity. I’m guessing it will close at or at least trade to 625.

3. Slag \$MSFT all u want, but it gets how TV is evolving. The next gen Kinect is something I want to buy, NOT \$GOOG TV v.39.3 beta

Detecting sentiment on the first tweet will be done by utilizing comparative sentiment analysis techniques. We will conclude that the writer is positive on PriceLine (PCLN) and Apple (AAPL) and negative on

Figure 2. Sentiment graph of Chesapeake Energy (<http://www.thestocksonar.com>).



Google. Analyzing the second tweet will reveal a negative sentiment on Google (shorting opportunity). Since Google closed on Friday, July 27,

2012 at \$634.96, the author predicts a down movement of 1.57% to \$525. Clearly, we need to be able to get historical prices of stocks to do proper

analysis of the tweets. The third and last tweet is the most difficult to analyze since it requires background knowledge not available inside the tweet. We need to know that Kinect is a product of Microsoft (MSFT) and hence the author has a positive opinion on MSFT and a negative opinion on Google (by utilizing the sentiment shifter “NOT”). These examples show some of the challenges facing sentiment analysis systems when trying to analyze short messages that include reference to additional objects (products and stock prices in this case). The systems must utilize background knowledge in order to determine the relationship between the sentiment targets and the other objects.

An application that utilizes comparative sentiment analysis to assess the market structure of sedan cars and drugs for diabetes is described in Netzer et al.²⁵ In Figure 4 we can see a visual map that shows the various connections between drugs and symptoms. Two types of connections are extracted by the sentiment analysis system: Drug Causes Symptom (negative, shown in red) and Drug Remedies Symptom (positive, shown in blue).

Research Issues

There are many open research issues in sentiment analysis, including:

1. There is a need for better modeling of compositional sentiment. At the sentence level, this means more accurate calculation of the overall sentence sentiment of the sentiment-bearing words, the sentiment shifters, and the sentence structure.

2. Each product has many names that refer to it even within the same document and clearly across documents. This issue of automatic entity resolution is not yet solved. Another related major hurdle is handling of anaphora resolution in an accurate way. This is a problem for aspect extraction too, that is, how to group aspects, for example, “battery life” and “power usage” refer to the same aspect of a phone.

3. When a document discusses several entities, it is crucial to identify the text relevant to each entity. Current accuracy in identifying the relevant text is far from satisfactory.

Figure 3. The negative events for CHK on May 9th (<http://www.thestocksonar.com>).

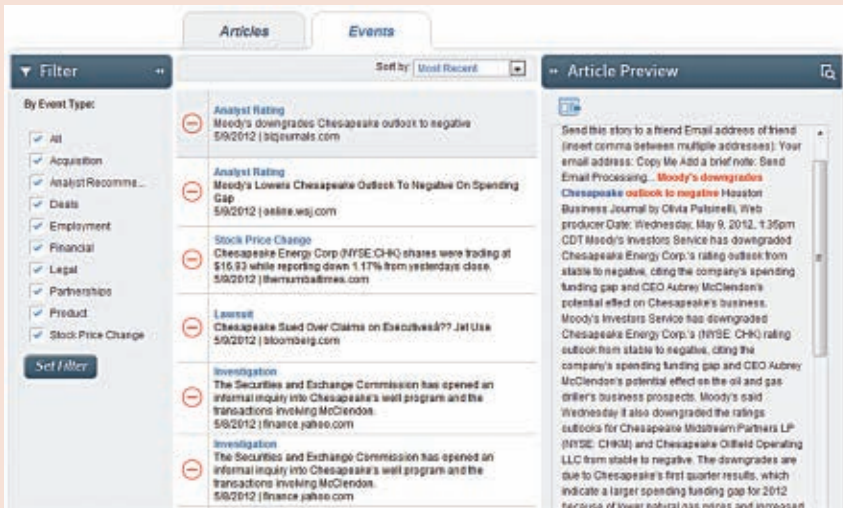
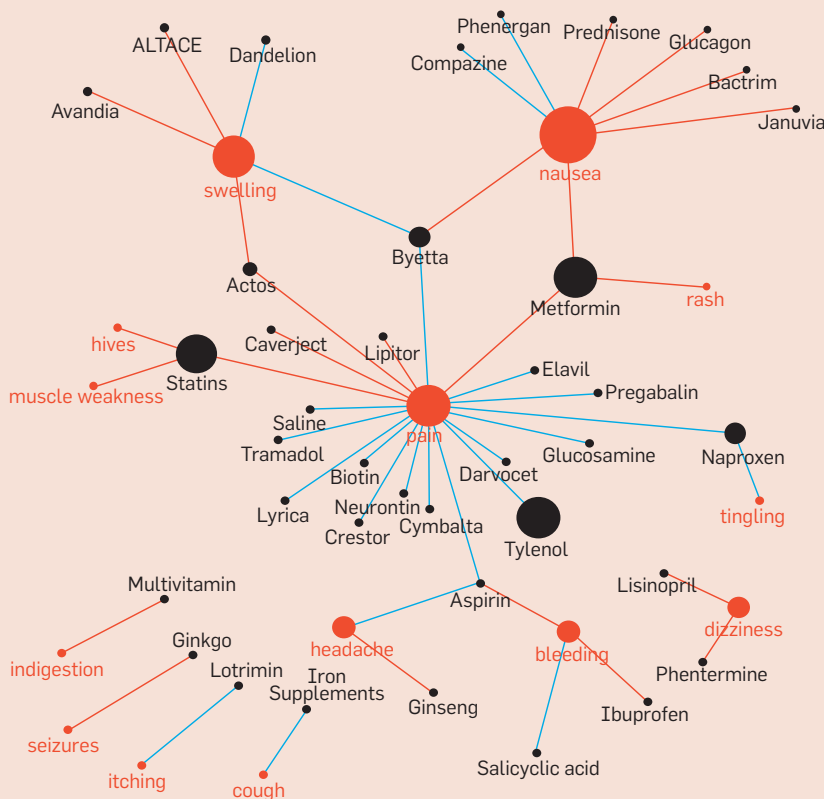


Figure 4. Drugs and symptoms (diabetes forums) based on extractions done by Visual Care (<http://www.digitltrowel.com>).



4. Although there are some approaches that use classification methods to identify sarcasm, they are not yet integrated within autonomous sentiment analysis systems.

5. Noisy texts (those with spelling/grammatical mistakes, missing/problematic punctuation and slang) are still a big challenge to most sentiment analysis systems.

6. Many of the statements about entities are factual in nature and yet they still carry sentiment. Current sentiment analysis approaches determine the sentiment of subjective statements and overlook such objective statements. There is a need for algorithms that use context to attach sentiment scores to objective (factual) statements. Such statements occur frequently in news articles.

Resources

The following resources contain sentiment lexicons that can be used within sentiment analysis systems:

- ▶ General Inquirer lexicon;³³ http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm.
- ▶ Sentiment lexicon;¹³ <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>.
- ▶ MPQA subjectivity lexicon;³⁸ http://www.cs.pitt.edu/mpqa/subj_lexicon.html.
- ▶ SentiWordNet;⁶ <http://sentiwordnet.isti.cnr.it/>.
- ▶ Emotion lexicon;²³ <http://www.purl.org/net/emolex>.
- ▶ Financial Sentiment Lexicons (suited for the determination of the sentiment of financial documents);²² http://nd.edu/~mcdonald/Word_Lists.html.

Conclusion

This article reviewed some of the main research problems within the field of sentiment analysis and discussed several algorithms that aim to solve each of these problems. I have also described some of the major applications of sentiment analysis and provided a few major open challenges. Many of the commercial sentiment analysis systems still use simplistic techniques in order to avoid these open challenges and hence their performance leaves a lot to be desired. Providing satisfactory solutions to these challenges will make the area of sentiment analysis far more widespread.

Acknowledgments

I thank Lyle Ungar, Bing Liu, Benjamin Rosenfeld, and Roy Bar-Haim for helpful comments on drafts of this article. ☐

