

# COMMUNICATIONS

ACM.ORG/CACM

OF THE

# ACM

02/2011 VOL.54 NO.2

## **Vision-Based Hand-Gesture Applications**

Against Cyberterrorism

Finding Usability  
Bugs with  
Automated Tests

Still Building Memex

ACM Launches New  
Digital Library



# 2011 International Conference on Software Engineering

Waikiki, Honolulu, Hawaii, USA • May 21-28, 2011

[2011.icse-conferences.org](http://2011.icse-conferences.org)

## Aloha!

ICSE is the premier software engineering conference, providing a forum for researchers, practitioners, and educators to present and discuss the most recent innovations, trends, experiences and concerns in the field of software engineering. The conference features the presentation of research papers, technical briefings, workshops, research demonstrations, and updates on the use of advanced software engineering techniques in industry. ICSE 2011 is the ideal venue for meeting and learning from like-minded colleagues from around the world. The ICSE 2011 theme of **Software by Design** reflects the widely-held view that the most important ingredient in ensuring a software system's long-term success is its design.

### Main Conference Tracks

Research/Technical Track - over 60 papers to be presented  
Software Engineering in Practice  
New Ideas and Emerging Results  
Demonstrations  
Impact Project Focus Area

### Highlights

24 Workshops & 4 Co-Located Events before and after the main conference  
Technical Briefings - 1 day/3 tracks of presentations on state-of-the-art topics  
Doctoral Symposium  
New Faculty and Researchers Symposium  
Festschrift  
ACM Student Research Competition  
Student Contest on Software Engineering (SCORE)

### General Chair

Richard N. Taylor - Univ. of California, Irvine

### Program Committee Chairs

Harald Gall - Univ. of Zurich

Nenad Medvidović - Univ. of Southern California

### Sponsored By





# ACM TechNews Goes Mobile

## iPhone & iPad Apps Now Available in the iTunes Store

ACM TechNews—ACM's popular thrice-weekly news briefing service—is now available as an easy to use mobile apps downloadable from the Apple iTunes Store.

These new apps allow nearly 100,000 ACM members to keep current with news, trends, and timely information impacting the global IT and Computing communities each day.



### TechNews mobile app users will enjoy:

- **Latest News:** Concise summaries of the most relevant news impacting the computing world
- **Original Sources:** Links to the full-length articles published in over 3,000 news sources
- **Archive access:** Access to the complete archive of TechNews issues dating back to the first issue published in December 1999
- **Article Sharing:** The ability to share news with friends and colleagues via email, text messaging, and popular social networking sites
- **Touch Screen Navigation:** Find news articles quickly and easily with a streamlined, fingertip scroll bar
- **Search:** Simple search the entire TechNews archive by keyword, author, or title
- **Save:** One-click saving of latest news or archived summaries in a personal binder for easy access
- **Automatic Updates:** By entering and saving your ACM Web Account login information, the apps will automatically update with the latest issues of TechNews published every Monday, Wednesday, and Friday

The Apps are freely available to download from the Apple iTunes Store, but users must be registered individual members of ACM with valid Web Accounts to receive regularly updated content.

<http://www.apple.com/iphone/apps-for-iphone/>

<http://www.apple.com/ipad/apps-for-ipad/>

# ACM TechNews



## Departments

5 **ICPS Editor's Letter**  
**ICPS Offers Major Research Venue**  
*By Tom Rodden*

6 **Letters To The Editor**  
**Shine the Light of Computational Complexity**

9 **In the Virtual Extension**

10 **BLOG@CACM**  
**Matters of Design**  
Jason Hong considers how software companies could effectively incorporate first-rate design into their products.

12 **CACM Online**  
**End of Days for Communications in Print?**  
*By David Roman*

43 **Calendar**

108 **Careers**

## Last Byte

112 **Puzzled**  
**Parsing Partitions**  
*By Peter Winkler*



**About the Cover:**  
Hand gestures are the most primary and expressive form of human communication. While such gestures may seem a natural means for interacting with computers, progress has often been lost in translation. This month's cover story details the requirements of hand-gesture interfaces and the challenges of meeting the needs of various apps.

## News

13 **Chipping Away at Greenhouse Gases**  
Power-saving processor algorithms have the potential to create significant energy and cost savings.  
*By Gregory Goth*

16 **Information Theory After Shannon**  
Purdue University's Science of Information Center seeks new principles to answer the question 'What is information?'  
*By Neil Savage*

19 **Maurice Wilkes: The Last Pioneer**  
Computer science has lost not only a great scientist, but an important link to the electronic computing revolution that took place in the 1940s.  
*By Leah Hoffmann*

20 **Following the Crowd**  
Crowdsourcing is based on a simple but powerful concept: Virtually anyone has the potential to plug in valuable information.  
*By Samuel Greengard*

23 **ACM Launches New Digital Library**  
More than 50 years of computing literature is augmented, streamlined, and joined to powerful new tools for retrieval and analysis.  
*By Gary Anthes*

25 **ACM Fellows Honored**  
Forty-one men and women are inducted as 2010 ACM Fellows.

## Viewpoints

26 **Privacy and Security**  
**Against Cyberterrorism**  
Why cyber-based terrorist attacks are unlikely to occur.  
*By Maura Conway*

29 **Economic and Business Dimensions**  
**Household Demand for Broadband Internet Service**  
How much are consumers willing to pay for broadband service?  
*By Gregory Rosston, Scott Savage, and Donald Waldman*

32 **Inside Risks**  
**The Growing Harm of Not Teaching Malware**  
Revisiting the need to educate professionals to defend against malware in its various guises.  
*By George Ledín, Jr.*

35 **Code Vicious**  
**Forest for the Trees**  
Keeping your source trees in order.  
*By George V. Neville-Neil*

37 **Education**  
**From Science to Engineering**  
Exploring the dual nature of computing education research.  
*By Mark Guzdial*

41 **Viewpoint**  
**Technology, Conferences, and Community**  
Considering the impact and implications of changes in scholarly communication.  
*By Jonathan Grudin*



**The Need for a New Graduation Rite of Passage**  
A proposal for a new organization recognizing students graduating in the computer sciences as professionals with ethical responsibilities in service to society.  
*By John K. Estell and Ken Christensen*



## Practice

44 **Finding Usability Bugs with Automated Tests**

Automated usability tests can be valuable companions to in-person tests.

*By Julian Harty*

50 **A Plea from Sysadmins to Software Vendors: 10 Do's and Don'ts**

What can software vendors do to make the lives of system administrators a little easier?

*By Thomas A. Limoncelli*

52 **System Administration Soft Skills**

How can system administrators reduce stress and conflict in the workplace?

*By Christina Lear*



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

## Contributed Articles

60 **Vision-Based Hand-Gesture Applications**

Body posture and finger pointing are a natural modality for human-machine interaction, but first the system must know what it's seeing.

*By Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Yael Edan*

72 **Structured Data on the Web**

Google's WebTables and Deep Web Crawler identify and deliver this otherwise inaccessible resource directly to end users.

*By Michael J. Cafarella, Alon Halevy, and Jayant Madhavan*

10 **Scientific Problems in Virtual Reality**

Modeling real-world objects requires knowing the physical forces shaping the world and the perception shaping human behavior.

*By Qinping Zhao*

## Review Articles

80 **Still Building the Memex**

What would it take for a true personal knowledge base to generate the benefits envisioned by Vannevar Bush?

*By Stephen Davies*

VE **Are the OECD Guidelines at 30 Showing Their Age?**

Created to protect the transborder flow of personal data, the OECD Guidelines are in need of some updating.

*By David Wright, Paul De Hert, and Serge Gutwirth*

## Research Highlights

90 **Technical Perspective****Markov Meets Bayes**

*By Fernando Pereira*

91 **The Sequence Memoizer**

*By Frank Wood, Jan Gasthaus, Cédric Archambeau, Lancelot James, and Yee Whye Teh*

99 **Technical Perspective****DRAM Errors in the Wild**

*By Norman P. Jouppi*

100 **DRAM Errors in the Wild:****A Large-Scale Field Study**

*By Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber*



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 Membership

Lillian Israel

#### Director, Office of SIG Services

Donna Cappel

#### Director, Office of Publications

Bernard Rous

#### Director, Office of Group Publishing

Scott Delman

#### ACM COUNCIL

##### President

Alain Chesnais

##### Vice-President

Barbara G. Ryder

##### Secretary/Treasurer

Alexander L. Wolf

##### Past President

Wendy Hall

##### Chair, SGB Board

Vicki Hanson

##### Co-Chairs, Publications Board

Ronald Boisvert and Jack Davidson

##### Members-at-Large

Vinton G. Cerf;

Carlo Ghezzi;

Anthony Joseph;

Mathai Joseph;

Kelly Lyons;

Mary Lou Soffa;

Salil Vadhhan

##### SGB Council Representatives

Joseph A. Konstan;

G. Scott Owens;

Douglas Terry

#### PUBLICATIONS BOARD

##### Co-Chairs

Ronald F. Boisvert; Jack Davidson

##### Board Members

Nikil Dutt; Carol Hutchins;

Joseph A. Konstan; Ee-Peng Lim;

Catherine McGeoch; M. Tamer Ozsu;

Holly Rushmeier; 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

2 Penn Plaza, Suite 701

New York, NY 10121-0701 USA

T (800) 401-1799; F (541) 687-1840

#### Association for Computing Machinery (ACM)

2 Penn Plaza, Suite 701

New York, NY 10121-0701 USA

T (212) 869-7440; F (212) 869-0481

# 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](mailto:publisher@cacm.acm.org)

#### Executive Editor

Diane Crawford

#### Managing Editor

Thomas E. Lambert

#### Senior Editor

Andrew Rosenbloom

#### Senior Editor/News

Jack Rosenberger

#### Web Editor

David Roman

#### Editorial Assistant

Zarina Strakhan

#### Rights and Permissions

Deborah Cotton

#### Art Director

Andrij Borys

#### Associate Art Director

Alicia Kubista

#### 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 Eng

#### 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](mailto:permissions@cacm.acm.org)

### Calendar items

[calendar@cacm.acm.org](mailto:calendar@cacm.acm.org)

### Change of address

[acmcoa@cacm.acm.org](mailto:acmcoa@cacm.acm.org)

### Letters to the Editor

[letters@cacm.acm.org](mailto:letters@cacm.acm.org)

## WEB SITE

<http://cacm.acm.org>

## AUTHOR GUIDELINES

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

## ADVERTISING

### ACM ADVERTISING DEPARTMENT

2 Penn Plaza, Suite 701, New York, NY

10121-0701

T (212) 869-7440

F (212) 869-0481

#### Director of Media Sales

Jennifer Ruzicka

[jen.ruzicka@hq.acm.org](mailto:jen.ruzicka@hq.acm.org)

Media Kit [acmm mediasales@acm.org](mailto:acmm mediasales@acm.org)

## EDITORIAL BOARD

### EDITOR-IN-CHIEF

Moshe Y. Vardi

[eic@cacm.acm.org](mailto:eic@cacm.acm.org)

## NEWS

### Co-chairs

Marc Najork and Prabhakar Raghavan

### Board Members

Brian Bershad; Hsiao-Wuen Hon;

Mei Kobayashi; Rajeev Rastogi;

Jeannette Wing

## VIEWPOINTS

### Co-chairs

Susanne E. Hambrusch; John Leslie King;

J Strother Moore

### Board Members

P. Anandan; William Aspray;

Stefan Bechtold; Judith Bishop;

Stuart I. Feldman; Peter Freeman;

Seymour Goodman; Shane Greenstein;

Mark Guzdial; Richard Heeks;

Rachelle Hollander; Richard Ladner;

Susan Landaur; Carlos Jose Pereira de Lucena;

Beng Chin Ooi; Loren Terveen



## PRACTICE

### Chair

Stephen Bourne

### Board Members

Eric Allman; Charles Beeler; David J. Brown;

Bryan Cantrill; Terry Coatta; Mark Compton;

Stuart Feldman; Benjamin Fried;

Pat Hanrahan; Marshall Kirk McKusick;

George Neville-Neil; Theo Schlossnagle;

Jim Waldo

The Practice section of the CACM

Editorial Board also serves as

the Editorial Board of *queue*.

## CONTRIBUTED ARTICLES

### Co-chairs

Al Aho and Georg Gottlob

### Board Members

Yannis Bakos; Elisa Bertino; Gilles

Brassard; Alan Bundy; Peter Buneman;

Andrew Chien; Peter Druschel;

Anja Feldmann; Blake Ives; James Larus;

Igor Markov; Gail C. Murphy; Shree Nayar;

Lionel M. Ni; Sriram Rajamani;

Jennifer Rexford; Marie-Christine Rousset;

Avi Rubin; Fred B. Schneider;

Abigail Sellen; Ron Shamir; Marc Snir;

Larry Snyder; Manuela Veloso;

Michael Vitale; Wolfgang Wahlster;

Andy Chi-Chih Yao; Willy Zwaenepoel

## RESEARCH HIGHLIGHTS

### Co-chairs

David A. Patterson and Stuart J. Russell

### Board Members

Martin Abadi; Stuart K. Card; Jon Crowcroft;

Deborah Estrin; Shafi Goldwasser;

Monika Henzinger; Maurice Herlihy;

Dan Huttenlocher; Norm Jouppi;

Andrew B. Kahng; Gregory Morrisett;

Michael Reiter; Mendel Rosenblum;

Ronitt Rubinfeld; David Salesin;

Lawrence K. Saul; Guy Steele, Jr.;

Madhu Sudan; Gerhard Weikum;

Alexander L. Wolf; Margaret H. Wright

## WEB

### Co-chairs

James Landay and Greg Linden

### Board Members

Gene Golovchinsky; Marti Hearst;

Jason I. Hong; Jeff Johnson; Wendy E. MacKay



## ACM Copyright Notice

Copyright © 2011 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](mailto: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](http://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) 626-0654.