References

1. Brooke, J., Tofiloski, M. and Taboada, M. Cross-linguistic sentiment analysis: From English to Spanish. In *Proceedings of RANLP* (2009).
2. Ding, X., Liu, B. and Yu, P.S. A holistic lexicon-based approach to opinion mining. In *Proceedings of the Conference on Web Search and Web Data Mining* (2008).
3. Ding, X., Liu, B. and Zhang, L. Entity discovery and assignment for opinion mining applications. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2009).
4. Dragut, E.C., Yu, C., Sistla, P. and Meng, W. Construction of a sentimental word dictionary. In *Proceedings of ACM International Conference on Information and Knowledge Management* (2010).
5. Du, W., Tan, S., Cheng, X. and Yun, X. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *Proceedings of ACM International Conference on Web Search and Data Mining* (2010).
6. Esuli, A. and Sebastiani, F. Determining term subjectivity and term orientation for opinion mining. In *Proceedings of Conf. of the European Chapter of the Association for Computational Linguistics* (2006).
7. Feldman, R., Rosenfeld, B., Bar-Haim, R. and Fresko, M. The Stock Sonar—Sentiment Analysis of Stocks Based on a Hybrid Approach. *IAAI-12* (2011), 1642–1647.
8. Fellbaum, C.D. *Wordnet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.
9. Feng, S., Bose, R. and Choi, Y. Learning general connotation of words using graph-based algorithms. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics* (Edinburgh, Scotland, UK, 2011), 1092–1103.
10. Hai, Z., Chang, K. and Kim, J.-j. Implicit feature identification via co-occurrence association rule mining. *Computational Linguistics and Intelligent Text Processing* (2011), 393–404.
11. Hatzivassiloglou, V. and K. McKeown, Predicting the semantic orientation of adjectives. In *Proceedings of the Joint ACL/EACL Conference* (1997), 174–181.
12. Hu, M. and Liu, B. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2004), 168–177.
13. Hu, M. and Liu, B. Mining opinion features in customer reviews. In *Proceedings of AAAI* (2004), 755–760.
14. Jakob, N. and Gurevych, I. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (2010).
15. Jindal, N. and Liu, B. Identifying comparative sentences in text documents. In *Proceedings of ACM SIGIR Conf. on Research and Development in Information Retrieval* (2006).
16. Kamps, J., Marx, M., Mollen, R.J. and de Rijke, M. Using WordNet to measure semantic orientation of adjectives. *LREC*, 2004.
17. Kim, S.-M. and Hovy, E. Crystal: Analyzing predictive opinions on the Web. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2007).
18. Lafferty, J., McCallum, A. and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA, 2001, 282–289.
19. Lin, D. Minipar; <http://webdocs.cs.ualberta.ca/~lindek/minipar.htm>. 2007.
20. Liu, B. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*. N. Indurkha and F.J. Damerau, eds. 2010.
21. Liu, B. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers, 2012.
22. Loughran, T. and McDonald, B. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance* 66, 1 (2011), 35–65.
23. Mohammad, S.M. and Turney, P.D. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text* (2010).
24. Narayanan, R., Liu, B. and Choudhary, A. Sentiment analysis of conditional sentences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (Singapore, 2009). Association for Computational Linguistics, 180–189.
25. Netzer, O., Feldman, R., Fresko, M. and Goldenberg, Y. Mine your own business: Market structure surveillance through text mining. *Marketing Science*, 2012.
26. Pang, B. and Lee, L. A Sentimental Education: Sentiment Analysis using Subjectivity Summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics* (2004), 271–278.
27. Pang, B. and Lee, L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2, 1-2 (2008), 1–135.
28. Pang, B., Lee, L. and Vaithyanathan, S. Thumbs up? Sentiment Classification using machine learning techniques. In *Proceedings of EMNLP-02, 7th Conference on Empirical Methods in Natural Language Processing* (Philadelphia, PA, 2002). Association for Computational Linguistics, Morristown, NJ, 79–86.
29. Peng, W. and Park, D.H. Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media* (2011).
30. Popescu, A.-M. and Etzioni, O. Extracting product features and opinions from reviews. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (2005).
31. Qiu, G., Liu, B., Bu, J. and Chen, C. Opinion word expansion and target extraction through double propagation. *Computational Linguistics* 37, 1 (2011), 9–27.
32. Riloff, E. and Wiebe, J. Learning extraction patterns for subjective eExpressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2003).
33. Stone, P. The general inquirer: A computer approach to content analysis. *Journal of Regional Science* 8, 1 (1968).
34. Taboada, M., J. Brooke, J., Tofiloski, M., Voll, K. and Stede, M. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37, 2 (2011), 267–307.
35. Tsur, O., Davidov, D. and Rappoport, A. A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Fourth International AAAI Conference on Weblogs and Social Media* (2010).
36. Turney, P. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association for Computational Linguistics* (2002), 417–424.
37. Wan, X. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (Honolulu, Hawaii, 2008). Association for Computational Linguistics, 553–561.
38. Wilson, T., Wiebe, J. and Hoffmann, P. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing* (2005), 347–354.
39. Wu, Y., Zhang, Q., Huang, X. and Wu, L. Phrase dependency parsing for opinion mining. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (2009).
40. Yu, H. and Hatzivassiloglou, V. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2003).

Ronen Feldman (Ronen.Feldman@huji.ac.il) is a professor of information systems in the School of Business Administration at The Hebrew University of Jerusalem, Israel.

research highlights

P. 91

Technical Perspective Understanding Pictures of Rooms

By David Forsyth

P. 92

Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding

By Huayan Wang, Stephen Gould, and Daphne Koller

Technical Perspective

Understanding Pictures of Rooms

By David Forsyth

THE RICH WORLD is getting older, so we will see many efforts to build robots that can provide some in-home care for frail people. These robots will need computer programs that can see and understand rooms because rooms are where people live much of their lives. These robots could use, for example, motion or range data to measure the shape of the room. This does not always apply; motion data might not be available. For example, when a robot first opens a door, it should likely determine whether it might fall before it moves. In some applications, range data is not available either.

But how can we understand a room from a single picture? A picture shows us a two-dimensional projection of a three-dimensional world. This seldom confuses people. We can usually report where objects are in a scene without difficulty, and we can usually get the answer about right, too. We can reason about the empty space we see in pictures, and answer questions like: Would a bed fit here? Is the table far away from the chair? And our answers are usable, if not perfect.

There are now very accurate computer vision methods for reconstructing geometry from videos or from collections of pictures. These methods operate at prodigious scales (substantial fractions of cities have been reconstructed) and with high accuracy. Recovering geometry from a single image still presents important puzzles. A precise representation of all geometric information is likely too much to ask for except in very special cases.

Even rough estimates of geometric information are surprisingly useful. Some years ago, Derek Hoiem and colleagues showed that estimating the horizon in an outdoor scene could improve the performance of a pedestrian detector. This works because real pedestrians have feet below the horizon (they can't levitate). Perspective effects mean the closer a pedestrian's feet are

to the horizon, the smaller the image region the pedestrian should occupy. Detector outputs that do not follow this rule are likely wrong, and discarding them significantly improves overall performance.

What is not yet known is (a) what is useful, (b) what is available, and (c) how to balance errors in the representation recovered. The primary sources of error are bias—where the method cannot represent the right answer, and so must give an answer that is wrong—and variance—where the method could represent the right answer, but becomes overwhelmed by the need to estimate too much information. A representation that tries to recover the depth and the surface normal at every image pixel will likely get most pixels wrong, and so have variance problems, because the image is savagely ambiguous. Representing a room as a box incurs bias; the representation is usually wrong because rooms very often are not exactly boxes, and because there is usually other geometry (beds, chairs, tables) lying around.

For many applications, it is enough to find a box that is a good approxima-

tion to the room. Varsha Hedau and colleagues have demonstrated that knowing a fair box estimate makes it easier to detect beds and large items of furniture. Kevin Karsch and colleagues show how to use an approximate box to infer the lighting in a room, and so insert correctly shaded virtual objects into real pictures. Getting a good approximate box is difficult, because the edges and corners of the room are typically hidden by furniture, which is unceremoniously called “clutter” in the literature. So we must fit a box to the reliable features in the image, and discount the furniture when we do so.

Wang, Gould and Koller's work, detailed in the following paper, is the best current method to do this. Their method rests on two important points. First, they show how to learn a function to score room hypotheses. This function is trained such that the best scoring room will tend to be close to the right answer. Therefore, to fit a room to a new picture, one searches for the best scoring room. Second, clutter tends to be consistent in appearance and to be in consistent places. For example, beds tend to be on the floor next to a wall. So one can tell which parts of the image to discount when computing the scoring function: it looks like clutter, and it is where clutter tends to be.

Their method scores better than any other on the current standard test of accuracy for estimating rooms. Moreover, as figures 3 and 4 in the paper illustrate, there is more to come. Knowing what parts of the image are clutter gives us very strong cues to where the furniture is. There will soon be methods that can produce very detailed maps of room interiors, fit for robots to use. 

How can we understand a room from a single picture? A picture shows us a two-dimensional projection of a three-dimensional world.

David Forsyth (daf@illinois.edu) is a professor in the Thomas M. Siebel Center for Computer Science at the University of Illinois, Urbana, IL.

© 2013 ACM 0001-0782/13/04

Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding

By Huayan Wang, Stephen Gould, and Daphne Koller

Abstract

We address the problem of understanding an indoor scene from a single image in terms of recovering the room geometry (floor, ceiling, and walls) and furniture layout. A major challenge of this task arises from the fact that most indoor scenes are cluttered by furniture and decorations, whose appearances vary drastically across scenes, thus can hardly be modeled (or even hand-labeled) consistently. In this paper we tackle this problem by introducing latent variables to account for clutter, so that the observed image is jointly explained by the room and clutter layout. Model parameters are learned from a training set of images that are only labeled with the layout of the room geometry. Our approach enables taking into account and inferring indoor clutter *without* hand-labeling of the clutter in the training set, which is often inaccurate. Yet it outperforms the state-of-the-art method of Hedau et al.⁷ that requires clutter labels. As a latent variable based method, our approach has an interesting feature that latent variables are used in direct correspondence with a concrete visual concept (clutter in the room) and thus interpretable.

1. INTRODUCTION

We address holistic understanding of indoor scenes from a single image. Our model takes an image of an indoor scene as input, and produces boundaries between the floor, the walls, and the ceiling (we call them the *box layout*), as well as segmentation of the clutter such as furniture and decorations (Figure 1). Learning the model parameters requires training images with hand-labeled box layout but not the clutter significantly reducing labeling effort.

Holistic scene understanding has attracted much attention in computer vision.^{5,9,10,11} One of the goals is to make use of *image context* to help improve traditional vision tasks, such as object detection. The image context can be represented by many aspects. For example, one could use a category label (e.g., street and kitchen), as it imposes a strong prior on the likely and unlikely objects to be detected (e.g., cars on the street). In this paper we focus on indoor scenes, for which we represent the image context by a box layout of the room and clutter. On one hand, such knowledge could be useful as a geometric constraint in a variety of traditional computer vision tasks such as object detection¹⁴ and motion

* This work was done when S. Gould was a Ph.D. candidate at Stanford University.

Figure 1. Output of our method. First row: the inferred box layout illustrated by red lines representing face boundaries. Second row: the inferred clutter layout.



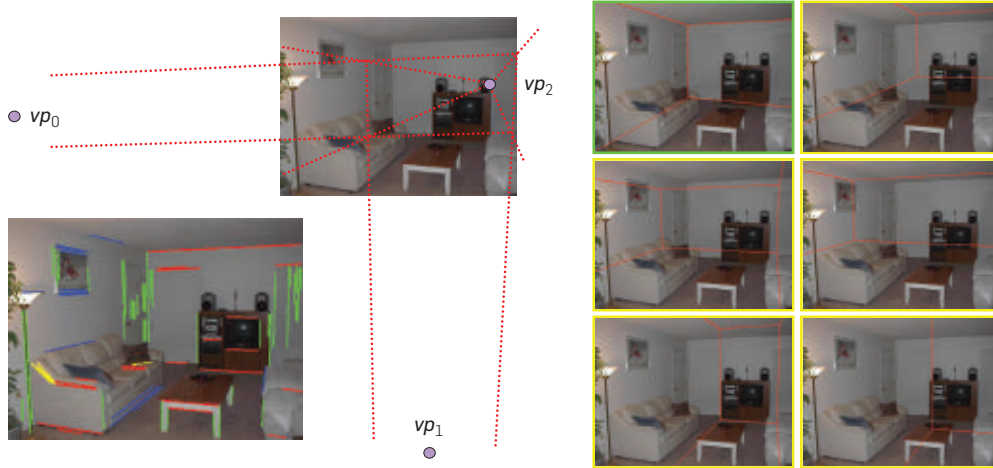
planning.⁶ On the other hand it could also be used as input to infer more fine-grained contextual information such as the 3D layout.

Given an image of an indoor scene, we perform two preprocessing steps. First, we detect the *vanishing points*, which play an essential role in characterizing the structure of an indoor scene. Due to the effect of perspective projection, a set of parallel lines in 3D are either parallel or intersect at a common point in the 2D image. The points where sets of 3D parallel lines meet in 2D are called vanishing points. Hedau et al.⁷ observed that most indoor scenes are characterized by three vanishing points. Given these points, we can generate a family of box layouts. Specifically, as shown in Figure 2, we detect long lines in the image, and cluster them into three dominant groups corresponding to three vanishing points vp_0 , vp_1 , and vp_2 . Candidate box layouts can be generated by extending two rays from vp_0 , two rays from vp_1 , and connecting the four intersections with vp_2 . Infinitely many candidate box layouts can be generated in this manner, and the task becomes to identify the one that best fits the observed image.

Our method also infers clutter within the scene as shown in the second row of Figure 1. It is computationally expensive to perform reasoning on each pixel as to whether or not it belongs to clutter. We therefore perform an over-segmentation of the image using the mean-shift algorithm² and reason

The original version of this paper was published in *the 11th European Conference on Computer Vision, Part II, LNCS 6312 Proceedings*, Springer.

Figure 2. Lower-left: Three groups of lines (shown in R, G, B) corresponding to the three vanishing points. There are also “outlier” lines (shown in yellow). Upper-left: A candidate box layout is generated. Right: Different candidate box layouts (in yellow frames) are generated in the same way, and the hand-labeled true box layout (in green frame).



about the so-called superpixels, that is, we reason over small contiguous segments that have been predetermined according to their spatial and appearance properties. We typically have less than a hundred segments (superpixels) for each image. In reasoning about the clutter we treat each segment as a single entity.

Building on these preliminaries, we introduce our model, inference and learning approach in the next section. Experimental results are discussed in Section 3.

2. APPROACH

2.1. Model

We use $\mathbf{x}$ to denote the input, that is, the image together with its over-segmentation and vanishing points. To capture the appearance of pixels, we use a 21 dimensional vector to represent each pixel, including various color and texture cues. They are also included in $\mathbf{x}$. The output variable $\mathbf{y}$ defines the box layout, and the latent variable $\mathbf{h}$ specifies the clutter as discussed below.

As we have mentioned, the box layout is determined by four parameters (two rays sent from $\mathbf{vp}_0$ and two from $\mathbf{vp}_1$). We therefore parameterize $\mathbf{y}$ as $\mathbf{y} = \{y_i\}_{i=1}^4$ that specify where these four rays intersect with the image central line. We use vertical and horizontal central line for $\mathbf{vp}_0$ and $\mathbf{vp}_1$, respectively. The resulting box layout divides the image into at most five faces: *ceiling*, *left-wall*, *right-wall*, *front-wall*, *floor* as shown in Figure 2. Note that some of the five faces could be absent if one or multiple rays do not intersect within the extent of the image. Moreover, there exist ambiguous interpretations (*left/front* vs. *front/right*) when only two walls are visible. We also define a base distribution $p_0(\mathbf{y})$ (used in inference) over the four dimensional space of $\mathbf{y}$. This is estimated by fitting a multivariate Gaussian with diagonal covariance to the training data.

The latent variable $\mathbf{h}$ is a binary vector with one entry for each segment in the over-segmented image. The entries indicate whether or not each corresponding segment belongs to the clutter. The variables are called “latent” because they are

never observed, that is, we do not require the clutter layout to be labeled in the training images. In fact, drawing the boundaries of the furniture and decorations by hand is not only time-consuming but also ambiguous in many cases. For example, should windows and floor rugs be labeled as clutter? In spite of these difficulties, accounting for clutter appropriately is essential to the success of modeling the scene geometry. This is because the clutter often obscures the geometric structure and occludes boundaries between faces. Moreover, appearance and layout of clutter can vary drastically across different scenes, so it is extremely difficult (if not impossible) to model it consistently. Our latent variable approach effectively addresses this issue.

Our energy-based model measures the consistency among $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{h}$, that is, scores how well the observed image agrees with a given box and clutter layout.

$$E_w(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) \rangle + E^0(\mathbf{x}, \mathbf{y}, \mathbf{h}). \quad (1)$$

The joint feature mapping $\Psi \in \mathbb{R}^n$ is a vector that contains many features that take into account image cues from various aspects including color, texture, perspective consistency, and overall layout. The parameter vector $\mathbf{w}$ specifies the weights of these features and is learned from the labeled training set. E^0 is an energy term that captures our prior knowledge on the role of the latent variables. Given an energy function (with Ψ , $\mathbf{w}$, and E^0 specified), the problem of inferring the box and clutter layout can be formulated as:

$$(\bar{\mathbf{y}}, \bar{\mathbf{h}}) = \underset{(\mathbf{y}, \mathbf{h})}{\operatorname{argmin}} E_w(\mathbf{x}, \mathbf{y}, \mathbf{h}). \quad (2)$$

The prior energy term E^0 consists of two parts,

$$E^0(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \alpha^a E^a(\mathbf{x}, \mathbf{y}, \mathbf{h}) + \alpha^c E^c(\mathbf{y}, \mathbf{h}). \quad (3)$$

Here E^a summarizes the appearance variance of each major face excluding all clutter segments. This encodes the prior belief that the major faces should have a

relatively consistent appearance after the clutter is taken out. Specifically,

$$\mathbf{E}^a = \sum_{f \in \mathbb{F}} \sum_{a \in \mathbb{A}} \frac{\sum_{p \in f} \mathbf{1}(p_h = 0) \cdot (p_a - p_a^f)^2}{\sum_{p \in f} \mathbf{1}(p_h = 0)} \quad (4)$$

where $\mathbb{F}$ is the set of five major faces: *floor, ceiling, left wall, front wall, right wall*, $\mathbb{A}$ is the set of 21 appearance features, and $\mathbf{1}(\cdot)$ is the indicator function returning one if its argument is true and zero otherwise. Here p is a pixel; p_a is its appearance feature value indexed by a ; p_a^f is the average of that appearance feature value within face f ; and p_h is the value of the latent variable at that pixel. We define $p_h = 0$ to mean that pixel p is *not* a clutter. Note, however, that this term could be minimized by assigning almost everything as clutter and leaving only a tiny uniform piece with very consistent appearance. To avoid such degenerate solutions we introduce the second term:

$$\mathbf{E}^c = \sum_{f \in \mathbb{F}} \left[|f| \cdot \exp \left(\beta \cdot \frac{\sum_{p \in f} \mathbf{1}(p_h = 1)}{|f|} \right) \right] \quad (5)$$

which penalizes “clutteriness” of each face. We adopt the exponential form because it exhibits superlinear penalty as the percentage of clutter increases. We will address how to determine the parameters β , α^a , α^c , as well as $\mathbf{w}$ in Section 2.3.

The features in Ψ are defined in a similar spirit: capturing the synergy of $\mathbf{y}$ and $\mathbf{h}$ in explaining the observation. For brevity we only give a high level description.

We have features to account for face boundaries. Ideally the boundaries between the five major faces should either be explained by a long line or occluded by some furniture. Therefore we introduce two features for each boundary: the percentage of its length in clutter regions (i.e., occlusion), and the percentage of its length that approximately overlaps with a detected line segment.

We also have features for perspective consistency, which we adopt from Hedau et al.⁷ Note that the lines in the image fall into three groups corresponding to the three vanishing points (as in Figure 2). For each face, we are more likely to observe lines from two of the three groups. For example, on the front wall we are more likely to observe lines belonging to $\mathbf{vp}_0$ and $\mathbf{vp}_1$, but not $\mathbf{vp}_2$. We capture this with features that quantify the total length of line segments from each of the three groups in each of the five faces, and have a separate feature value for clutter and non-clutter regions.

In addition we have features that measure cross-face differences. For the 21 appearance values, we compute the difference between each pair of adjacent faces excluding clutter. Finally, we have features for some global properties of the box layout. For each of the five major faces, we use a binary feature indicating whether or not it is absent, and its percentage area in the image. For each of the four parameters $\{y_i\}_{i=1}^4$, we compute their likelihood under $p_0(\mathbf{y})$ as bias features.

2.2. Approximate inference

For now we assume that all model parameters are fixed and we want to solve the minimization problem (2). Because the joint feature mapping Ψ and prior energy $\mathbf{E}^0$ are defined in a rather complex way, it cannot be solved analytically. Our inference procedure is based on ICM (Iterated Conditional Modes¹ as shown in Algorithm 1. The basic idea is to iteratively perturb $\mathbf{h}$ and $\mathbf{y}$ and to accept the move if the energy decreases. To perturb $\mathbf{h}$ we flip one segment at a time (between *clutter* and *non-clutter*). To perturb $\mathbf{y}$ we sample one of its four components from a Gaussian centered at its original value.