## Single Copies

Single copies of *Communications of the ACM* are available for purchase. Please contact [acmhlp@acm.org](mailto:acmhlp@acm.org).

## COMMUNICATIONS OF THE ACM

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

## POSTMASTER

Please send address changes to

*Communications of the ACM*

2 Penn Plaza, Suite 701

New York, NY 10121-0701 USA



Association for  
Computing Machinery



Printed in the U.S.A.



Over the coming years we hope the International Conference Proceedings Series will emerge as a venue that is trusted by researchers to publish novel and adventurous work of the highest possible quality.



DOI:10.1145/1897816.1897817

Tom Rodden

## ICPS Offers Major Research Venue

The ACM International Conference Proceedings Series (ICPS) has recently being relaunched as a publication venue for research activities.

The ICPS complements publications from existing ACM-supported venues by allowing the results of high-quality research meetings not formally sponsored by ACM or its SIGs to be distributed via the ACM Digital Library. Entries will carry a distinctive logo indicating they are published as an ICPS member.

First initiated in 2002, the ICPS has successfully provided conference organizers a means of electronically publishing proceedings that ensures high visibility and wide distribution. Over 400 volumes have been published in the last eight years.

The goal is to establish the series as a major research venue that is trusted by researchers to handle high-quality academic papers arising from workshops, research meetings, and smaller conferences. We wish the series to provide a publication outlet for a broad range of research meetings that often find it difficult to make high-quality research available to a broad population. These might include:


**Research workshops** and smaller conferences from emerging domains seeking to establish new research areas in computing. The ICPS offers these communities the opportunity to reach a broad audience through the ACM Digital Library.

**Specialist meetings** focusing on the issues surrounding a detailed research domain or topic that tends to have smaller numbers of attendees but high potential impact. The ICPS offers these meetings an archival venue and broad

distribution across the worldwide computing community.

**Interdisciplinary meetings**, where researchers work across disciplinary boundaries, is of growing importance in domains such as cyber-infrastructure and e-research. Current topics include synthetic biology, privacy, and ethics. The ICPS offers an early publication venue that spans the various constituent disciplines involved.

**Agenda-setting workshops**, where leading researchers gather to reflect on key critical challenges and debate issues surrounding computing and its use, are often important in shaping our discipline. Publication through the series allows this work to be shared by a broad community.

The success of the ICPS is first and foremost dependant on the quality of the work that is published. Over the coming years we hope to emerge as a venue that is trusted by researchers to publish novel and adventurous work of the highest possible quality. If you are involved in a meeting that will produce high-quality research you feel would benefit from publication through the ACM Digital Library, I strongly encourage you to provide details of your meeting for consideration in the ICPS. Submission details can be found at [http://www.acm.org/publications/icp\\_series](http://www.acm.org/publications/icp_series). 

**Tom Rodden**, a professor at the University of Nottingham, England, and joint director of the Mixed Reality Laboratory, is the editor-in-chief of ICPS.

© 2011 ACM 0001-0782/11/0200 \$10.00

# Shine the Light of Computational Complexity

**R**EGARDING MOSHE Y. Vardi's view of computational complexity in his Editor's Letter "On **P**, **NP**, and Computational Complexity" (Nov. 2010), I'd like to add that the goal of computational complexity is to explore the potential and limitation of efficient computation. While **P** vs. **NP** is a central pivot in that direction, computational complexity is not reduced to it exclusively; nevertheless, my comments are limited to **P** vs. **NP**.

**P** vs. **NP** refers to the relative difficulty of finding solutions to computational problems in comparison to checking the correctness of solutions to these problems. Common sense suggests that finding solutions is more difficult than checking their correctness, and it is widely believed that **P** is different from **NP**. Vardi advocated the study of **P** vs. **NP**, saying that knowing is different from believing and warning that beliefs are sometimes wrong.

The ability to prove a central result is connected to obtaining a much-deeper understanding of the main issues at the core of a field. Thus, a proof that **P** is different from **NP** is most likely to lead to a better understanding of efficient computation, and such a theoretical understanding is bound to have a significant effect on computer practice. Furthermore, even ideas developed along the way, attempting to address **P** vs. **NP**, influence computer practice; see, for example, SAT solvers.

This does not dispute the claim that there is a gap between theory and practice; theory is not supposed to replace but rather inform practice. One should not underestimate the value of good advice or good theory; neither should one overestimate it. Real-life problems are solved in practice, but good practice benefits greatly from good theory.

One should also realize that the specific formulation of the **P** vs. **NP** question (in terms of polynomial running time) is merely the simplest formulation of a more abstract question. Ditto with respect to the focus on worst-case complexity. In either case, the current

formulation should be viewed as a first approximation, and it makes sense to study and understand it before moving forward.

Unfortunately, we lack good theoretical answers to most natural questions regarding efficient computation—not because we ask the wrong questions but because answering is so difficult.

Despite our limited understanding compared to the questions, we have made significant progress in terms of what we knew several decades ago. Moreover, this theoretical progress has influenced computer practice (such as in cryptography). It makes sense that most of computer science deals with actually doing the best it can at the moment—develop the best computational tools, given the current understanding of efficient computation—rather than wait for sufficient progress in some ambitious but distant project. It also makes sense that theoretical computer science (TCS) helps meet today's practical challenges. But it is crucial for a particular aspect of TCS—complexity theory—to devote itself to understanding the possibilities and limitations of efficient computation.

**Oded Goldreich**, Rehovot, Israel

## Hold Manufacturers Liable

In his Viewpoint "Why Isn't Cyberspace More Secure?" (Nov. 2010), Joel F. Brenner erroneously dismissed the value of making software manufacturers liable for defects, with this misdirected statement: "Deciding what level of imperfection is acceptable is not a task you want your Congressional representative to perform." But Congress doesn't generally make such decisions for non-software goods. The general concept of "merchantability and fitness for a given application" applies to all other goods sold and likewise should be applied to software; the courts are available to resolve any dispute over whether an acceptable level of fitness has indeed been met.

In no other commercial realm do we tolerate the incredible level of un-

reliability and insecurity characteristic of today's consumer software; and while better engineering is more challenging and the software industry could experience dislocations as its developers learn to follow basic good engineering practices in every product they bring to market, that lesson does not excuse the harm done to consumers from not employing basic good engineering practices.

**L. Peter Deutsch**, Palo Alto, CA

## Author's Response:

*The challenge is in writing standards that would improve security without destroying creativity. "Basic good engineering" is not a standard. A "merchantability and fitness" standard works for, say, lawnmowers, where everyone knows what a defect looks like. It doesn't work for software because defining "defect" is so difficult, and the stuff being written is flying off the shelves; that is, it's merchantable. It's also sold pursuant to enforceable contracts. So while courts are indeed available to resolve disputes, they usually decide them in favor of the manufacturer. Deutsch and I both want to see more secure and reliable software, but, like it or not, progress in that direction won't be coming from Congress.*

**Joel F. Brenner**, Washington, D.C.

## Code Syntax Is Understanding

In his article "Sir, Please Step Away from the ASR-33!" (Nov. 2010), Pohl-Henning Kamp was deliberately provocative regarding programming language syntax, but his arguments were confused and off the mark. To call attention to new directions available to language designers, he focused on syntax, and surprisingly, complained about the "straightjacket" imposed by ASCII but also made a good point regarding the importance of "expressing our intentions clearly." Still, he distorted the role of "syntax," which involves critical goals beside "expressivity," including machine interpretation, amenability to formal analysis, efficiency (in many dimensions), and



persistence over time. A computer language also concerns communicating with the computer.

Kamp seemed to conflate the formal syntax of a language with a variety of presentation and communication issues in programming environments, rather than with the language itself. His examples even demonstrated my point; no matter what the formal syntax, contemporary tools can overlay useful semantics, making it much easier for humans to express their ideas. Why in the world would we want to enshrine the vagaries of human perception and cognition in the syntax of a computer language?

I also have reservations about many of Kamp's suggested improvements. He clearly privileges "expression" over "communication," and his reference to using color and multi-column layouts is highly problematic. These concepts make assumptions about the technical capabilities available to users that are as likely to change as the perceived technical constraints that led to the syntax of C. Despite his intention to be provocative, Kamp was also quite conservative in his technological assumptions, staying with two dimensions, eschewing sound, ignoring handheld technology, and generally expecting WIMP interfaces on commodity PCs.

I find the biggest problem with programming languages involves understanding, not expression. I'm probably typical in that I've read far more code than I've written, most of it by strangers in contexts I can only guess. Many of my toughest challenges involve unraveling the thoughts of these other programmers based on limited evidence in the code. For the reader, adding color coding, complex nonlinear texts, and thousands of glyphs means only more cognitive load. It's difficult enough to "execute C/Java/etc in my head"; mapping complex, multi-colored hypertext to formal logic only makes it more difficult.

Kamp made a great case for why humans should never see programming languages, just like they should never see XML or RDF. They should express themselves through environments that promote communication by humans, with the added ability for the machine to generate correct code (in multiple languages) as needed. While such

communication could be implemented through a programming language, the language itself would need features not generally found in conventional languages, including semantics (not easy to define or express) that model displays and layouts.

All in all, I thank Kamp for his comments, but ASCII isn't the heart of the matter. The real heart is the design of programming environments.

**Robert E. McGrath**, Urbana, IL

### Author's Response:

*As desirable as McGrath's vision might be, in a trade where acrophobia is the norm, giant leaps may be ineffective. So while others work on the far future, I am happy to start the stairs that will eventually get us there.*

**Poul-Henning Kamp**,

Slagelse, Denmark

### Credit Due for Tandem's Hardware and OS

Joe Armstrong explained in his article "Erlang" (Sept. 2010) how a programming language originally developed for "building high-performance telecom switches" is today used for a range of high-availability, scalable applications, but I would like to clarify two parts of that explanation: The first was saying, "This technique was used by Jim Gray<sup>2</sup> in the design of the fault-tolerant Tandem computer." Gray made major contributions at Tandem, but by late 1980, when he joined the company, its fundamental hardware and operating system fault-tolerance techniques were already established. Assigning credit accurately for the various aspects of a large and complex system design (such as Tandem's NonStop systems) may be tricky, but Bartlett<sup>1</sup> and Katzman<sup>4</sup> were key contributors to the hardware and the operating system, respectively. Bartlett's paper acknowledged Dennis McEvoy, Dave Hinders, Jerry Held, and Robert Shaw as contributing to design, implementation, and testing. Finally, the co-inventors listed on the first patent granted on the Tandem system, 4,228,496, October 14, 1980, filed September 7, 1976, were: Katzman, James A. (San Jose, CA); Bartlett, Joel F. (Palo Alto, CA); Bixler, Richard M. (Sunnyvale, CA); Davidow, William H. (Ather-ton, CA); Despotakis, John A. (Pleasanton, CA); Graziano, Peter J. (Los Altos, CA); Green, Michael D. (Los Altos, CA); Greig, David A. (Cupertino, CA); Hayashi, Steven J. (Cupertino, CA); Mackie, David R. (Ben Lomond, CA); McEvoy, Dennis L. (Scotts Valley, CA); Treybig, James G. (Sunnyvale, CA); and Wierenga, Steven W. (Sunnyvale, CA).

Armstrong also said, "Adding transactions is easy," then sketched an implementation. While the implementation may be useful in the telecom world, it does not handle the durability provided by database transactions; see, for example, Gray and Reuter.<sup>3</sup>

**Paul McJones**, Mountain View, CA

### References

1. Bartlett, J.F. A 'nonstop' operating system. In *Proceedings of the Hawaii International Conference on System Sciences* (1978), 103–119.
2. Gray, J. *Why Do Computers Stop and What Can Be Done About It?* Technical Report 85.7. Tandem Computers, Inc., 1985.
3. Gray, J. and Reuter, A. *Transaction Processing: Concepts and Techniques*. Morgan Kaufman, 1993.
4. Katzman, J.A. System architecture for NonStop computing. *CompCon* (1977), 77–80.

### Author's Response:

*McJones is correct in saying the transactions described in my article are nondurable. Database transactions with ACID—atomicity, consistency, isolation, durability—properties are provided by the mnesia database included in the Erlang distribution. The purpose of that section was not to cover kinds of transactions but to show the ease a particular type of nondurable transaction involving an in-memory reversion to an earlier state could on failure be accomplished through single-assignment variables.*

**Joe Armstrong**, Stockholm

**Communications** welcomes your opinion. To submit a Letter to the Editor, please limit your comments to 500 words or less and send to [letters@cacm.acm.org](mailto:letters@cacm.acm.org).