Algorithm 1 Stochastic Hill-Climbing for Inference

```

1: Input:  $\mathbf{w}, \mathbf{x}$ 
2: Output:  $\bar{\mathbf{y}}, \bar{\mathbf{h}}$ ,
3: for a number of random seeds do
4:   sample  $\bar{\mathbf{y}}$  from  $p_0(\mathbf{y})$ 
5:    $\bar{\mathbf{h}} \leftarrow \operatorname{argmin}_{\mathbf{h}} \mathbf{E}_{\mathbf{w}}(\mathbf{x}, \bar{\mathbf{y}}, \mathbf{h})$  by ICM
6:   repeat
7:     repeat
8:       perturb a parameter of  $\mathbf{y}$  as long as it decreases
       the objective
9:     until convergence
10:     $\bar{\mathbf{h}} \leftarrow \operatorname{argmin}_{\mathbf{h}} \mathbf{E}_{\mathbf{w}}(\mathbf{x}, \bar{\mathbf{y}}, \mathbf{h})$  by ICM
11:    until convergence
12: end for
```

Complexity of the algorithm depends on a number of design choices. For example, a larger number of segments (dimensionality of $\mathbf{h}$) may be able to model the clutter at a finer scale but could potentially make inference slow to converge as introducing more latent variables generally increases the number of iterations required by the ICM algorithm. In our experimental setting, the inference running time is typically one or two minutes for each image.

In our experiments, we also compare to another baseline inference method that does not make use of the continuous parametrization of $\mathbf{y}$. Specifically, we independently generate a large number of candidate boxes from $p_0(\mathbf{y})$, infer the latent variable for each of these discrete choices, and pick the one with the lowest energy.

2.3. Parameter learning

First, consider the parameters in the prior term $\mathbf{E}^0$. When we introduce the latent variables we bear in mind that they should account for the *clutter* such as chairs, desks, sofas, *etc.* However, the algorithm has no access to any supervised information on the latent variables. Given the limited training data, it is hopeless to expect the learning process to figure out the concept of *clutter* by itself. To tackle this, we introduce the prior term $\mathbf{E}^0$ to capture the concept of clutter and constrain the learning process. Specifically, the parameters in $\mathbf{E}^0$, namely α^a , α^c , and β , are determined by cross-validation on the training set and fixed throughout the learning process.

Given the training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, we learn the parameters $\mathbf{w}$ discriminatively using struct-SVM,^{17, 20} which is a

large margin based learning formulation for structured prediction problems. Let $\underline{E}_w(\mathbf{x}, \mathbf{y}) = \min_h \underline{E}_w(\mathbf{x}, \mathbf{y}, \mathbf{h})$. Our learning objective is:

$$\text{minimize}_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{m} \sum_{i=1}^m \xi_i, \text{ s.t. } \forall i, \xi_i \geq 0 \text{ and} \quad (6)$$

$$\forall i, \mathbf{y} \neq \mathbf{y}_i, \underline{E}_w(\mathbf{x}_i, \mathbf{y}) - \underline{E}_w(\mathbf{x}_i, \mathbf{y}_i) \geq 1 - \frac{\xi_i}{\Delta(\mathbf{y}, \mathbf{y}_i)}, \quad (7)$$

where $\Delta(\mathbf{y}, \mathbf{y}_i)$ is the loss function that measures the difference between the candidate output $\mathbf{y}$ and the ground truth layout $\mathbf{y}_i$. Here we use the percentage of pixels that are labeled differently by the two box layouts as the loss function.

Roughly speaking, the learning objective states that the true box layout $\mathbf{y}_i$, when accompanied with proper clutter estimation $\mathbf{h}$, should better explain the observation $\mathbf{x}_i$ than any other combinations of $\mathbf{y}$ and $\mathbf{h}$. To account for model limitations, we introduce slack variables ξ_i scaled by the loss function. This scaling has the effect of requiring a larger margin (or confidence) from hypotheses that are far from the ground truth box layout $\mathbf{y}_i$ than from hypotheses that are similar to it.

Optimizing the learning objective is difficult because the number of constraints in equation (7) is infinite. Even if we discretize the parameter space of $\mathbf{y}$ in some way, the total number of constraints is still intractably large. And each constraint involves an embedded inference problem for the latent variables. Generally this is tackled by the *cutting plane method*, i.e., gradually adding violated constraints to the optimization problem,^{12, 17} which involves an essential step of *loss augmented inference* that tries to find the output variable $\hat{\mathbf{y}}$ for which the constraint is most violated given the current parameters $\mathbf{w}$. In our problem, it corresponds to following inference problem:

$$(\hat{\mathbf{y}}, \hat{\mathbf{h}}) = \underset{\mathbf{y}, \mathbf{h}}{\operatorname{argmax}} (1 + \underline{E}_w(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}_i) - \underline{E}_w(\mathbf{x}_i, \mathbf{y}, \mathbf{h})) \cdot \Delta(\mathbf{y}, \mathbf{y}_i), \quad (8)$$

where the latent variables $\mathbf{h}_i$ should take the value that best explains the ground truth box layout under current model parameters:

$$\mathbf{h}_i = \underset{\mathbf{h}}{\operatorname{argmin}} \underline{E}_w(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}). \quad (9)$$

Equations (8) and (9) are solved by the same inference method as that we introduced in Section 2.2. However, we use a looser convergence criterion to speed up loss augmented inference as it has to be performed a large number of times in learning. The overall learning algorithm (following Tsochantaridis et al.¹⁷) is shown in Algorithm 2.

3. EXPERIMENTAL RESULTS

We experimentally verified our method on the dataset introduced by Hedau et al.⁷ The dataset consists of 314 images, each with hand-labeled box layout and clutter. It also specifies the training-test split (209 for training, 105 for test), which we use for reporting results. Performance is measured by pixel-error-rate: the percentage of misclassified

Algorithm 2 Overall Learning Procedure

```

1: Input:  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m, C, \epsilon_{final}$ 
2: Output:  $\mathbf{w}$ 
3:  $Cons \leftarrow \emptyset$ 
4:  $\epsilon \leftarrow \epsilon_0$ 
5: repeat
6:   for  $i = 1$  to  $m$  do
7:     find  $(\hat{\mathbf{y}}, \hat{\mathbf{h}})$  by solving (8) using Algorithm 2
8:     if the constraint in (7) corresponding to  $(\hat{\mathbf{y}}, \hat{\mathbf{h}})$  is
        violated more than  $\epsilon$  then
9:       add the constraint to  $Cons$ 
10:    end if
11:  end for
12:  update  $\mathbf{w}$  by solving the relaxed QP (6) given  $Cons$ 
13:  for  $i = 1$  to  $m$  do
14:    update  $\mathbf{h}_i$  by solving (9)
15:  end for
16:  if number of new constraints in last iteration is less
    than threshold then
17:     $\epsilon \leftarrow \epsilon/2$ 
18:  end if
19: until  $\epsilon < \epsilon_{final}$  and number of new constraints in last
    iteration is less than threshold

```

pixels in the task of classifying them into one of the five classes. Our approach achieves an error-rate of 20.1% *without* clutter labels, compared to 26.5% in Hedau et al.⁷ *without* clutter labels and 21.2% with clutter labels. Details are shown in Table 1.

Each row in Table 1 shows a different performance metric and each column represents a different algorithm. Briefly we have: *Row 1*: pixel error rate. *Row 2 and 3*: the number of test images (out of 105) with pixel error rate under 20% and 10%. *Column 1*: Hoiem et al.’s algorithm. *Column 2*: Hedau et al.’s method without clutter label. *Column 3*: Hedau et al.’s method with clutter label. The first three columns are directly copied from Hedau et al.⁷ *Column 4*: Our method (without clutter label). The remaining columns will be explained below.

In order to validate the effects of prior knowledge in constraining the learning process, we take out the prior knowledge by adding the two terms E^a and E^c as ordinary features and try to learn their weights. The performance of recovering box layouts in this case is shown in Table 1, column 5 (labeled “without prior”). Although the difference between columns 4 and 5 is small, there are many cases where recovering more reasonable clutter does help in recovering the correct box layout.

One typical example is shown in Figure 3. In Figure 3(a), we can see that the boundary between the *floor* and the *front-wall* (the wall on the right) is correctly recovered even though it is largely occluded by the bed, which is correctly inferred as “clutter”, and the boundary is probably found by the appearance difference between the floor and the wall. However, in the model learned without prior constraints, the bed is regarded as non-clutter whereas major parts of the floor and walls are

Table 1. Quantitative results.

	Hoiem et al. ¹⁰	Hedau et al. ⁷ without	Hedau et al. ⁷ with	Ours without	without prior	$h = 0$	$h = GT$	cheat
Pixel	28.9%	26.5%	21.2%	$20.1 \pm 0.5\%$	$21.5 \pm 0.7\%$	$22.2 \pm 0.4\%$	$24.9 \pm 0.5\%$	$19.2 \pm 0.6\%$
$\leq 20\%$	—	—	—	62 ± 3	58 ± 4	57 ± 3	46 ± 3	67 ± 3
$\leq 10\%$	—	—	—	30 ± 3	24 ± 2	25 ± 3	20 ± 2	37 ± 4

See text for explanation.

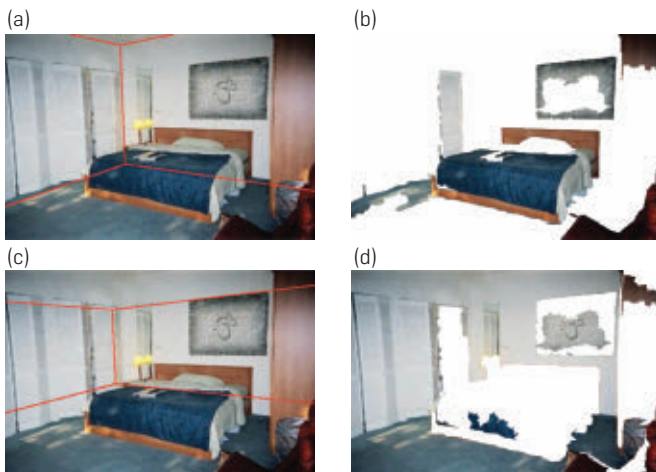
inferred as clutter (this is probably because the term E^c is not acting effectively with the learned weights), so it appears that the boundary between the *floor* and the *front-wall* is decided incorrectly due to the strong contrast between the white pillow and blue sheet.

More examples of inference result are shown in Figure 4.

In a second experiment, we fixed the latent variables h to be all zeros (i.e., assuming no clutter). The results are shown in column 6 of Table 1 (labeled $h = 0$). The no clutter assumption gives rise to an accuracy of 22.2%, a 2% reduction over our model. However, it still considerably improves upon the 26.5% accuracy obtained by Hedau et al. in the case of no clutter labels. We note, however, that in this analysis, Hedau et al. only used “perspective consistency” features. When using the clutter labels during training, however, Hedau et al. incorporated other kinds of features and the supervised surface label classification method in Hoiem et al.¹⁰ By fixing h to be all zeros we decompose our performance improvement upon Hedau et al.⁷ into two parts: (i) using the richer set of features, and (ii) accounting for clutter with latent variables. Although the improvement brought by the richer set of features is larger, the effect of accounting for clutter is also significant.

We also tried to evaluate a supervised learning approach, in which we fix the latent variables h to be the hand-labeled clutter layout.

Figure 3. Example result of recovering box and clutter layout. The clutter layouts are shown by removing all non-clutter segments.
 (a) Inferred box layout using model learned *with* prior knowledge.
 (b) Inferred clutter layout using model learned *with* prior knowledge.
 (c) Inferred box layout using model learned *without* prior knowledge.
 (d) Inferred clutter layout using model learned *without* prior knowledge.



The results are shown in column 7 of Table 1 (labeled $h = GT$). Somewhat surprisingly, the results are considerably worse than those obtained by our model, and even worse than assuming no clutter ($h = 0$). To understand why, we quantitatively compare our recovered clutter to the hand-labeled clutter, and see that the average pixel difference is around 30% on both, the training and test set. A closer examination, shown in Figure 5, demonstrates the difference between the hand-labeled clutter and the clutter recovered by our method (on the test set). Generally, the hand-labels include much less clutter than our algorithm recovers. Because delineating objects by hand is very time consuming, usually only one or two pieces of major furniture are labeled as clutter. Some salient clutter is missing in the hand-labels such as the cabinet and the TV in the image of the 1st row, the smaller sofa in the image of the 5th row, and nothing is labeled in the image of the 3rd row. Therefore, it is not surprising that learning with hand-labeled clutter does not result in a better model. We also tried to fix the latent variable to be the hand-labeled clutter in *both* learning and inference. Note that the algorithm is actually “cheating” as it has access to the labeled clutter even in the testing phase, so it should be expected to perform well. Surprisingly, it only gives slightly better results (Table 1, column 8, labeled “cheat”) than our method. It is also worth noting that there is around 6–7% (out of the 20.1%) of pixel error due to incorrect vanishing point detection results. The error rate of 6–7% is estimated by assuming a perfect model that always picks the best box generated from the vanishing point detection result, and performing stochastic hill-climbing to infer the box using the perfect model. The rest of the error attributes to the limitation of the model and inference algorithm. For example, the features we used do not perfectly characterize the structure of an indoor scene.

We compare our inference method (Algorithm 2) to the baseline method (of evaluating hypotheses independently) described in Section 3.2. Figure 6 shows the average pixel error rate over test set versus the number of calls (in log-scale) to the joint feature mapping Ψ , which is an indication of running time.

The actual running time of the inference algorithm depends on the number of random starts and convergence criteria. In our current experiments we use 50 random re-starts and run a maximum of 100 inner loop iterations. It takes on average 16 seconds to run inference for one image, 25% of which are for performing ICM on latent variables.

In Figure 7 we show the performance of the learned model on test set versus the number of iterations in learning. Empirically the learning procedure attains a small

Figure 4. More results for comparing learning with and without prior constraints. The 1st and 2nd columns are the result of learning with prior constraints. The 3rd and 4th columns are the result of learning without prior constraints. In many cases, recovering more reasonable clutter does help in recovering the correct box layout.



error rate after a small number of iterations, and then fluctuates around this error rate due to the approximate loss-augmented inference step of learning.

4. RELATED WORK

Our method is closely related to a recent work of Hedau et al.⁷ We adopt their idea of generating box layouts from the vanishing points. However, they use supervised classification of surface labels¹⁰ to identify clutter (furniture), and use the trained surface label classifier to iteratively refine the box layout estimation. Specifically, in each iteration they use the estimated box layout to add features to supervised surface label classification, and then use the classification result to lower the contribution of “clutter” image regions in estimating the box layout. Thus, their method requires the user to label clutter regions in the training set.

Latent variables have been exploited in the computer vision literature in various tasks such as object detection,

recognition and segmentation. They can be used to represent visual concepts such as occlusion,¹⁸ object parts,³ and image-specific color models.¹⁵ Introducing latent variables into structured prediction was shown to be effective in several applications.²⁰ An interesting aspect of our work is that latent variables are used in direct correspondence with a concrete visual concept (clutter in the room).

Since the publication of our initial work,¹⁹ reasoning about the 3D geometry and semantics of scenes has been further addressed in many recent works such as Geiger et al.,⁴ Gupta et al.,⁶ Hedau et al.,⁸ Lee et al.,¹³ Pepik et al.,¹⁴ and Tsai et al.¹⁶ to name a few, which have demonstrated obtaining more fine-grained 3D context information or using it to help other vision tasks such as object detection.

5. DISCUSSION

In this paper we addressed the problem of recovering the geometric structure as well as clutter layout from a single

Figure 5. Sample results for comparing the recovered clutter by our method and the hand-labeled clutter in the dataset. The 1st and 2nd columns are recovered box and clutter layouts by our method. The 3rd column (right) is the hand-labeled clutter layouts. Our method usually recovers more objects as “clutter” than people would bother to delineate by hand. For example, the rug with a different appearance from the floor in the 2nd image, paintings on the wall in the 1st, 4th, 5th, and 6th images, and the tree in the 5th image. There are also major pieces of furniture that are missing in the hand-labels but recovered by our method, such as the cabinet and TV in the 1st image, everything in the 3rd image, and the small sofa in the 5th image.

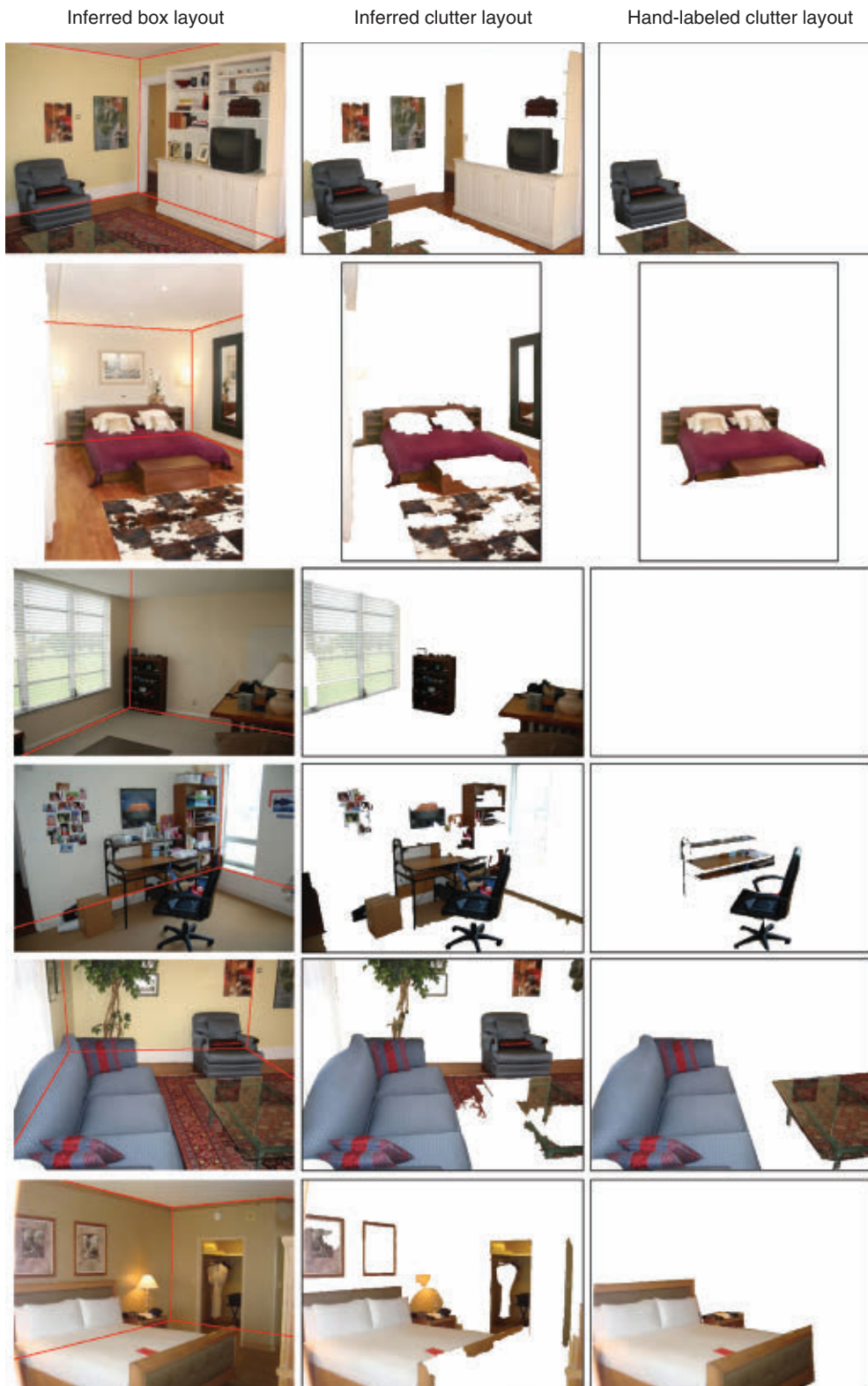


Figure 6. Comparison between the inference method described in Algorithm 2 and the baseline inference method that evaluates hypotheses independently.