© 2011 ACM 0001-0782/11/0200 \$10.00

## Coming Next Month in COMMUNICATIONS

*Plug and Play Macroscopes*

*Seven Principles for  
Understanding Scam Victims*

*Data Structure in  
the Multicore Age*

**Also, the latest news on memristors,  
TeraGrid, PCAST, and interpreting  
Twitter's data stream.**



Association for  
Computing Machinery

Advancing Computing as a Science & Profession

# membership application & digital library order form

Priority Code: AD10

## 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/serving/ethics.html>

## 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](http://www.acm.org)

Professional membership dues include \$40 toward a subscription to *Communications of the ACM*. 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](mailto: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:10.1145/1897816.1897821

# In the Virtual Extension

*To ensure the timely publication of articles, Communications created the Virtual Extension (VE) to expand the page limitations of the print edition by bringing readers the same high-quality articles in an online-only format. VE articles undergo the same rigorous review process as those in the print edition and are accepted for publication on merit. The following synopses are from articles now available in their entirety to ACM members via the Digital Library.*

## viewpoint

DOI: 10.1145/1897816.1897846

### The Need for a New Graduation Rite of Passage

John K. Estell and Ken Christensen

The use of computers is pervasive throughout our society. Given the ever-increasing reliance placed upon software, graduates from computing-related degree programs must be more aware than ever of their responsibilities toward ensuring that society is well served through their creative works. The authors propose a new organization establishing a rite-of-passage ceremony for students graduating in the computing sciences that is similar in nature and scope to the Ring Ceremony employed by the Order of the Engineer for students graduating from engineering programs. This new organization is solely intended to promote and recognize the ethical and moral behavior in graduates of computing-related degree programs as they transition to careers of service to society.

The proposal is not a call for accreditation, licensure, or certification at any level. It is also not a call for the formation of a new professional society such as ACM. The proposed new organization would not be a membership organization; there would be no meetings, no conferences, and no annual dues. Its sole purpose would be to facilitate and promote a rite-of-passage ceremony where students take a pledge to affirm and uphold the ethical tenets of the profession they are about to enter.

Two institutions—Ohio Northern University and the University of South Florida—have already experimented with this concept. The authors seek to start a larger conversation on this concept by soliciting input from the community on what they believe is a significant need for an organization that can benefit both graduates and the computing profession.

## contributed article

DOI: 10.1145/1897816.1897847

### 10 Scientific Problems in Virtual Reality

Qinping Zhao

Virtual reality was one of the 14 Grand Challenges identified as awaiting engineering solutions announced in 2008 by the U.S. National Academy of Engineering. In this article, the authors explore 10 related open VR challenges, with hoped-for potential breakthroughs promising to advance VR techniques and applications.

VR today is being applied in multiple contexts, including training, exercise, engineering design, and entertainment, while also serving as a research tool in such fields as neuroscience and psychology, as explored in Michael Heim's pioneering 1993 book *Metaphysics of Virtual Reality*. More recently, scholars have described the Internet itself as representing a virtual world modeling its real-world counterpart. The relationship between VR and its application fields is, in terms of expression and validation, like the relationship between mathematics and physics, while VR is attracting attention from a growing number of governments and science/engineering communities. Along with the NAE Committee on Engineering 14 Grand Challenges, the Chinese government's 2006 report *Development Plan Outline for Medium- and Long-Term Science and Technology Development (2006–2020)* and the Japanese government's 2007 long-term strategic report *Innovation 2025* both included VR as a priority technology worthy of development.

VR has also emerged as an important research area for many Chinese universities and research institutes. For example, Zhejiang University in Hangzhou and Tsinghua University in Beijing are known for realistic modeling and rendering; Peking University in Beijing focuses on computer vision and human-machine interaction; the Beijing Institute of Technology in Beijing emphasizes head-mounted displays; and the Institute of Computing Technology of Chinese Academy of Sciences in Beijing has made significant progress in crowd simulation.

## review article

DOI: 10.1145/1897816.1897848

### Are the OECD Guidelines at 30 Showing Their Age?

David Wright, Paul De Hert, and Serge Gutwirth

Three decades have passed since the Organisation for Economic Co-operation and Development (OECD) promulgated Guidelines on the Transborder Flows of Personal Data, and still the issue of transborder flows of data continues to plague policymakers, industry, and individuals who have no idea what happens to their data once it is transmitted beyond their national jurisdictions. This article briefly reviews what happened in the 1970s, the factors that led to production of the guidelines, and some of their key points. The authors highlight the success of the guidelines, but also the shortcomings, and what is happening now to bridge the gap. They ask the defining question: "Is an international binding convention or standard still needed?"

In the 1970s, the decade before the OECD Guidelines were promulgated, some countries had already begun to enact privacy laws applicable to the public and private sectors, including Germany, France, Sweden, and the U.S. In the seven-year stint between 1973 and 1980, one-third of the OECD's 30 member countries enacted legislation intended to protect individuals against abuse of data related to them and to give individuals the right of access to data with a view to checking their accuracy and appropriateness. Some countries were enacting statutes that dealt exclusively with computers and computer-supported activities. Other countries preferred a more general approach irrespective of the particular data processing technology involved. The OECD became concerned that these disparities in legislation might "create obstacles to the free flow of information between countries."

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:10.1145/1897816.1897820

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

## Matters of Design

*Jason Hong considers how software companies could effectively incorporate first-rate design into their products.*



**Jason Hong**  
"Why is Great Design so Hard?"

<http://cacm.acm.org/blogs/blog-cacm/96476>

I want to take a slight detour from usable privacy and security to discuss issues of design. I was recently at the Microsoft Faculty Summit, an annual event where Microsoft discusses some of the big issues and directions they are headed.

In one of the talks, a designer at Microsoft mentioned two data points I've informally heard before but had never confirmed. First, the ratio of developers to user interface designers at Microsoft is 50:1. Second, this ratio is better than any other company out there.

As someone who teaches human-computer interaction, I wanted to push on this point, so I asked the hard question: "On a relative and absolute scale, Microsoft has more designers than Apple or Google. However, I think most people would argue that Apple and Google have much better and easier-to-use products. Is it an organizational issue, a process issue, a skills issue? What do I tell the students I teach?"

I want to make clear that my question wasn't meant to be negative of Microsoft specifically. It actually hits on a per-

vasive issue facing the software industry today: How do we effectively incorporate great design into products? Because, to be blunt, a lot of user interfaces today are just plain bad. Painfully bad.

Companies like Nintendo, Apple, Google, and Amazon have proven that good design matters. Why haven't other companies followed suit? Or perhaps the real question is, Why haven't other companies *been able* to follow suit?

I discussed this issue with a lot of people at the faculty summit and in Silicon Valley. I used Microsoft as my example, but I think the same problems apply to pretty much every software company.

People offered five explanations. The first is the "geek culture" where engineers rule. At most software companies, there has never been a strong mandate at the top for great design. Instead, engineers tend to have wide latitude for features and implementation.

Second, there is often no one in charge of the overall interaction and user experience. As such, there tends to be no consistency and unified flow for how a system looks and feels.

Third, designers tend to be brought in the process too late, to do superficial work on a system that has already

been built. One person basically said it was "lipstick on a pig." Design is more than just the surface layer of the visual interface. It's also about understanding how people work and play, how things fit into their lives, how people see and understand things, what problems they have, and how to craft a product that fits these constraints.

Fourth, designers have no real power in most organizations. They are typically brought in as consultants, but cannot force (sometimes necessary) changes to happen.

Fifth, a lot of people in technical professions just don't "get" design. They think it's just putting on a pretty interface or worse, the equivalent of putting pink flamingos on one's lawn. A lot of this is due to education in computer science programs today, which focuses heavily on algorithms and engineering, but leaves little room for behavioral science, social science, the humanities, and any form of design, whether it be industrial design, graphic design, or interaction design.

Overall, this issue of design is a fundamental problem that industry doesn't quite know how to get its head around. Just hiring an interaction designer isn't enough. Improving the ratio of developers to designers isn't enough. With the advent of trends like "design thinking," the increasing number of human-computer interaction degree programs, and visibly great products, people are starting to realize that design really matters, but just don't know how to go about making it a fundamental way for organi-



zations to make software rather than just “lipstick on a pig.” The problem becomes even harder when the system combines hardware, software, and services, which describes pretty much every system we will have in the future.

I want to make clear that we, as researchers and educators, don’t know the answer either. But, in my next blog entry, I’ll write about my discussion with someone who used to work at Apple, with some insights about how its process works. I’ll also be visiting Google next week, so I’ll be sure to ask them these same questions.

P.S. I want to get your comments and feedback, but please don’t use the word “intuitive,” talk about “dumbing down interfaces,” or say that users are “dumb, stupid, and naïve.” These are signals that you really don’t know what you’re talking about. And, yes, not all interfaces have to be walk-up-and-use.

### Readers’ comments

*On one side we have the “geek culture” people, oblivious of what happens to the user, doing self-referential design. On the other hand, we have the designers with their MacBooks, reluctant to delve into technical issues.*

*Between them there is a void that supports the “nobody is in charge” stance. Actually, the void was created before everything happened, before the software was written and way before the designers cringed. The void is, in fact, the lack of a usable user interface (UI) definition in the requirements and software design stages. If it’s not specified then, the outcome might be anything. For example, the outcome might be a good UI every now and then, as it happens nowadays, making you say “a lot of user interfaces today are just plain bad.”*

*Functional analysts and software architects have to raise the flag, not graphics designers.*

*See, for example, some writings by Larry Constantine. He got it many years ago, and he is a geek like Alan Cooper and Jakob Nielsen. Not a designer. See <http://www.foruse.com/articles/whatusers.htm> or other articles in <http://www.foruse.com/publications/index.htm>.*

*The design activity that determines the usability of a nontrivial UI is the one that happens before writing the first line of code. Moreover, before starting to envision a disposable prototype.*

—Juan Lanus

JASON HONG

## “This issue of design is a fundamental problem that industry doesn’t quite know how to get its head around.”

*Many of these observations are spot on—even uncomfortably accurate. However, perhaps a missing observation is that these five different issues are not independent and many share a similar root cause. Much of it comes down to the leadership in the company and the resulting decision-making processes. Projects, resources, and timelines are typically dictated at a very high level, often by those who exclusively focus on a multiyear competitive business strategy. That calculus is generally of the form “To improve our performance in market X, we need to create product Y, with features Z by Q2 of next year because our competitors A, B, C are likely releasing E, F, and G.”*

*The notion of “good design” ends up becoming a product “feature” that just like any other feature has to be scoped and completed by a particular deadline and preferably not revisited. Good design is typically not at the essence of a product’s reason for existing. More often than not, projects are green-lit based solely on whether there is a pressing business need, rather than on the belief something genuinely great could or should be created. This core issue, in my opinion, sets into place a series of processes, priorities, and corporate culture that accounts for nearly all of your observations. Prioritizing good design requires a culture where by “creating a great experience” is a slightly higher priority than “maintaining a strategically competitive market share.” It’s a little bit of a “built it and they will come” mentality—which, understandably, is not a pill many executives readily swallow, especially if their compensation is connected to the stock price.*

*Occasionally, business needs and good hw/sw/ux design do happen to align. But it is fairly rare. A leadership team that can see past the battlefield of business needs with enough vision and taste to identify the seedlings of a great and profitable experience ... is rarer still.*

—Johnny Lee

*I’d change the question to “Why is great design so infrequent?” because we have solved much harder problems. It requires a leader who needs to enforce the principle of profitability—i.e., “We are going to design a product that will outsell our competitors’ products because they will want to buy it”—and the infrequency of great design is a result of the infrequency of these leaders. I suspect they are often only one person who comes to the project with a picture of what is needed.... [Y]ou don’t need 50 design engineers; rather, what’s needed is one responsible “profit” engineer—one who is willing to delay each step of development until trial users give their feedback. You can’t predict what users will want, but you can put prototypes in their hand and have them tell you what they want.*


—Neil Murphy

### Jason Hong responds

One of the goals of emerging human-computer interaction programs is to cross-train “Renaissance teams” that can combine the best of design, computer science, and behavioral science.

I take the term “Renaissance team” from Randy Pausch, who noted that given the scope of knowledge today, “Renaissance man” is a near impossibility today. Plus, people tend to listen to me more when I quote him. :-)

On the computer science side, I’ve been advocating that all computer science undergrads should be required to take at least 1) a machine learning course, and 2) a human-computer interaction course. My rationale is that these two themes are fundamental to the future of computer science. Taking a machine learning course is an easy sell in CS departments and starting to happen, but it’s less so with HCI.

I’m not sure yet what to advocate on the design and behavioral science side of things, though. 

Jason Hong is an assistant professor of computer science at Carnegie Mellon University.

© 2011 ACM 0001-0782/11/0200 \$10.00



DOI:10.1145/1897816.1897821

David Roman

## End of Days for Communications in Print?

Calls to update the peer-review publishing model to accommodate the rise in online publishing (<http://cacm.acm.org/magazines/2011/1/103232>) raise a question about the durability of print publications and of the printed magazine version of *Communications of the ACM*. How long will it last? ACM asked this very question in 2006 before starting the *Communications* revitalization project. “We talked to numerous ACM members about their expectations for the flagship publication. We explicitly asked whether they would like to continue to see *Communications* as a print publication,” says *Communications* Editor-in-Chief Moshe Y. Vardi. “The vast majority expressed a strong desire to continue to see *Communications* in print.”