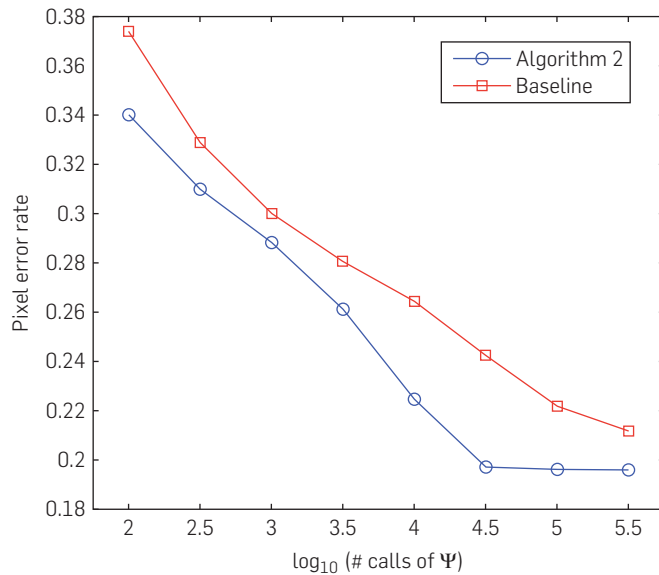


Figure 7. Empirical convergence of the learning procedure.

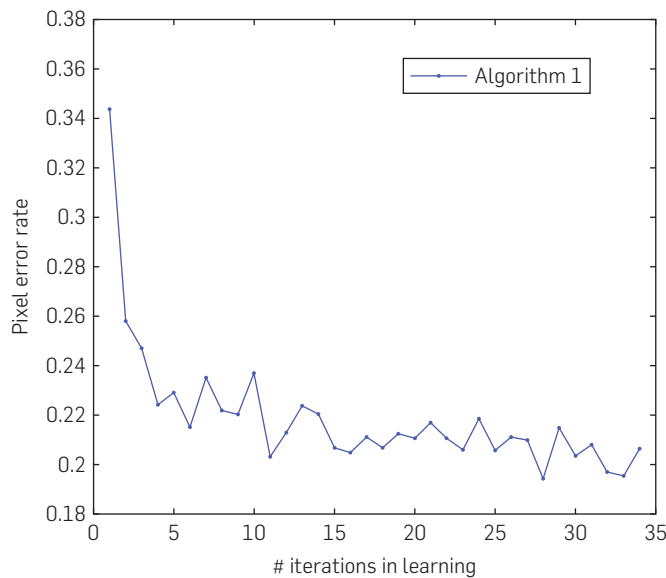


image. We used latent variables to account for indoor clutter, and introduced prior terms to define the role of latent variables and to constrain the learning process. Our approach, without using clutter labels in training, outperforms a baseline method that does use them.

This improvement can be attributed to three main technical contributions: (i) we introduce latent variables and the prior terms to account for the clutter in a principled manner; (ii) we design a rich set of joint features to capture the compatibility between image and the box-clutter layouts; and (iii) we perform more efficient and accurate inference by making use of the parametrization of the “box” space.

Learning latent variable based models is known to be susceptible to local optima of the learning objective. A lesson from this paper that we believe could be useful is that, imposing prior knowledge on how the model should function (using fixed prior energy terms in our case) can help guide the learning process to a desirable region of the parameter space, and thereby give rise to a better model.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. RI-0917151, the Office of Naval Research under the MURI program (N000140710747) and the Boeing Corporation. 

References

- Besag, J. On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc. B-48*, 3 (1986), 259–302.
- Comaniciu, D., Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 5 (2002), 603–619.
- Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 9 (2010), 1627–1645.
- Geiger, A., Wojek, C., Urtasun, R. Joint 3D estimation of objects and scene layout. In *NIPS* (2011), 1467–1475.
- Gould, S., Fulton, R., Koller, D. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2009).
- Gupta, A., Satkin, S., Efros, A.A., Hebert, M. From 3D scene geometry to human workspace. In *Computer Vision and Pattern Recognition (CVPR)* (2011).
- Hedau, V., Hoiem, D., Forsyth, D. Recovering the spatial layout of cluttered rooms. In *ICCV* (2009).
- Hedau, V., Hoiem, D., Forsyth, D. Thinking inside the box: Using appearance models and context based on room geometry. *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, eds. Volume 6316 of *Lecture Notes in Computer Science*. (2010), Springer, 224–237.
- Heitz, G., Koller, D. Learning spatial context: Using stuff to find things. In *ECCV* (2008), 1: 30–43.
- Hoiem, D., Efros, A.A., Hebert, M. Recovering surface layout from an image. *Int. J. Comput. Vis.* 75, 1 (Oct. 2007), 151–172.
- Hoiem, D., Efros, A.A., Hebert, M. Closing the loop in scene interpretation. In *CVPR* (2008), 1–8.
- Joachims, T., Finley, T., Yu, C.N.J. Cutting-plane training of structural SVMs. *Mach. Learn.* 77, 1 (2009), 27–59.
- Lee, D., Gupta, A., Hebert, M., Kanade, T. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Advances in Neural Information Processing Systems* (2010), J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds. Volume 23, 1288–1296.
- Pepik, B., Stark, M., Gehler, P., Schiele, B. Teaching 3d geometry to deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).
- Shotton, J.D.J., Winn, J., Rother, C., Criminisi, A. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vis.* 81, 1 (Jan. 2009), 2–23.
- Tsai, G., Xu, C., Liu, J., Kuipers, B. Real-time indoor scene understanding using Bayesian filtering with motion cues. In *ICCV* (2011), IEEE, 121–128.
- Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y. Large margin methods for structured and interdependent output variables. *JMLR* 6 (2005), 1453–1484.
- Vedaldi, A., Zisserman, A. Structured output regression for detection with partial truncation. In *Advances in Neural Information Processing Systems* (2009).
- Wang, H., Gould, S., Koller, D. Discriminative learning with latent variables for cluttered indoor scene understanding. In *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, eds. Volume 6314 of *Lecture Notes in Computer Science* (2010), Springer, 497–510.
- Yu, C.N.J., Joachims, T. Learning structural svms with latent variables. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM, New York, NY, USA, 1169–1176.

Huayan Wang (huayanw@cs.stanford.edu), Computer Science Department, Stanford University.

Daphne Koller (koller@cs.stanford.edu), Computer Science Department, Stanford University.

Stephen Gould (stephen.gould@anu.edu.au), School of Computer Science, Australian National University.

CAREERS

California State University – San Bernardino School of Computer Science and Engineering Assistant Professor

The School of Computer Science and Engineering invites applications for a tenure track position at the Assistant Professor level. Candidates must have a Ph.D. in Computer or Electrical Engineering, or a closely related field. We are particularly interested in candidates with strengths in embedded systems or signal processing. Other areas of computer engineering will also be considered. The position is primarily to support the B.S. in Computer Engineering program which is in the process of seeking ABET accreditation. The program has strong support from the local industry and government entities. In addition, the school offers the degrees B.S. in Computer Science (ABET accredited), B.S. in Bioinformatics, B.A. in Computer Systems, and M.S. in Computer Science. The candidate must display potential for excellence in teaching and scholarly work. The candidate is expected to supervise student research at both the undergraduate and graduate levels, and to actively participate in other types of academic student advising. The candidate will actively contribute to the School's curriculum development, and serve on committees at the School, College and University levels. Women and underrepresented minorities are strongly encouraged to apply. For more information about the School of Computer Science and Engineering, please visit <http://cse.csusb.edu>. DEADLINE AND APPLICATION PROCESS: April 15, 2013 or until filled. Submit a curriculum vitae with letter of application that includes statement on teaching philosophy, and research interests. Also submit the names, telephone and fax numbers, and e-mail address of three references, along with an official copy of most recent transcripts. The position will start in September 2013. PLEASE SEND ALL MATERIALS TO: Dr. Kerstin Voigt, Director, School of Computer Science and Engineering, 5500 University Parkway, San Bernardino, CA 92407-2393, Telephone: (909) 537-5326, email: kvoigt@csusb.edu.

Maharishi University of Management Computer Science Dept Assistant/Associate Professor of Computer Science

The Computer Science Department at Maharishi University of Management invites applications for a full-time faculty position beginning Fall 2013. Qualifications include Ph.D. in Computer Science (or closely related area), or M.S. and seven years of professional software development experience. Candidates will be considered for Assistant, Associate, or full Professor depending on experience and qualifications.

The primary responsibility is teaching computer science courses at the MS level. Applications will be reviewed as they are received until the position is filled. To apply, email curriculum

vitae (pdf file) to cssearch2011@mum.edu.

For further information, see <http://www.mum.edu/> and <http://mscs.mum.edu/>. MUM is located in Fairfield, Iowa, and is an equal opportunity employer.