Members want multiple formats, including print. Institutional libraries and individual members like print for archival reasons and for “ease of use,” says Scott Delman, ACM’s Director of Group Publishing. “Until this demand completely or significantly disappears, publishers will likely still continue to print paper issues.”

While commercial publishers struggle to update their business models, ACM and other scholarly publishers have already made a capable transition from print to digital media.

Half of commercial publishers generated less than 10% of their 2009 revenue through e-media, according to *Folio* magazine (<http://www.foliomag.com/2009/e-media-reality-check>). In contrast, “most mid-sized to large [scholarly] publishers currently experience something like 85% to 90% of their revenues from online business,” Delman says. ACM has been a predominantly digital publisher for a number of years, he says.

Even so, the forecast for print is fuzzy. “I am convinced that we will not have a print issue of *Communications* in 25 years, but it is hard to predict when it will go away,” says EiC Vardi. Change will start, says BLOG@CACM blogger Ed Chi of the Palo Alto Research Center, with scholarly journals going paperless and digitally publishing online first. “This will happen gradually, and only if there are ways to manage the publication process, and the archival aspect of publications.”

Newer, dynamic digital formats will also marginalize print media. “More and more, there is experimentation with delivery in mobile-friendly formats, but it is fair to say that these formats will not likely supplant print or Web-based formats for some time,” Delman says. With its development of a mobile Web site and mobile apps, *Communications* will continue to follow the technology curve of e-readers and tablet computers, Vardi says. “At some point in the future, the user experience of reading on mobile devices will be so good that print will start fading away as an anachronism.”

Digital media will free *Communications* from the constraints of print, Vardi says. “As we gradually shift the focus from print to digital publishing, the articles we publish will become less textual and less linear, and will take advantage of the flexibility and richness of the digital medium. I can imagine in the not-too-distant future an article on computational whiskey making, where the reader can actually smell the whiskey!”

Salut!

## ACM Member News

### AN ADVOCATE FOR WOMEN IN INDIA



Gayatri Buragohain, India’s ACM-W Ambassador and a member of the ACM India Council,

is an outspoken advocate for women in India who face an opportunity deficit due to a legacy of gender bias. Buragohain founded the nonprofit Feminist Approach to Technology (FAT) in 2007, and was the recipient of the Anita Borg Institute’s Change Agent Award for 2010 for her advocacy and mentoring efforts.

“Sadly, there is not much awareness about women’s role in technology in India,” Buragohain says. “My motivation for starting FAT was to create awareness and encourage discussions and actions to bridge the gap.”

Based in New Delhi, Buragohain not only focuses on providing local women with technical training and vocational guidance through FAT, but also leads a company called Joint Leap Technologies, a technology consulting and development firm that works closely with the FAT training center and serves as its primary financial donor.

Buragohain says her advocacy efforts could be significantly aided by more studies that are specific to India and address the concerns she is raising. “The concept of women’s rights is not very popular in India,” Buragohain says. “While there are a lot of efforts to eradicate violence and discrimination against women, most women are unaware of these efforts.”

Looking ahead, Buragohain says she is optimistic that education and work opportunities will improve through policy changes designed to counteract long-standing stereotypes about women not being adept at scientific or analytical work. “When we talk of a world of equality and gender justice, equal participation of women in decision-making is a must,” she says.

—Kirk L. Kroeker



## Chipping Away at Greenhouse Gases

*Power-saving processor algorithms have the potential to create significant energy and cost savings.*

THE INFORMATION TECHNOLOGY industry is in the vanguard of “going green.” Projects such as a \$100 million hydro-powered high-performance data center planned for Holyoke, MA, and green corporate entities such as Google Energy, the search giant’s new electrical power subsidiary, are high-profile examples of IT’s big moves into reducing the greenhouse gases caused by computers.

However, the true benefits of such projects are likely to be limited; most users in areas supplied by coal, oil, or natural gas-fired power plants would likely find it difficult to change to a fully sustainable supply source.

These market dynamics have not been lost on government research directors. Agencies such as the U.S. National Science Foundation (NSF) have begun encouraging just the sort of research into component-level power management that might bring significant energy savings and reduced climatic impact to end users everywhere without sacrificing computational performance.

In fact, the NSF has held two workshops in the newly emphasized science of power management, one in



**An intelligent power-management application, Granola uses predictive algorithms to dynamically manage frequency and voltage scaling in the chips of consumer PCs.**

2009 and one in 2010. Krishna Kant, a program director in the Computer Systems Research (CSR) cluster at the NSF, says the power management project is part of the NSF’s larger Science, Engineering, and Education for Sustainability (SEES) investment area.

“There are some fundamental ques-

tions that haven’t been answered, and NSF funding might help answer them,” Kant says. “These have been lingering for quite some time. For instance, when you look at the question of how much energy or power you really need to get some computation done, there has been some research, but it tends

to be at a very, very abstract level to the extent it's not very useful."

### Thermal Head Start

However abstract the state of some of the research into power management might be, basic computer science has given the IT industry a head start over other industries in addressing power issues. Whereas an auto manufacturer could continue to make gas-guzzling vehicles as long as a market supported such a strategy, two factors in particular have focused microprocessor designers' efforts on the imperatives of power efficiency.

One of the factors is the thermal limitations of microprocessors as each succeeding generation grew doubly powerful per unit size. The other is the proliferation of laptops and mobile computing devices, which demand advanced power management features to extend battery life. Kirk Cameron, associate professor of computer science at Virginia Polytechnic Institute, says this shift in product emphasis has given engineers working on power management theories more tools with which to work on the central processing unit (CPU); these chips are also installed on desktop machines and servers as chip manufacturers design one family for numerous platforms, based on overall market demand. Examples of these tools include application programming interfaces such as Intel's SpeedStep and AMD's PowerNow, which allow third-party software to dynamically raise or lower the frequency of cycles and the voltage surging through the

processor, depending on the computational load at any given time.

However, the default power management schemes supported by current operating systems, which allow users to specify either a high-performance or battery-maximizing mode on laptops, for instance, have numerous handicaps, including their static nature. The fact they need to be manually configured hampers their popularity.

Some power-management products, incubated by university researchers, are already available to dynamically manage power within a computer's CPU. Cameron is also the CEO of Miserware, a startup funded in part by an NSF Small Business Innovation Research Grant. Miserware produces intelligent power-management applications—called Granola for consumer PCs and Miserware ES for servers—that use predictive algorithms to dynamically manage frequency and voltage scaling. Company benchmarks claim that users can reduce power usage by 2%–18%, depending on the application in use; best savings are generated by scaling down power during low-intensity activities.

Granola was launched on Earth Day last year, and has 100,000 downloads. Cameron says the dynamic voltage and frequency scaling (DVFS) technology is very stable, available on most systems, and "kind of the low-hanging fruit" in power management.

Susanne Albers, professor of computer science at Humboldt University of Berlin, believes speed scaling will be a standard approach to power manage-

ment for some time. "I am confident that dynamic speed scaling is an approach with a long-term perspective," she says. "In standard office environments the technique is maybe not so important. However, data and computing centers, having high energy consumption, can greatly benefit from it."

### Multicore Architectures

Ironically, although the DVFS technology is currently the most ubiquitous power management solution for processors, Cameron and other researchers say new fundamentals of computing architecture will mandate wholly different solutions sooner rather than later.

The onset of mass production of multicore processors, for example, is mandating that researchers begin practically anew in exploring speed scaling approaches.

"Generally speaking, there exists a good understanding of speed scaling in single processor systems, but there are still many challenging open questions in the area of multicore architectures," Albers notes.

"The new technologies bring new algorithmic issues," says Kirk Pruhs, professor of computer science at the University of Pittsburgh, and an organizer of both NSF workshops. For instance, if a heterogeneous-cored processor is programmed correctly, the utility of using frequency and voltage scaling at all might be moot—applications needing lower power can be sent to a slower core.

However, Pruhs says programming these will be "much more algorithmi-

## ACM Awards News

# 2011 ACM Fellows Nominations

The ACM Fellow program was established by the ACM Council in June 1993 to recognize outstanding ACM members for technical, professional, and leadership contributions that advance the arts, sciences, and practices of information processing; promote the free interchange of ideas and information in the field; develop and maintain the integrity and competence of individuals in the field;

and advance the objectives of ACM.

Each candidate is evaluated as a whole individual and is expected to bring honor to the ACM. A candidate's accomplishments are expected to place him or her among the top 1% of ACM members. In general, two categories of accomplishments are considered: achievements related to information technology and outstanding

service to ACM or the larger computing community. A person selected as an ACM Fellow should be a role model and an inspiration to other members.

Nominations and endorsements must be submitted online no later than Sept. 1, 2011. For Fellows Guidelines, go to [http://awards.acm.org/html/fellow\\_nom\\_guide.cfm/](http://awards.acm.org/html/fellow_nom_guide.cfm/).

Nomination information organized by a principal

nominator should include excerpts from the candidate's current curriculum vitae, listing selected publications, patents, technical achievements, honors, and other awards; a description of the work of the nominee, drawing attention to the contributions which merit designation as Fellow; and supporting endorsements from five ACM members. For the list of 2010's ACM Fellows, see p. 25.

cally difficult for the operating system to manage, and the same thing happens in memories. The fact everything is changing means you have to go back and reexamine all the algorithmic issues that arise.”

In the case of power management in a parallel environment, Cameron says his research has shown that one cannot take the principles of Amdahl's Law for parallelization—which states that any parallelized program can only speed up at the percentage of a given task within that program not run serially—and get a correct assumption about power savings by simply taking into account the processors running a given application.

“In Amdahl's Law, you have one thing that changes, the number of processors,” Cameron says. “In our generalization, we ask what if you have two observable changes? You might think you could apply Amdahl's Law in two dimensions, but there are interactive effects between the two. In isolation, you could measure both of those using Amdahl's Law, but it turns out there is a third term, of the combined effects working in conjunction, and that gets missed if you apply them one at a time.”

### Doing Nothing Well

In the long term, power management may borrow from sensor networks and embedded systems, which have extensively dealt with power constraints. Both David Culler, professor of computer science at the University of California, Berkeley, and Bernard Meyer-son, vice president of innovation at IBM, cite the disproportionately large power demands of processors doing little or no work as an area where great savings may be realized.

Culler says processor design might take a lesson from network sensor design in principle. Measuring performance during active processing “talk” time is misplaced, he says. Instead, efficiency must be introduced while awaiting instruction—“talk is cheap, listening is hard.”

Culler says theories behind effectively shutting down idle processors (“doing nothing well”) essentially fall into two basic camps that “hearken back to dark ages”—the principles following Token Ring or other time

## In the long term, processor power management may borrow from sensor networks and embedded systems, which have extensively dealt with power constraints.

division multiplex technologies, or a Carrier Sense Multiple Access approach akin to Ethernet topology, in which nodes about to transmit can first “sense” whether or not a network is idle before proceeding.

He says this principle can apply to any scenario, be it a Wi-Fi network or a bus protocol on a motherboard. “Doing nothing well and being able to respond to asynchronous events anyway is the key to power proportionality, and can apply across the board,” says Culler.

### Management From a Chip

Market demand for dynamically provisioned processors is still an unknown. Albers says processor-level power management is not particularly viewed as a critical issue among European users.

“Energy and environmental issues have always received considerable attention in Europe. However, the typical person is probably more concerned about energy consumption in his household and private car than about the consumption of his PC or laptop,” Albers observes.

IBM has placed a bet on combining chip-level energy allotment with the network architectures of homes and offices. The company has introduced fabricating technology for dedicated power management chips that control power usage while they communicate wirelessly in real time with systems used to monitor smart buildings, energy grids, and transportation systems. The main function of power-management chips is to optimize power usage

and serve as bridges so electricity can flow uninterrupted among systems and electronics that require varying levels of current.

Meyerson says that, while reducing battery usage on end user devices may be sexy, “that’s not the win for society. The win for society is when there’s an area of a building and the sensors over a period of time crawl through all the data of the occupancy of all the offices, and they autonomically adjust for the fact this is Paris in August—and in Paris in August people just aren’t showing up.”

IBM estimates the new technology can cut manufacturing costs by about 20% while allowing for the integration of numerous functions, resulting in one chip where previously three or four were needed. Meyerson says the technology can work for any appropriate algorithm researchers can come up with.

“Discovery algorithms that can look ahead and be predictive instead of reactive can be incredibly important,” he says. “What we are doing is ensuring that if they come up with a solution, there’s a way to execute it in a single chip, in a very efficient, synergistic way. It is a real footrace to stay ahead of the energy demands of society and IT.” ■

### Further Reading

Albers, S.

Energy-efficient algorithms, *Communications of the ACM* 53, 5, May 2010.

Bansal, N., Kimbrel, T., and Pruhs, K.

Speed scaling to manage energy and temperature, *Journal of the ACM* 54, 1, March 2007.

Ge, R. and Cameron, K.W.

Power-aware speedup. *IEEE International Parallel and Distributed Processing Symposium*, Long Beach, CA, March 26–March 30, 2007.

Gupta, R., Irani, S., and Shukla, S.

Formal methods for dynamic power management. *Proceedings of the International Conference on Computer Aided Design*, San Jose, CA, Nov. 11–13, 2003.

Yao, F., Demers, A., and Shenker, S.

A scheduling model for reduced CPU energy. *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, Milwaukee, WI, Oct. 23–25, 1995.

Gregory Goth is an Oakville, CT-based writer who specializes in science and technology.

© 2011 ACM 0001-0782/11/0200 \$10.00



# Information Theory After Shannon

*Purdue University's Science of Information Center seeks new principles to answer the question 'What is information?'*

**I**N A SENSE, Claude Shannon invented the Internet. An electronic engineer at Bell Labs, Shannon developed information theory in "A Mathematical Theory of Communication," a landmark paper published in 1948. By quantifying the limits to which data could be compressed, stored, and transmitted, he paved the way for high-speed communications, file compression, and data transfer, the basis for the Internet, the CD and the DVD, and all that they entail. Now scientists at Purdue University, with a \$25 million, five-year grant from the U.S. National Science Foundation, have created the Science of Information Center with the goal of moving beyond Shannon. They aim to develop principles that encompass such concepts as structure, time, space, and semantics. These principles might help design better mobile networks, lead to new insights in biology and neuroscience, drive research in quantum computing, and even aid our understanding of social networks and economic behavior.

"Whether we will build a new theory or not remains to be seen," says Wojciech Szpankowski, professor of computer science at Purdue University and leader of the project, which includes some 40 researchers at nine universities. "It's definitely time, after 60 years, to revisit information theory. It's basically communication and storage today, but we need to go beyond that."

In Shannon's theory, information, which consists of bits, is that which reduces a recipient's statistical uncertainty about what a source transmitted over a communications channel. It allows engineers to define the capacity of both lossless and lossy channels and state the limits to which data can be compressed. Shannon theory ignores the meaning of a message,



Wojciech Szpankowski, project leader for the Science of Information Center at Purdue University, is revisiting Claude Shannon's information theory with a team of some 40 researchers.

focusing only on whether the 1s and 0s of binary code are being transmitted accurately. It doesn't care about the physical nature of the channel; information theory is the same for a telegraph wire or a fiber optic cable. And it assumes infinite delay, so a receiver has all the time it needs to receive and recognize a signal.

**A growing challenge for information theory is the field of quantum information and quantum computing.**

Szpankowski argues that information goes beyond those constraints. Another way to define information is that which increases understanding and that can be measured by whether it helps a recipient to accomplish a goal. At that point, semantic, temporal, and spatial factors come into play. If a person waiting for a train receives a message at 2 P.M. saying the train leaves at 1 P.M., the message contains essentially no information. In mobile networks, the value of information can change over the time it takes to transmit it because the person receiving it has moved; instructions to make a left turn are pointless, for example, if the driver has already passed the intersection. And there's no good way to measure how information evolves on the Web. "We cannot even understand how much information is transmitted on the Internet because we don't understand the temporal aspect of information," Szpankowski says.

Sergio Verdu, an electrical engineer at Princeton University and a co-principal investigator, is trying to find the fundamental limits of compression and data transmission. The math is easy if engineers focus on signal-to-noise ratio, but that turns out to not be the best measure of fidelity in audio and video. The human brain is very tolerant of visual noise in general, but less so of particular types of noise; for instance, sharp edges are much more important to identifying an image than color. It's not just a question of how many bits have to be received to deliver a comprehensible message, but which bits. "In lossy compression, the bottleneck has been to come up with measures of distortion that are relevant in the real world," says Verdu.

It's also difficult to find the limits of compression for structural information, says Szpankowski. Proteins behave differently and transmit different information depending on their shape, and scientists would like to build computer models that would help them better understand how structure contributes to cell development or the rise of diseases. But the math becomes complex, because structural elements such as edges and vertices can add several dimensions to an equation. So far, Szpankowski says, no theory exists to provide a metric for how much information is embedded in structure. There's not even a good way to quantify complexity; we can say a car is more complex than a bicycle, but how much more?

Shannon theory works well for point-to-point transmission or in systems with several transmitters and one receiver. But once there are two transmitters and two receivers, the problem of cross-talk arises, where a receiver can pick up a signal from the wrong transmitter. With the growth in mesh networks and mobile communications, and even high-frequency transmissions turning old copper wires into miniature antennas, the point-to-point solution isn't adequate. "We still don't know what are the ultimate, fundamental limits to how much information we can send robustly in the presence of noise," Verdu says.

### Quantum Information

A growing challenge for information theory is the field of quantum infor-

mation and quantum computing. Classical information has always been understood as a collection of bits, says Madhu Sudan, principal researcher at Microsoft Research New England (now on leave from Massachusetts Institute of Technology) and a co-principal investigator. Quantum information, by contrast, is a continuum; the qubits used in quantum computing can have a value of 1 and 0 and any of the infinite possible values in between simultaneously. In such a circumstance, Sudan asks, "What does it even mean to be in-

## NSF Science and Technology Centers

The U.S. National Science Foundation (NSF) selected the Purdue University's Science of Information Center as one of the Science and Technology Centers (STCs) to receive funding in 2010. The purpose of the NSF's ambitious STC program is to "support integrative partnerships that require large-scale, long-term funding to produce research and education of the highest quality." In their quest to launch the next information revolution, the Purdue scientists will collaborate with colleagues at Bryn Mawr College; Howard University; Massachusetts Institute of Technology; Princeton; Stanford; University of California, Berkeley; University of California, San Diego; and University of Illinois at Urbana-Champaign.

NSF funded four other STCs. The Center for Dark Energy Biosphere Investigations, headed by the University of Southern California, will explore sub-surface life in deep mines and aquifers and below the ocean floor and how they influence global energy cycles. The Center for the Study of Evolution in Action, based at the University of Michigan, will develop computer models to study complex biological questions that can't be studied in nature. Emergent Behaviors of Integrated Cellular Systems, led by Massachusetts Institute of Technology, will try to engineer biological machines. And the Center for Energy Efficient Electronics Science, led by the University of California, Berkeley, will try to develop technology that can eventually reduce power consumption in electronics by a millionfold.

### Technology

## Social Media Trends

The use of social networking is becoming more prevalent worldwide, with people from countries of varying economic development increasingly accessing the Internet to participate in networking sites. In addition, cell phone ownership has increased significantly in 16 countries (for which trends are available) over the last three years, from a median of 45% in 2007 to 81% in 2010.

These are among the findings of a new survey by the Pew Research Center's Global Attitudes Project. The survey, "Global Publics Embrace Social Networking," examined technology usage in 22 countries, with 24,790 people surveyed either face-to-face or by phone. Social networking is especially widespread in the U.S., Pew says, with 46% of the U.S. survey respondents saying they use social networking sites. Other top-ranking countries are Poland (43%), Britain (43%), and South Korea (40%).

Pew notes that while involvement in social networking is relatively low in less economically developed nations, this is largely due to the fact that many in those countries are unable to access the Internet, rather than having a disinterest in social networking.

"In middle- and low-income countries, when people go online they also tend to participate in social networking," says Richard Wike, associate director of the Pew Global Attitudes Project. "In places like Poland, Russia, and Brazil, the vast majority of Internet users also use social networking sites. If you look at the two sub-Saharan African nations we surveyed, Nigeria and Kenya, relatively few people use the Internet, but among those who do, social networking is very popular."

For the most part, the study shows, men and women tend to engage in social networking at about the same rates. The U.S. is the only country in which women (52%) are significantly more likely than men (41%) to use social networking.

—Bob Violino

formation? What does it mean for you to send me information?”

Many differences exist between classical and quantum information, Sudan says, notably that classical information can be copied and quantum information, by definition, cannot. Sudan is not merely interested in discovering fundamental principles; he wants to figure out which ones have practical importance. “The next level is going to focus on ‘How do I understand this information? Where did it come from? And how can I manipulate it in ways that are convenient?’ ” he says.

Researchers are hoping to apply information theory to fields beyond communications and computing. Christopher Sims, an economist at Princeton, applies Shannon’s insights about channel capacity to how consumers respond to economic information. In theory, if the central bank alters interest rates or the money supply, prices should change quickly, but in reality they don’t. Sims says that’s because people have limited physical capacity to process all the data, so they usually don’t act on the information until it crosses some threshold, such as appearing on the Yahoo! homepage. They’re striking a balance between the cost of processing information and the

**“We cannot even understand how much information is transmitted on the Internet,” says Wojciech Szpankowski, “because we don’t understand the temporal aspect of information.”**

reduction in uncertainty—the payoff in tracking small interest rate fluctuations isn’t worth the effort it takes to react to them. Sims has dubbed this strategy “rational inattention.”

Sims is hoping to combine information theory with control theory, which looks at the behavior of dynamic systems, and come up with new economic insights. Perhaps, he says, some of those insights will feed back into engi-

neering. “There’s a long way to go here, and we’ve only just begun to get information theorists interested in what we’re doing here,” Sims says. “We’re still using information theory in a relatively primitive way.”

“I’m hoping in the first five years we’ll make some advances,” Szpankowski says. “We will at least formulate the right questions.” **G**

#### Further Reading

Goldreich, O., Juba, B., and Sudan, M. A theory of goal-oriented communication, Electronic Colloquium on Computational Complexity TR09-075, Sept. 17, 2009.

Konorski, J. and Szpankowski, W. What is information? *Zeszyty Politechniki Gdanskiej*, 5, 2007

Shannon, C.E. A mathematical theory of communication, *The Bell System Technical Journal* 27, July and October, 1948.

Sims, C.A. Rational inattention: a research agenda. Deutsche Bundesbank Spring Conference, Berlin, Germany, May 27, 2005.

Verdu, S. Fifty years of Shannon theory, *IEEE Transactions on Information Theory* 44, 6, October 1998.

**Neil Savage** is a science and technology writer based in Lowell, MA.

© 2011 ACM 0001-0782/11/0200 \$10.00

## Biology

# Algorithmic Entomology

Network software engineers have long found inspiration in ant colonies, whose collective wayfinding strategies have shed light on the problems of routing data across a busy network. In recent years, ant colony optimization has emerged as a proven algorithm for network optimization, one of several swarm intelligence models based loosely on the behavior of nature’s social animals.

Now, a recent study from the University of Sydney suggests that ant colonies may possess even greater problem-solving abilities than previously thought. The study, “Optimization in a Natural System: Argentine Ants Solve the Towers of Hanoi,” published in *The Journal of Experimental*

*Biology*, demonstrates an ant colony’s ability to adapt its navigational “algorithm” in response to changing environmental conditions. The findings may help open up new research avenues for optimizing the flow of traffic across data networks.

The international team of researchers enlisted a colony of Argentine ants to solve the famous Towers of Hanoi problem in which participants must find an optimal solution for arranging disks of varying sizes onto a set of three rods.

Translating the logic of the puzzle into a maze with 32,768 possible pathways to a food source, the researchers turned the ant colony loose for one hour, then abruptly

blocked some of the pathways. In response, the ants swiftly recalibrated their methods, and soon found an optimal new route to the food supply.

The ant colonies’ ability to rapidly adapt took the researchers by surprise. “Even simple mass-recruiting ants have much more complex and labile problem-solving skills than we ever thought,” says lead author Chris Reid, a Ph.D. student at the University of Sydney.

The ants’ successful rerouting holds potentially important lessons for network software developers. Traditionally, networking algorithms have focused on optimizing solutions to rigid, pre-programmed formulas. These approaches, while

effective, also tend to constitute a brute-force response to computational challenges.

“Although inspired by nature, these computer algorithms often do not represent the real world because they are static and designed to solve a single, unchanging problem,” says Reid.

To survive in the wild, ant colonies must do more than just follow a set of rules; they must sense, respond, and adapt to the world around them. “Nature is full of unpredictability and one solution does not fit all,” says Reid, whose research suggests that ants may have more to teach us about how to thrive in a changing environment. “Are they fixed to a single solution,” he asks, “or can they adapt?”

—Alex Wright



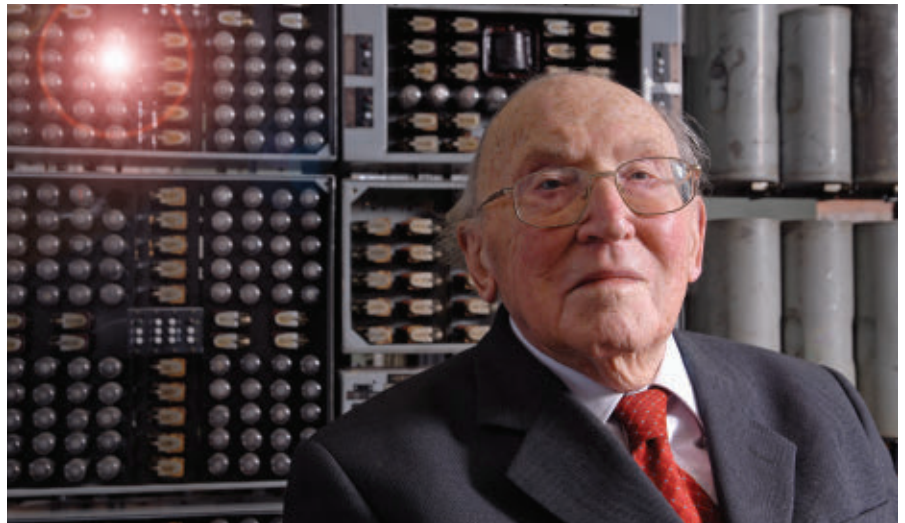
# Maurice Wilkes: The Last Pioneer

*Computer science has lost not only a great scientist, but an important link to the electronic computing revolution that took place in the 1940s.*

**S**IR MAURICE WILKES didn't like to bicker about who was first when it came to ground-breaking technical achievements. Nonetheless, history credits him with a number of important innovations, including the creation of the world's first practical stored-program computer—the earliest machine capable of running realistic programs and producing useful results—as well as the invention of microprogramming. With his death on November 29, 2010, at the age of 97, computer science lost not only a great scientist, but an important link to the electronic computing revolution that took place in the 1940s.