North Carolina State University Department of Computer Science Assistant/Associate/Full Professor

The Department of Computer Science (<http://www.csc.ncsu.edu>) at North Carolina State University in partnership with the Chancellor's Faculty Excellence (CFE) Program invites applications for tenure track positions at the senior Assistant, Associate, and Full Professor levels. NC State CFE Program (<http://ncsu.edu/workthatmatters>) is a cluster hire program that marks the first major initiative of the university's 2011-2020 strategic plan - "The Pathway to the Future." Starting in 2012, NC State will hire thirty-eight faculty members across twelve research areas, or "clusters," to promote interdisciplinary scholarship and the development of innovative curricula in emerging areas of strategic strength. Inclusiveness, interdisciplinary collaboration and diversity are academic and position imperatives and university goals. More information is available on the CFE Program site and the Computer Science Department site <http://www.csc.ncsu.edu/employment>

Digital Transformation of Education: This cluster is hiring four tenure-track or tenured faculty in computer science, psychology and education. The two computer science positions focus on *intelligent learning environments* and *adaptive game technologies*. We invite applications from nationally recognized researchers who are engaged in innovative and transformative scholarship that will further NC State's position as a leader in research and development on educational innovation. Successful applicants will work closely with the other cluster faculty, and with faculty in the Digital Games Research Center, the Center for Education Informatics, and in the Friday Institute for Educational Innovation (<https://www.fi.ncsu.edu>). Further information about all four Digital Transformation of Education positions, including application instructions, is available at <http://go.ncsu.edu/dte>.

Data-Driven Science: NC State is in the process of aggressively expanding its large-scale ("Big Data") analytics and data-driven sciences facilities, research programs, and faculty. This cluster is hiring four faculty members. It is anticipated that up to two positions will be homed in the Computer Science Department. One position focuses on text analytics including natural language processing, spoken language systems, information retrieval, word sense disambiguation, language understanding or machine learning. Second position focuses on methods for storing, manipulating and accessing large-scale data sets. Successful appli-

cants will work closely with the other cluster faculty, and with the faculty affiliated with the Institute for Next Generation Information Technology Systems (<https://www.itng.ncsu.edu/>), and Institute for Advanced Analytics (<http://analytics.ncsu.edu>). Further information about all four positions, including application instructions, is available at (<https://jobs.ncsu.edu/postings/8516>).

Geospatial Analytics: NC State will create a unique interdisciplinary research and PhD program in Geospatial Analytics to address the extensive needs in both basic geospatial sciences and the corresponding computer science and mathematical modeling disciplines. This cluster is hiring three faculty members. It is anticipated that one position will be homed in the Computer Science Department. Successful applicant will work closely with the other cluster faculty, and with the faculty affiliated with the Information Science and Technology research and teaching programs (GIS.ncsu.edu). To apply, please visit <https://jobs.ncsu.edu> and designate position number 00102987 Academic contact: Dr. Hugh Devine (Hugh_Devine@ncsu.edu).

NC State University is an equal opportunity and affirmative action employer. All qualified applicants will receive consideration for employment without regard to race, color, national origin, religion, sex, age, veteran status, or disability. In addition, NC State University welcomes all persons without regard to sexual orientation or genetic information. We welcome the opportunity to work with candidates to identify suitable employment opportunities for spouses or partners. Persons with disabilities requiring accommodations in the application and interview process please call (919) 515-3148.

Platphorm, LLC. Lead Software Engineer

Work in San Francisco, CA or telecommute. Pls send cover letter & resume to jobs@platphormcorp.com

Requires MS in Comp Sci or related field. Must also possess coursework/exp background in Ruby on Rails, Javascript, HTML and CSS, MySQL db programming, Java programming, and high performance & highly scaled system.

Portland State University Faculty Positions in Design Verification/Validation Assistant/Associate Professor (Tenure-Track)/ Senior Instructor/Assistant Professor (Fixed-Term)

The Electrical and Computer Engineering Department at Portland State University seeks candidates for tenure-track and non-tenure track fixed-term faculty in design verification/validation (DV).

The tenure-track position requires excellent teaching and research skills. The candidate will build and lead a research program in DV, collaborating with local industry, other faculty, and worldwide EDA leaders.



جامعة الملك عبد الله
للعلوم والتقنية
King Abdullah University of
Science and Technology

FACULTY POSITION: DIRECTOR OF GEOMETRIC MODELING AND SCIENTIFIC VISUALIZATION CENTER

The Geometric Modeling and Scientific Visualization Center (GMSV) (<http://gmsv.kaust.edu.sa>) at King Abdullah University of Science and Technology (KAUST) is seeking a leading scientist in visual computing as Director. The Center is part of the Computer, Electrical, and Mathematical Sciences and Engineering (CEMSE) Division, and the associated faculty appointment will be for Full Professor either in Computer Science or Applied Mathematics.

The Director manages an interdisciplinary center with a multi-million dollar annual portfolio whose members have dedicated extraordinary research facilities for simulation, visualization and immersive environments, generous research space, and stable funding to support the Director, faculty, postdocs, and graduate students. In terms of facilities, the center has prime access to the CORNEA Visualization center and the university has a related supercomputer (Shaheen). The successful candidate will be an internationally recognized leader in visual computing. The center has a research portfolio in areas such as computer graphics, computational design, computer vision, image processing, data visualization, data analysis and understanding, and high-performance scientific visualization. With currently 65 members, it is still undergoing significant growth and has exceptional opportunities through multi-disciplinary interactions with other KAUST research thrusts in extreme scale computing, computational biology, materials science, marine sciences, water desalination, and solar energy, to name a few.

All candidates should have the ability to pursue a high impact research program and have a commitment to teaching at the graduate level. Applicants should apply at <http://apptkr.com/316454>. Applications received by March 31, 2013 will receive full consideration and the position will remain open until filled.

King Abdullah University of Science and Technology (KAUST) is an international, graduate research university dedicated to advancing science and technology through interdisciplinary research, education, and innovation. Located on the shores of the Red Sea in Saudi Arabia, KAUST offers superb research facilities, and internationally competitive salaries. The university attracts top international faculty, scientists, engineers, and students to conduct fundamental and goal-oriented research to address the world's pressing scientific and technological challenges related to the sustainability of water, food, energy, and the environment.

The ideal fixed-term position requires a passion for teaching with proven teaching skills, a desire to develop compelling curriculum in DV, and industry experience in verification of large-scale systems. Expertise with contemporary DV methodologies is required. Expertise in hardware emulation is a plus.

Additional information and requirements for applying are at <http://pdx.edu/hr/faculty-administrative-openings>. **Positions #D93193 and #D93195.**

University of Illinois Springfield Assistant Professor Computer Science

The Computer Science Department at the University of Illinois at Springfield (UIS) invites applications for 2 beginning assistant professor, tenure track positions to begin August, 2013. A Ph.D. in Computer Science or closely related field is required. The position involves graduate and undergraduate teaching, supervising student research, and continuing your research. Many of our classes are taught online. All areas of expertise will be considered, but the ability to teach core computer science is of special interest for the Department. Review of applications will begin on March 25, 2013 and continue until the position is filled or the search is terminated. **Please send your vita and contact information for three references to Chair Computer Science Search Committee; One University Plaza; UHB 3100; Springfield, IL 62703-5407.**



Florida Institute of Technology
High Tech with a Human Touch™

Florida Institute of Technology offers a master's program and Ph.D. program in Human-Centered Design (HCD).



- Candidates with backgrounds and degrees in engineering, science and human factors, as well as arts and architecture are encouraged to apply.
- A graduate degree in HCD supports independent scholarly work, opportunities in academia or pursuit of advanced research and leadership in government, industry and business.
- Current research is in: cognitive engineering, life-critical systems, complexity analysis for HCD, human-centered organization design and management, modeling and simulation, advanced interaction media, creativity and design thinking, functional analysis, industrial design, and usability engineering.
- Internationally connected with best research and professional institutions in HCD.

For more information:
(321) 309-4960 • dcaballe@fit.edu
Visit: <http://research.fit.edu/hcdi>
150 W. University Blvd., Melbourne, FL 32901

EN-109-213

Located in the state capital, the University of Illinois Springfield is one of three campuses of the University of Illinois. The UIS campus serves approximately 5,000 students in 23 undergraduate and 21 graduate degree programs. The academic curriculum of the campus emphasizes a strong liberal arts core, an array of professional programs, extensive opportunities in experiential education, and a broad engagement in public affairs issues of the day. The campus offers many small classes, substantial student-faculty interaction, and a rapidly evolving technology enhanced learning environment. Its diverse student body includes traditional, non-traditional, and international students. Twenty-five percent of majors are in 17 undergraduate and graduate online degree programs and the campus has received several national awards for its implementation of online learning. UIS faculty are committed teachers, active scholars, and professionals in service to society. You are encouraged to visit the university web page at <http://www.uis.edu> and the department web page at <http://csc.uis.edu>. UIS is an affirmative action/equal opportunity employer with a strong institutional commitment to recruitment and retention of a diverse and inclusive campus community. Women, minorities, veterans, and persons with disabilities are encouraged to apply.

APPLY FOR THIS JOB

Contact: Chair Computer Science Search Committee
Email Address: csc@uis.edu
Phone: 217-206-6770

University of Tulsa Tandy Endowed Chair in Cyber Security

The Tandy School of Computer Science at the University of Tulsa is seeking a candidate to fill the Tandy Endowed Chair in Cyber Security. Applicants should have a distinguished record in research, education, and service in Cyber Security, Information Assurance, or a related area at the rank of full professor. The applicant should be open to collaborative and multi-disciplinary research activities. Responsibilities will include spearheading the development of new research and curriculum areas within the school and continuing to advance the international recognition of the University of Tulsa in this field. Applicants should possess a PhD or equivalent in a closely related field for this tenure track position.