Wilkes was born on June 26, 1913 in Dudley, Worcestershire, England. He initially struggled in school due to recurring bouts of asthma. By his teens, however, he found his stride in the study of science and mathematics, supplementing his education with a subscription to *Wireless World* and a keen interest in amateur radio transmission. In 1931, he entered St. John's College, Cambridge to study mathematics. Subsequent graduate studies on the propagation of radio waves provided his first experiences with computing, as he seized an opportunity to work with the university's differential analyzer, a device that used wheel-and-disc mechanisms to solve differential equations.

World War II interrupted his budding career, and Wilkes left for war service in 1939, working on radar and operational research. Although Alan Turing had been a classmate at Cambridge, Wilkes was unaware of the computing developments under way at Bletchley Park during the war. After World War II, he returned to Cambridge as the head of the mathematics laboratory (later named the computer laboratory), where he was tasked with investigating new possibilities in calcu-



**Maurice Wilkes in front of the Harwell computer, now being restored at the National Museum of Computing, Bletchley Park.**


lating machinery. A chance encounter with John von Neumann's 1946 draft report on the EDVAC—the yet-to-be-built successor to the electronic computer designed by Americans John Mauchly and J. Presper Eckert during the war—convinced him which direction to take.

The EDSAC, or Electronic Delay Storage Automatic Calculator, took two-and-a-half years to build. Thirty-two tanks of mercury provided memory by delaying pulses that were sent from an electrically charged quartz crystal. Programs were entered with punched tape. On May 6, 1949, EDSAC successfully computed a table of squares, and the machine remained operational until 1958.

Wilkes was intimately involved with computers for the rest of his career. In 1951, with David Wheeler and Stanley Gill, both research students at the time, he published the first textbook on programming methods. (Recalling those early efforts in his memoir, Wilkes remarked that he quickly realized the remainder of his life would be spent finding errors in his programs.) Later that

year, while laying plans for the EDSAC 2, he hit upon the idea of using a stored program to represent the sequences of control signals within the computer. He called the technique “microprogramming.”

Wilkes received numerous honors during his lifetime, including being the second recipient of the ACM A.M. Turing Award, in 1967.

Wilkes is remembered by colleagues as a thorough and meticulous researcher. “He was relentlessly professionally driven,” says Andrew Hopper, a Cambridge computer science professor who collaborated with Wilkes, Roger Needham, and others in the 1970s on an experimental local area network known as the Cambridge Ring. Post-retirement, Wilkes continued his research through a series of consultancies, working steadily in areas like network systems and multimedia conferencing. “He was completely active every day,” says Hopper. 

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

© 2011 ACM 0001-0782/11/0200 \$10.00

# Following the Crowd

*Crowdsourcing is based on a simple but powerful concept: Virtually anyone has the potential to plug in valuable information.*

**I**F ONE THING is entirely clear about the Internet it's that today's ability to democratize information and tasks is nothing short of remarkable. Increasingly, groups aggregate knowledge through wikis, track incidents during a political uprising or emergency through text messages and email, and create instant teams and organizations in order to solve tasks and accomplish work.

"There's an ability to mobilize information and groups quickly and effectively," observes Peter Lee, former director of the Transformational Convergence Technology Office at the U.S. Defense Advanced Research Project Agency (DARPA) and currently director of research at Microsoft. "This capability is fundamentally changing society—and the way we approach common tasks and problems."

At the heart of this equation is crowdsourcing. The concept revolves around large groups of people or a community handling tasks that have traditionally been associated with a specialist or small group of experts. Jeff Howe, author of *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*, coined the term in 2006, citing technology as a way to draw a greater number of people into large tasks while tapping knowledge and expertise that previously flew under the radar.

Over the last few years, crowdsourcing has emerged as a viable solution for businesses, relief agencies, researchers, politicians, the military, and others looking to grab bits and bytes of information in a nontraditional and decidedly more chaotic way. Howe describes it as a way for many to do the work and tasks previously handled by a few. Crowdsourcing has social, economic, cultural, business, and political implications, he says.

Crowdsourcing is gaining momentum across a wide swath of industries

and organizations. However, the concept isn't without controversy. Some organizations have found that crowdsourcing is expensive and unreliable. Even those that have used it successfully have found that bad data and faulty observations sometimes get tossed into the mix. The old axiom "you get what you pay for" can become glaringly apparent when *anyone* can join the fray.

## Groupthink Unleashed

Crowdsourcing is based on a simple but powerful concept: Virtually anyone has the potential to plug in valuable information. As Howe noted in *Crowdsourcing*, "Technological advances in everything from product design software to digital video cameras are breaking down the cost barriers that once separated amateurs from professionals. Hobbyists, part-timers, and dabblers suddenly have a market for their efforts [as organizations] tap the latent talent of the crowd."

The roots of crowdsourcing extend back to the 1990s. That's when individuals and institutions began volunteering spare computing cycles to help solve major research projects involving everything from mathematical

formulas to medical problems. This community-based approach extended to wikis and other collaboration tools in the age of the Internet. In today's Web 2.0 world, peer-to-peer and collaboration-based platforms play an increasingly important role in an array of fields.

It's safe to say that crowdsourcing is a thoroughly disruptive tool. "Normally," Lee says, "business, science, and high-tech development takes place in fancy laboratories or in academic ivory towers. The idea of taking the development process out to the public is alluring yet intimidating. When such a powerful technology is unleashed it leads to unpredictable and sometimes surprising results."

Many organizations are now developing their own crowdsourcing software or platforms. These systems typically tap into mashups and other Web 2.0 tools accessible through a standard Web browser. In some cases, these applications rely on mapping software such as Google Maps. Others depend on wiki-type software to collect observations, comments, and other pertinent data. Along the way, a person or software application populates a data-



Ushahidi enables volunteers to map everything from natural disasters to political turmoil.

PHOTOGRAPH BY ERIK HERSMAN

base or a mashup to provide actionable information.

One of the most successful commercial crowdsourcing platforms is Amazon's Mechanical Turk. It offers businesses and developers access to an on-demand, scalable work force. Essentially, potential employers post tasks and workers select jobs they would like to perform. Payment is established up front and fund transfers take place through Amazon.com.

In fact, an increasing number of organizations are turning to crowdsourcing to reengineer an array of processes. The military is exploring ways to collect intelligence data through crowdsourcing, government agencies are using it to collect data on everything from road repairs to urban planning, and relief agencies are turning to it to better understand how to focus aid and resources.

One organization that's leading the way with crowdsourcing is Ushahidi. The Web site's software platform—collaboratively written by developers in Ghana, Kenya, Malawi, Netherlands, South Africa, and the U.S.—enables volunteers throughout the world to map everything from natural disasters to political turmoil. The open source platform, now used by dozens of organizations, was developed to track reports of violence in Kenya in the aftermath of its disputed 2007 presidential election.

Ushahidi is based on a simple enough concept: as an event unfolds, a volunteer on the ground sends a brief report through a Web browser or text message and software maps the entry by time and location. Organizations can download the free software and deploy it as they see fit. Ushahidi (which means "testimony" in Swahili) incorporates powerful content management capabilities, a robust database, and server-side map clustering. When data from media, nongovernment organizations, and citizens is tossed together, the result is a mashup that provides powerful geographical mapping tools.

Over the last few years, Ushahidi has been used by organizations to pinpoint medical needs associated with an earthquake in Haiti; monitor local elections in Afghanistan, India, and Mexico; map incidences of violence in Pakistan; track medicine shortages in the Philippines; and analyze human

## In today's Web 2.0 world, peer-to-peer and collaboration-based platforms play an increasingly important role in an array of fields.

rights abuse in the Congo. Its uses are limited only by the creativity and needs of those using it. "It produces real-world results," says Patrick Meier, director of crisis mapping for Ushahidi.

Another organization that has tapped into the power of crowdsourcing is DARPA, the research and development office for the U.S. Department of Defense. "Computer technology has led to all sorts of surprising and disruptive outcomes," Lee explains. "Crowdsourcing offers tremendous potential."

DARPA has experimented with a number of crowdsourcing initiatives, including the DARPA Network Challenge, a balloon hunt that involved more than 4,000 teams attempting to locate 10 moored, 8-foot tall, red weather balloons at 10 fixed locations across the continental U.S. (For more about the DARPA Network Challenge, see "Mechanism Design Meets Computer Science," August 2010.) The agency created incentives for participants to lie, keep secrets, and infiltrate each other's teams. "We wanted to simulate all the dirty tricks that would take place in a wartime military environment," Lee says.

The project encompassed the Internet, social networking, real-time communications, wide-area collaboration, and other crowdsourcing techniques. The winner, a team from Massachusetts Institute of Technology, received a \$40,000 prize. It took the participants nine hours to complete the task. The team used a multi-level marketing strategy to recruit students—paying as much as \$2,000 for information about balloon coordinates and lesser sums to those who invited them (and the person who invited the inviter). DARPA

### Society

## Embedded Chips

Microprocessors are increasingly appearing in a wider variety of devices, ranging from fishing lures to writing pens to otherwise ordinary tombstones, according to chip industry experts, who predict a future in which low-tech products become more like personal computers, smartphones, and other chip-powered "intelligent" devices.

The market for so-called embedded semiconductors accounts for one-half to two-thirds of the \$300 billion a year in chip sales worldwide, and is a fast-growing market segment. "The term 'embedded' used to refer to a low-level, limited-function semiconductor and nobody needed to pay attention to it," says Shane Rau, a chip expert at the market research firm IDC, in a recent interview with the *San Jose Mercury News*. "Now these devices are taking on more intelligence. They're becoming more programmable, they're getting faster, and they're getting communications functions built into them."

For example, Pro-Troll puts chips in its fishing lures, which makes a lure mimic "the electrical nerve discharge of a wounded bait fish," enticing other fish to attack it, according to the Concord, CA, company. Likewise, Livescribe sells an ink pen equipped with a chip, camera, and audio recorder that helps people recall what was said when they review their handwritten notes. And the Memory Medallion is a coin-size, stainless steel-encased chip that, when embedded in a gravestone, tells the deceased person's life story using audio, text, photos, and video.

One trend will be consumer products that have the potential to make life easier for humans by making decisions for them, according to Lori Dolnick, a spokeswoman for Miele. The German company manufactures household appliances, such as washing machines, equipped with semiconductors and wireless capabilities, and enables it to contact a customer before he or she might be aware of a problem with a Miele product.

—Jack Rosenberger



studied the resulting social interaction and is using the data to formulate more advanced wartime strategies.

### Advantages and Disadvantages

“Crowdsourcing offers both advantages and disadvantages,” Ushahidi’s Meier points out. “It is very efficient—with the right community in place—at gathering information quickly and effectively. It can help speed response and cut through the confusion that occurs during the initial stage of a disaster. It can quickly fill the information gap.” What’s more, he says, traditional surveys and techniques require more time and expense—often with less impressive results.

Meier says that concerns about the accuracy of data aren’t unfounded. “One of the challenges is developing trusted sources,” he says. Of course, it’s not possible to vet everyone. And restricting who posts data online defeats the entire purpose of crowdsourcing. In addition, some errors and inaccuracies are inevitable, even from well-meaning participants. “You have to operate under the assumption that most people are honest and most information is accurate,” he adds. “But it’s necessary to build in a margin for error.”

Another challenge is publicizing a crowdsourcing platform and establishing a network of volunteers. It’s a task that requires significant money, time, and effort—something that many non-governmental organizations lack. Lee says that organizations typically publicize efforts any way they can—through press releases, a Web site, and word of mouth. However, higher participation rates translate into a greater volume

**Over the last few years, Ushahidi has been used by organizations to monitor local elections in India, map incidences of violence in Pakistan, and track medicine shortages in the Philippines.**

of data, but sorting through it to spot what’s relevant and useful can prove taxing. “Managing the process can be difficult,” he admits.

Nevertheless, crowdsourcing continues to advance—and involve increasingly complex issues. In some cases researchers and computer scientists are attempting to attack age-old questions and challenges in new ways—and gain fresh perspectives. For example, when Vinay Deolalikar, a renowned computer scientist at Hewlett-Packard labs, sent an email to top researchers claiming that P doesn’t equal NP, it generated considerable interest.

But then, once the issue hit the blog of Richard J. Lipton, a computational complexity expert at the Georgia Institute of Technology, interest among other researchers and a lay audience peaked. An informal peer review pro-

cess followed. Participants discovered errors and the level of interaction and exchange exceeded that of any traditional process. In the end, one researcher described the entire episode as a “Nerd Super Bowl.”

Clearly, crowdsourcing is here to stay. “It is changing the way government, corporations, and others tackle complex issues and problems,” Lee notes. “It is leading to an entirely different mindset about how product development, problem solving, and decision making take place.” **C**

### Further Reading

Brabham, D.C.

Crowdsourcing as a model for problem solving: an introduction and cases, *Convergence: The International Journal of Research into New Media Technologies* 14, 1, Feb. 2008.

Brabham, D.C.

Crowdsourcing the public participation process for planning projects, *Planning Theory* 8, 3, August 2009.

Howe, J.

*Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*, Random House, New York, 2008.

Lakhani, K.R., Jeppesen, L.B.,

Lohse, J.A. and Panetta, P.A.

The value of openness in scientific problem solving, Harvard Business School Working Knowledge, October 2006.

Leimeister, J.M., Huber, M.,

Bretschneider, U., and Krcmar, H.

Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition, *Journal of Management Information Systems* 26, 1, Summer 2009.

Samuel Greengard is an author and freelance writer based in West Linn, OR.

© 2011 ACM 0001-0782/11/0200 \$10.00

### Milestones

## Computer Science Awards

The International Society for Ethics and Information Technology (INSEIT), Institute of Electrical and Electronics Engineers (IEEE), and E.W.R. Steacie Memorial Fund recently honored leading computer scientists.

### INSEIT AWARD

Don Gotterbarn, professor emeritus of computer science and director of the Software

Engineering Ethics Research Institute at East Tennessee State University, received the 2010 INSEIT/Joseph Weizenbaum Award for his contributions to the field of information and computer ethics.

### IEEE AWARDS

C.A.R. Hoare, a principal researcher at Microsoft Research Cambridge, was awarded the

John von Neumann Medal for “seminal contributions to the scientific foundation of software Design.” Shafi Goldwasser, a professor at Massachusetts Institute of Technology and Weizmann Institute of Science, received the Emanuel Piore award for “pioneering work in laying the foundations of modern cryptography and its relation to complexity theory.”

### STEACIE PRIZE

The E.W.R. Steacie Memorial Fund presented the 2010 Steacie Prize for Natural Sciences to Aaron Hertzmann, an associate professor of computer science at the University of Toronto. The award is given annually for exceptional research by a Canadian scientist or engineer aged 40 or younger.

—Jack Rosenberger

# ACM Launches New Digital Library

*More than 50 years of computing literature is augmented, streamlined, and joined to powerful new tools for retrieval and analysis.*

**T**HE ACM HAS just launched a new version of its Digital Library (DL), the first major overhaul of its vast store of computing literature in almost 10 years. The information services offered by the new library have been enhanced and tweaked dramatically, usability and performance have been improved, connections to external services have been added, and much new content, including multimedia, is available. Most fundamentally, the DL has evolved from a relatively simple keyword search and journal article retrieval model to one in which users can see the connections—and the strength of connections—among topics, journals, articles, authors, collaborators, institutions, Special Interest Groups (SIGs), and conferences.

“It’s all about showing users the context in which something fits, so they know that there is more to this space than just their ability to grab an article and go,” says ACM’s Director of Information Services Wayne Graves. “It’s about surfacing and leveraging information in different ways.”

The DL now explicitly recognizes that many users do not work top-down from the library’s home page (the original model) but come in directly at the citation page for a given article—often sent there by an external search engine. “The citation page has become the front door,” explains Graves, the chief architect of the new library system. “So a design goal was to get as much information there as we can.”

Information on the citation page now appears in three logical blocks. The first contains the basic information about the article such as title, author, and publication name; links to author profiles; links to the journal or conference home page, table of contents, and archives; and bibliometrics



**A conference page in the new ACM Digital Library.**

(numbers of downloads and citations). The second block serves up clickable options to buy an article, to request permissions, to save the article to a personal binder or workspace, to export it in various formats, or to send it via email or to external services such as Slashdot, Facebook, and Twitter. The third section has 10 tabs that bring up information about the article, such as source materials, references, citations, reviews, and comments. An “index terms” tab shows pointers to other articles on the same and related topics.

Joshua Hailpern, a Ph.D. candidate in computer science at the University of Illinois at Urbana-Champaign, has used ACM’s digital services for nearly a decade, and recently served on a panel of users to evaluate ACM DL prototypes. “The information used to be sort of dumped on the user in this long, difficult-to-navigate layout,” he says. “What they did is ask, ‘What’s the information users care about first?’ And they kept it simple.”

A key enabler of many of the DL’s new capabilities, and for a number of older

ones, is an architecture that puts a great deal of emphasis on the capture and use of metadata—searching and linkable data about entities in the library, such as publication, author, institution, citation, or SIG. “The amount of metadata that we capture is very large, and that’s absolutely key to our new capabilities,” says Bernard Rous, director of publications. “We made that as robust as possible so we can do all kinds of calculations and manipulations.”

For example, the DL now offers new and powerful ways to look at conferences. Before, a user could retrieve conference proceedings for a past conference, but other information about it, or about future conferences, was not readily at hand. Now a map of the world can be zoomed in on to show ACM conferences in a given city or region. Click on a conference and links to a conference profile page, the conference Web site, and proceedings are revealed. Tag cloud representations provide snapshots of subject area concentration for conferences and SIGs—with differing font sizes showing relative importance—and will soon do so for conference acceptance rates over time via color charts.

The new DL reflects user demand for greater interactivity and control of content. For example, a personal binder capability that had served essentially as a way to save search results in a folder has been greatly expanded. An interactive editor has been added so that, for example, a user can annotate a bibliography. “The WYSIWYG editor and HTML forms to comment on articles and to annotate personal binder collections introduces user-contributed content to the DL and allows it to begin functioning as a workspace,” Rous says. “It is also far easier to share binders more widely, and the user can generate a single PDF that stitches

together all the selected articles and annotations with a cover image. In essence, the new binder function enables the DL to become an authoring and a publishing platform for certain kinds of e-books."

### New Interactive Capabilities

Although ACM already provided one of the most comprehensive global resources for the computing field, a combination of forces prompted this major overhaul of the DL. "User feedback streams in on a daily basis," says Rous. For example, critical feedback from SIGs led to the creation of a Conference Profile page and a SIG Profile page. Analysis of such feedback, augmented by surveys, user workshops, focus groups, and advisory bodies, revealed strong demand for more interactive capabilities and for richer data views.

"Comments relating to change requests are collected, analyzed, and finally placed on a prioritized to-do list," explains Rous. "ACM's Librarian Advisory Group, Digital Library Advisory Board, and Publications Board have all reviewed and critiqued early versions of new functionality. Tests are run with focus groups and, finally, beta production versions are released for further feedback and refinement. The new DL reflects the collective wisdom of a broad community participating in the development process."

The new DL also reflects the reality that many users have dropped their print subscriptions, yet they occasionally want to order a printed copy of an issue. New e-commerce functions provide ways for online-only subscribers to order a print copy and for nonsubscribers to gain access to online content. "A print-on-demand function is sorely needed and will shortly appear alongside the other options," Rous says.

The trend from print to online is also a challenge to external libraries, Rous says, and they have responded by installing complex local infrastructures and by tapping into external systems to manage their digital resources. "ACM is committed to supporting libraries in these transformative efforts," he says.

One way ACM is doing that is by providing access to its increasingly rich metadata, with or without extracted full-texts, for local or third-party index-

## The new ACM Digital Library is more interactive, and puts a lot of emphasis on the capture and use of metadata.

ing to facilitate cross-platform discovery by the library's patrons.

Rous says ACM aims to make the DL the most reliable source for citation and usage statistics in the field, and to make that kind of information useful for the assessment of research and the contributions and influence of individuals and institutions.

Part of the DL's new look is a result of rebranding in response to user confusion over the differences among the terms ACM Digital Library, ACM Guide to Computing Literature, and ACM Portal, with most users simply lumping everything under the "Digital Library" moniker. The full-text library of ACM publications remains, as does the ACM Guide, the larger bibliographic database of computing literature. But these are now more tightly integrated and function as a single resource branded as the ACM Digital Library.


Jill Powell, a librarian at Cornell University's engineering library, says computer science faculty and students have for years used the library in traditional ways—"to discover articles by keyword and author"—but that usage of it will become more extensive now. She calls the new bibliometrics "very interesting" and says the features for automatic subject and table of contents alerts via email and RSS are attractive. "It's a rich site, it has a lot of features, and it's easy to use," Powell says.

A number of enhancements have been made "under the hood," says Graves, who wrote most of the new code. The pages of the DL are dynamically generated and the data is retrieved from precomputed data sets supported by a new, flatter data model, resulting in a much faster front-end interface. And a new caching scheme

fetches only the data needed, rather than the templates and logic required to generate the pages. The library served up about 10 pages per second, on average, under the old architecture but now runs at about 15 pages per second, Graves says. The library contains 1.6 million items, and has users in 190 countries, who download 13 million full-text files and conduct 12 million searches annually.

Rankings of the "influence" of people and institutions, as measured by such data as numbers of publications and their citations and usage history, as well as the frequency and kinds of collaboration, are on the short-term horizon. The new library takes the first small steps toward reporting measures of that influence. The Author Profile page, for example, supplies a snapshot of an author's contribution to the field with coauthors listed as "colleagues." The data that underlies these pages could enable views of professional networks. Indeed, several SIGs are using this data in research projects to produce visualizations of author relationships, Rous says.

Similarly, a beta version of Institutional Profiles aggregates all the information in the bibliographic database to give snapshots of the faculty and graduate students affiliated with that institution at the time of their publications. Their output is weighted and their publications are listed. The subject areas of greatest concentration at the institution are visually displayed. These profiles reveal for the first time the frequency of author collaborations across different institutions.

"This to me is the really exciting direction of the DL," Hailpern says. "There is amazing stuff that can be done with information networks—not just saying here's the word-count frequencies in a tag cloud, but really saying, 'How influential is this paper, what are the terms and areas and authors this paper has influenced? What is the world it relates to?'" This is the kind of stuff that ACM is uniquely positioned to leverage because no other community is so in touch with these information networks." 

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

© 2011 ACM 0001-0782/11/0200 \$10.00



# ACM Fellows Honored

*Forty-one men and women are inducted as 2010 ACM Fellows.*

**T**HE ACM FELLOW Program was established by Council in 1993 to recognize and honor outstanding ACM members for their achievements in computer science and information technology and for their significant contributions to the mission of the ACM. The ACM Fellows serve as distinguished colleagues to whom ACM and its members look for guidance and leadership as the world of information technology evolves.

The ACM Council endorsed the establishment of a Fellows Program and provided guidance to the ACM Fellows Committee, taking the view that the program represents a concrete benefit to which any ACM member might aspire, and provides an important source of role models for existing and prospective ACM Members. The program is managed by the ACM Fellows Committee as part of the general ACM Awards program administered by Calvin C. Gotlieb and James J. Horning. For details on Fellows nominations, see p. 14.

ACM has recognized 41 of its members for their contributions to computing and computer science that have provided fundamental knowledge to the field and generated multiple innovations in industry, commerce, entertainment, and education. The 2010 ACM Fellows, from the world's leading universities, corporations, and research labs, achieved accomplishments that are driving the innovations necessary to sustain competitiveness in the digital age. These 41 new inductees bring the total number of ACM Fellows to 726 (see <http://www.acm.org/awards/fellows/> for the complete listing of ACM Fellows). ACM will formally recognize the 2010 Fellows at its annual Awards Banquet on June 4, 2011, in San Jose, CA.

"These men and women have made advances in technology and contributions to the computing community that are meeting the dynamic demands of the 21st century," said ACM Presi-

dent Alain Chesnais. "Their ability to think critically and solve problems creatively is enabling great advances on an international scale. The selection of this year's Fellows reflects broad international representation of the highest achievements in computing, which are advancing the quality of life throughout society."

## ACM Fellows

**David Abramson**,  
Monash University

**Sarita Adve**,  
University of Illinois  
at Urbana-Champaign

**Lorenzo Alvisi**,  
The University of Texas at Austin

**Luiz André Barroso**,  
Google Inc.

**Doug Burger**,  
Microsoft Research

**Jennifer Chayes**,  
Microsoft Research  
New England Lab

**Peter M. Chen**,  
University of Michigan

**Anne Condon**,  
University of British Columbia

**Mark Crovella**,  
Boston University

**Ron K. Cytron**,  
Washington University

**Michael Dahlin**,  
The University of Texas at Austin

**Amr El Abbadi**,  
University of California, Santa Barbara

**Carla Ellis**,  
Duke University

**Christos Faloutsos**,  
Carnegie Mellon University

**Kathleen Fisher**,  
AT&T

**James Goodman**,  
University of Auckland

**Professor Dame Wendy Hall**,  
University of Southampton

**Jean-Pierre Hubaux**,  
EPFL (École Polytechnique  
Fédérale de Lausanne)

**Michael Jordan**,  
University of California, Berkeley

**Lydia Kavraki**,  
Rice University

**Sara Kiesler**,  
Carnegie Mellon University

**Philip Klein**,  
Brown University

**Donald Kossmann**,  
ETH Zurich (Swiss Federal  
Institute of Technology)

**John Launchbury**,  
Galois

**Richard F. Lyon**,  
Google Inc.

**Raymond Mooney**,  
The University of Texas at Austin

**S. Muthukrishnan**,  
Rutgers University/Google Inc.

**Fernando Pereira**,  
Google Inc.

**Pavel Pevzner**,  
University of California, San Diego

**Dieter Rombach**,  
University of Kaiserslautern  
and the Fraunhofer Institute  
for Experimental Software  
Engineering (IESE),  
Kaiserslautern, Germany

**David Rosenblum**,  
University College London

**Stefan Savage**,  
University of California, San Diego

**Robert Schnabel**,  
Indiana University

**Daniel Spielman**,  
Yale University

**Subhash Suri**,  
University of California,  
Santa Barbara

**Frank Tompa**,  
University of Waterloo

**Josep Torrellas**,  
University of Illinois  
at Urbana-Champaign

**Stephen Trimberger**,  
Xilinx Research Labs

**David Ungar**,  
IBM Thomas J. Watson Research  
Center

**Andreas Zeller**,  
Saarland University

**Shumin Zhai**,  
IBM Almaden Research Center

# Privacy and Security Against Cyberterrorism

*Why cyber-based terrorist attacks are unlikely to occur.*

**L**IKE THE 2007 cyber attacks on Estonia, the October 2010 Stuxnet botnet attack on Iranian nuclear facilities made cyber-based attacks global news. The Estonian attacks were largely labeled a cyberwar by journalists, although some did invoke the concept of cyberterrorism. The Stuxnet attack, on the other hand, has been very widely described as cyberterrorism, including by the Iranian government.