The University of Tulsa is a private university with approximately 4500 undergraduate, graduate, and law students. The Tandy School of Computer Science occupies the second floor of the new J. Newton Rayzor Hall dedicated in Nov. 2011. The School offers a B.S., M.S. and Ph.D. in Computer Science. The National Security Agency and U.S. Cyber Command have designated The University of Tulsa as a National Center of Academic Excellence in Cyber Operations. The University of Tulsa's information security programs have previously received similar nods of approval from the NSA, National Science Foundation, Department of Defense and U.S. Secret Service. The Tandy School of Computer Science houses TU's Cyber Corps Program which currently has 60

students from a variety of backgrounds including computer science, mathematics, electrical engineering, chemical engineering, mechanical engineering, law and business.

Tulsa is located in northeast Oklahoma in "Green Country," a region of rolling hills, lakes and wooded landscapes. With a metropolitan population of approximately one million, the city offers cosmopolitan amenities while maintaining the livability of a more modest urban center. Tulsa offers diverse arts, entertainment, and recreation venues appealing to young adults and families.

To apply, please send CV, teaching statement and research statement, and contact information for four references as a single PDF by e-mail to Dr. Rose Gamble, Chair of the Search Committee at gamble@utulsa.edu. Apply URL: <http://utulsa.edu>

The University and Tandy School of Computer Science share a strong commitment to achieving diversity among faculty and staff. We particularly encourage applications from underrepresented groups. The University of Tulsa is an Equal Opportunity/Affirmative Action Employer.

University of Wisconsin-Platteville Computer Science & Software Engineering Assistant Professor

Position starting August 20, 2013. See <http://www.uwplatt.edu/pers/faculty.htm> for complete announcement and application instructions. Application review begins April 2, 2013. AA/EEO employer.



THE CHINESE UNIVERSITY OF HONG KONG



Applications are invited for:-

Faculty of Engineering Professors / Associate Professors / Assistant Professors (Ref. 1213/124(255)/2)

The Faculty of Engineering invites applications for several faculty posts at Professor / Associate Professor / Assistant Professor levels with prospect for substantiation in the interdisciplinary area of 'Big Data Analytics', which is a new strategic research initiative supported by the University Focused Investments Scheme and will complement current/planned strengths in different Departments under the Faculty.

Currently, the Faculty is seeking candidates in the following areas:

- Theoretical, mathematical and algorithmic aspects in large data analytics;
- Large scale software systems and architecture in large data analytics;
- Application areas in large data analytics (including information analytics, network/Web analytics, financial analytics or security analytics, etc.).

Applicants should have (i) a PhD degree; and (ii) a strong scholarly record demonstrating potential for teaching and research excellence. The appointees will be expected to (a) teach both undergraduate and postgraduate courses; (b) develop a significant independent research programme with external funding; and (c) supervise postgraduate students.

Appointments will normally be made on contract basis for two to three years initially, which, subject to performance and mutual agreement, may lead to longer-term appointment or substantiation later. Applications will be accepted until the posts are filled. Further information about the Faculty is available at <http://www.erg.cuhk.edu.hk>.

Salary and Fringe Benefits

Salary will be highly competitive, commensurate with qualifications and experience. The University offers a comprehensive fringe benefit package, including medical care, a contract-end gratuity for appointments of two years or longer, and housing benefits for eligible appointees. Further information about the University and the general terms of service for appointments is available at <http://www.per.cuhk.edu.hk>. The terms mentioned herein are for reference only and are subject to revision by the University.

Application Procedure

Please send full resume, copies of academic credentials, publication list with abstracts of selected published papers, details of courses taught and evaluation results (if any), a research plan, a teaching statement, together with names of three to five referees, to the Dean, Faculty of Engineering by e-mail to recruit-bda@erg.cuhk.edu.hk.

For enquiries, please contact Professor John C.S. Lui, the leader of this strategic initiative (e-mail: cslui@cse.cuhk.edu.hk). Applicants are requested to clearly indicate that they are applying for the position under 'Big Data Analytics Initiative'. The Personal Information Collection Statement will be provided upon request. Please quote the reference number and mark 'Application – Confidential' on cover.

[CONTINUED FROM P. 104] “You mean, like houses?” she said, “but that’s a job for bots—”

“No, like a real artist, like Goya or Picasso or Wyeth. It’s already programmed into my synthetic neural circuits, awaiting activation all this time. I *see* the image *before* brush touches canvas. Such colors, such compositions. But, there are these...” I held out my paws.

She looked dubious but not without hope. “Well, under the law, we do offer surgical modification of pre-existing conditions. But first let’s analyze your genome and physiology.”

Later, presented with the Department’s report, my dreams crumbled again.

“According to the report,” the social worker said, “you shouldn’t even be alive. Your cells exhibit six different suites of synthetic amino acids, arrayed in triple-helix form, with a dozen encoding and transcription systems. Your ribosomes are twice as complex as baseline standards, and you have not one but three enteric nervous systems, not to mention multiple distributed ganglia. The geneticists say any kind of hand transplant would be rejected, no matter what kind of immunosuppressive therapy they could provide.”

“There’s no hope then?” I said, venting a modest honk from my sniffling snout.

Her face assumed the determined expression of the lead soldier in Frederic Remington’s “The Cavalry Charge.” “Easy, Nibbles,” she said, “perhaps we can track down your original bioengineer...”

“But,” I said, “he lost interest the moment he consigned me to the administrators. I don’t even know his name.”

“Leave that to the forensic analysts,” she said. “They can follow any bio- or neurological clue, even those embedded in your deepest synthetic unconscious.”

They were indeed able to locate him at his facility intently reengineering a synthetic mycoplasma genome, though his hobby was and always would be creating chimeras like me.

Our reunion was in the Bureau’s clinical center where my fear now turned to anger. He expressed no re-

My most distinctive (and problematic) feature, especially in light of my professional ambition, were my hands, which were really just big paws, like a leopard’s, no more capable of sewing a stitch than the pincers of a mindless bot crushing ore in an open pit.

gret for making me the way he did or for sending me away. He said only, “Those ears. What could I have been thinking? I do like the paws, though. Shame they have to go.”

But, but... that means I can have real hands instead?

I awoke following the surgery to find they blended nicely with the rest of my otherwise cobbled appearance. Not quite human—a raccoon would be proud—but refined enough to hold a brush and palette, of proper scale.

Now, on the eve of my first solo show, I, your humble Nibbles, recall my early despair with mixed feelings. Who would have imagined my very own social worker would consent to pose for my now-acclaimed portrait, “Naked Maja—With Horns,” sold already for more than my board-certified bioengineer makes in even a good year. 

Paul Di Filippo (pgdf@cox.net) has been a professional science fiction writer for more than 30 years. His book reviews appear regularly in *The Barnes & Noble Review* (<http://bnreview.barnesandnoble.com/?>). His latest book, with co-author Damien Broderick, is *Science Fiction: The 101 Best Novels 1985–2010*.

© 2013 ACM 0001-0782/13/04



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



From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/2436256.2436277

Paul Di Filippo

Future Tense Modified Is the New Normal

How I transcended the baseline for the sake of art and bioengineering.

I WAS ON edge going into the interview with my social worker from the Chimera Reassignment Bureau, finally facing the prospect of manumission and full citizenship. Unknown was how I would live up to my new privileges, considering I had been no more than property (by law) my entire life—abandoned property at that.

I was hyperconscious of my appearance, especially around the normals, considering how much I deviated from the baseline. Even among my fellows, in all their telltale variety, I was odd.

My face, covered with fur, like the rest of me, featured eyes proportionately as wide and alert as those of a tarsier, with acute night vision, due to syntho-geno-pheno design. Much of my remaining facial area was dominated by a prehensile snout as pointy as a tapir's. I sported fangs, despite being herbivorous, also by design. Tall supersensitive omnidirectional ears like a jackrabbit's sprouted from somewhere above my eyes. My expressive tail forked three-quarters of the way down its hyperarticulated length, the better to dangle from tree limbs when picking fruit, rendering me relatively self-sufficient, also by design. However, my most distinctive (and problematic) feature, especially in light of my professional ambition, were my hands, which were really just big paws, like a leopard's, no more capable of sewing a stitch than the pincers of a mindless bot crushing ore in an open pit.

Seated now, surrounded by my fellow chimeras, all clearly rendered by the hands of undisciplined bio-hack-



ers, I despaired of ever being understood or achieving my dream of producing art on canvas through a brush or on screen through code.

My mood was lifted a bit by the sound of a springy one-wheeled bot sing-song calling my name and ushering me into an office. In these tight quarters I could smell my own anxiety, despite the aroma having been pre-bioengineered to overcome any potential social revulsion, especially among the normals. My social worker took no notice, not even a wrinkled nose, standing instead to shake my hands, such as they were.

"Nibbles," she said, "I'm here to spell out your new rights and respon-

sibilities and enroll you in the programs that will help you transition to productive citizenship among the normal population." That is, at least they'll ignore me, I thought, mercifully.

She read me the text of the National Chimera Emancipation Act—I had read it myself a hundred times already—pinch-poked and swiped through a series of augmented-reality presentations with me, then asked the question I had been dreading:

"So, Nibbles, what kind of career and gainful employment do you see for yourself, given your aptitudes?"

"I want to be a painter," I said confidently. [CONTINUED ON P. 103]

Computing Reviews presents

The Best Reviews and
Notable Books & Articles
of 2012

Coming in April
online and in print

computingreviews.com

A daily snapshot of what is new and hot in computing.



Your **brain** will be **delighted**.
Both sides.

Find yourself among thousands of techno-enthusiasts as you engage in a dazzling array of mind-expanding programs, informative sessions, and blockbuster events showcasing the latest in computer graphics and interactive techniques.



SIGGRAPH2013
Left Brain + Right Brain

The **40th** International
Conference and **Exhibition**
on **Computer Graphics** and
Interactive Techniques

Conference 21–25 July 2013
Exhibition 23–25 July 2013
Anaheim Convention Center



Sponsored by ACM **SIGGRAPH**

www.siggraph.org/s2013