Cyberterrorism is a concept that appears recurrently in contemporary media. It is not just reported upon in newspapers and on television, but is also the subject of movies (such as 1990's *Die Hard II* and 2007's *Die Hard IV: Live Free or Die Hard*) and popular fiction books (for example, Winn Schwartau's 2002 novel *Pearl Harbor Dot Com*). This coverage is particularly interesting if one believes, as I do, that no act of cyberterrorism has ever yet occurred and is unlikely to at any time in the near future. Having said that, it is almost always portrayed in the press as either having already occurred or being just around the corner. As an academic, I'm not alone in

arguing that no act of cyberterrorism has yet occurred and, indeed, some journalists agree; most, however, seem convinced as to the salience of this threat. Why?

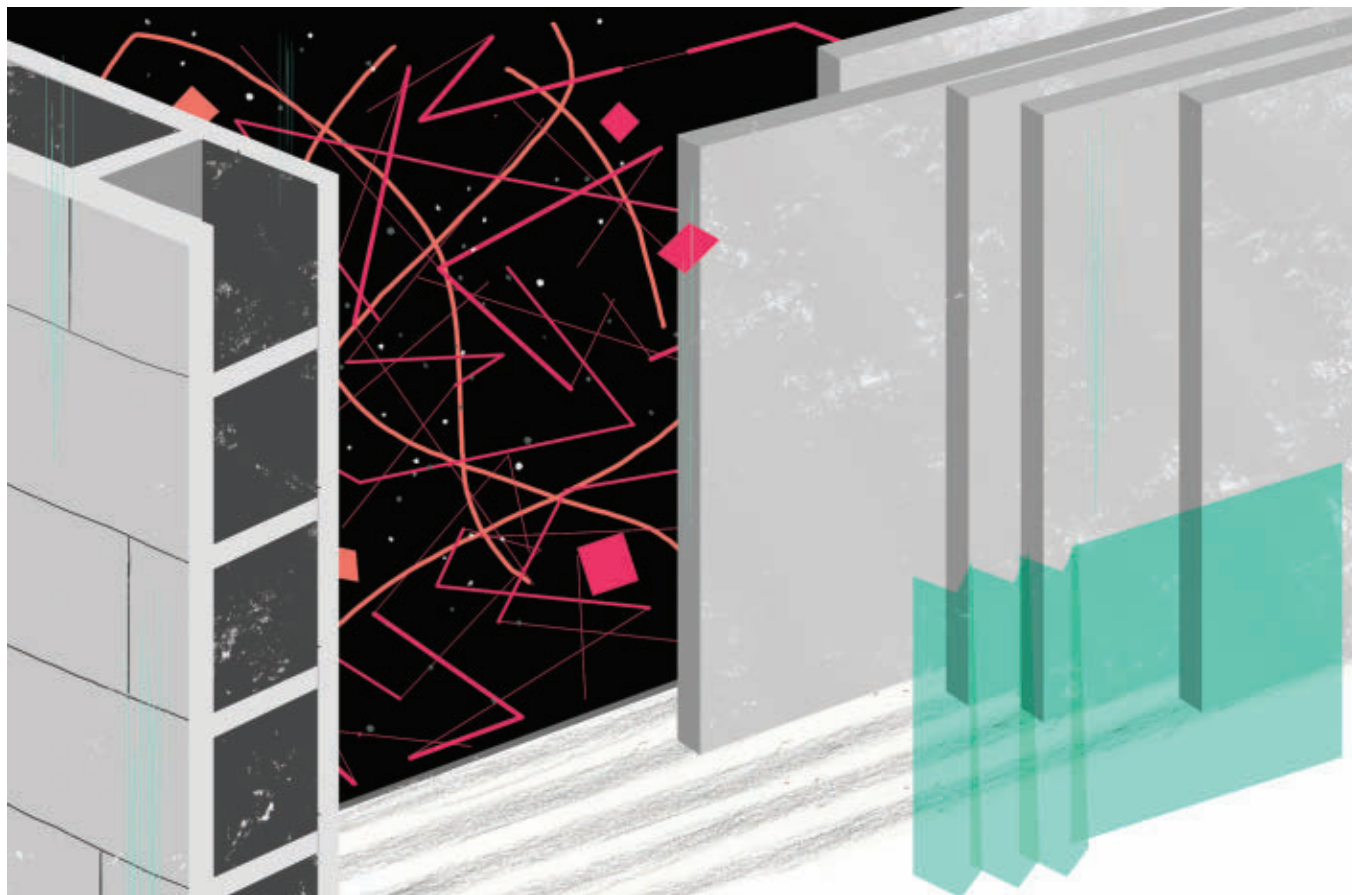
I can only surmise that, just as a large amount of social psychological research has shown, the uncertain and the unknown generally produce fear and anxiety. This is the psychological basis of an effective movie thriller: the fear is greatest when you suspect something, but you're not certain what it is. The term "cyberterrorism" unites two significant modern fears: fear of technology and fear of terrorism. Fear of terrorism, though the likelihood of any one of us being the victim of terrorism is statistically insignificant, has become perhaps normalized; but fear of technology? In fact, for those unfamiliar with the workings of complex technologies, these are perceived as arcane, unknowable, abstract, and yet increasingly powerful and ubiquitous. Many people therefore fear that technology will become the master and humankind the servant. Couple this relatively new anxiety with age-old fears associated with apparently random violence

and the result is a truly heightened state of alarm. Many journalists—although fewer technology journalists than others—have succumbed, like members of the general population, to these fears, to which the journalists have then added further fuel with their reporting.

## The Definition Issue

The second stumbling block for journalists is that just as the definition of terrorism is fraught, so too is the definition of cyberterrorism. My preference is to distinguish between cyberterrorism and terrorist use of the Net. This is the distinction FBI Director Robert Mueller seemed implicitly to be drawing in a March 2010 speech in which he stated that "the Internet is not only used to plan and execute attacks; it is a target in and of itself... We in the FBI, with our partners in the intelligence community, believe the cyber terrorism threat is real, and it is rapidly expanding."<sup>a</sup> Where the FBI

<sup>a</sup> The text of Director Mueller's March 2010 speech at a cyber security conference in San Francisco is available at <http://www.fbi.gov/pressrel/speeches/mueller030410.htm>.



Director and I diverge is in the efficacy of the cyberterrorist threat as opposed to that of everyday terrorist use of the Net (that is, for radicalization, researching and planning, financing, and other purposes).

Dorothy Denning's definitions of cyberterrorism are probably the most well known and respected. Her most recent attempt at defining cyberterrorism is: "...[H]ighly damaging computer-based attacks or threats of attack by non-state actors against information systems when conducted to intimidate or coerce governments or societies in pursuit of goals that are political or social. It is the convergence of terrorism with cyberspace, where cyberspace becomes the means of conducting the terrorist act. Rather than committing acts of violence against persons or physical property, the cyberterrorist commits acts of destruction or disruption against digital property."<sup>2</sup>

Analyses of cyberterrorism can be divided into two broad categories on the basis of where the producers stand on the definition issue: those who agree broadly with Denning versus those who wish to incorporate not just

use, but a host of other activities into the definition. The literature can also be divided on the basis of where the authors stand on the magnitude of the cyberterrorism threat. Dunn-Cavelty uses the term "Hypers" to describe those who believe a cyberterrorist attack is not just likely, but imminent,<sup>b</sup> and the term "De-Hypers" to describe those who believe such an attack is unlikely.<sup>1</sup> Most journalists are hypers, on the other hand I'm emphatically a de-hyper. In this column, I lay out the three major reasons why.

### Three Arguments Against Cyberterrorism

In my opinion, the three most compelling arguments against cyberterrorism are:

- ▶ The argument of Technological Complexity;
- ▶ The argument regarding 9/11 and the Image Factor; and
- ▶ The argument regarding 9/11 and the Accident Issue.

<sup>b</sup> See, for an exemplary example, journalist Dan Verton's *Black Ice: The Invisible Threat of Cyberterrorism*. McGraw-Hill, New York, 2003.

The first argument is treated in the academic literature; the second and third arguments are not, but ought to be. None of these are angles to which journalists appear to have devoted a lot of thought or given adequate consideration.

In the speech mentioned earlier, FBI Director Mueller observed "Terrorists have shown a clear interest in pursuing hacking skills. And they will either train their own recruits or hire outsiders, with an eye toward combining physical attacks with cyber attacks." That may very well be true, but the argument from Technological Complexity underlines that 'wanting' to do something is quite different from having the ability to do the same. Here's why:

**Violent jihadist' IT knowledge is not superior.** For example, in research carried out in 2007, it was found that of a random sampling of 404 members of violent Islamist groups, 196 (48.5%) had a higher education, with information about subject areas available for 178 individuals. Of these 178, some 8 (4.5%) had trained in computing, which means that out of the entire sample,



less than 2% of the jihadis came from a computing background.<sup>3</sup> And not even these few could be assumed to have mastery of the complex systems necessary to carry out a successful cyberterrorist attack.

**Real-world attacks are difficult enough.** What are often viewed as relatively unsophisticated real-world attacks undertaken by highly educated individuals are routinely unsuccessful. One only has to consider the failed car bomb attacks planned and carried out by medical doctors in central London and at Glasgow airport in June 2007.

**Hiring hackers would compromise operational security.** The only remaining option is to retain “outsiders” to undertake such an attack. This is very operationally risky. It would force the terrorists to operate outside their own circles and thus leave them ripe for infiltration. Even if they successfully got in contact with “real” hackers, they would be in no position to gauge their competency accurately; they would simply have to trust in same. This would be very risky.

So on the basis of technical know-how alone cyberterror attack is not imminent, but this is not the only factor one must take into account. The events of Sept. 11, 2001 underscore that for a true terrorist event spectacular moving images are crucial. The attacks on the World Trade Center were a fantastic piece of performance violence; look back on any recent roundup of the decade and mention of 9/11 will not just be prominent, but pictures will always be provided.

The problem with respect to cyberterrorism is that many of the attack scenarios put forward, from shutting down the electric power grid to contaminating a major water supply, fail on this account: they are unlikely to have easily captured, spectacular (live, moving) images associated with them, something we—as an audience—have been primed for by the attack on the World Trade Center on 9/11.

The only cyberterrorism scenario that would fall into this category is interfering with air traffic control systems to crash planes, but haven't we seen that planes can much more easily be employed in spectacular “real-world” terrorism? And besides,

## The term “cyberterrorism” unites two significant modern fears: fear of technology and fear of terrorism.

aren't all the infrastructures just mentioned much easier and more spectacular to simply blow up? It doesn't end there, however. For me, the third argument against cyberterrorism is perhaps the most compelling; yet it is very rarely mentioned.

In 2004, Howard Schmidt, former White House Cybersecurity Coordinator, remarked to the U.S. Senate Committee on the Judiciary regarding Nimda and Code Red that “we to this day don't know the source of that. It could have very easily been a terrorist.”<sup>4</sup> This observation betrays a fundamental misunderstanding of the nature and purposes of terrorism, particularly its attention-getting and communicative functions.

A terrorist attack with the potential to be hidden, portrayed as an accident, or otherwise remain unknown is unlikely to be viewed positively by any terrorist group. In fact, one of the most important aspects of the 9/11 attacks in New York from the perpetrators viewpoint was surely the fact that while the first plane to crash into the World Trade Center could have been accidental, the appearance of the second plane confirmed the incident as a terrorist attack in real time. Moreover, the crash of the first plane ensured a large audience for the second plane as it hit the second tower.

Alternatively, think about the massive electric failure that took place in the northeastern U.S. in August 2003: if it was a terrorist attack—and I'm not suggesting that it was—but *if it was*, it would have been a spectacular failure.

### Conclusion

Given the high cost—not just in terms

of money, but also time, commitment, and effort—and the high possibility of failure on the basis of manpower issues, timing, and complexity of a potential cyberterrorist attack, the costs appear to me to still very largely outweigh the potential publicity benefits. The publicity aspect is crucial for potential perpetrators of terrorism and so the possibility that an attack may be apprehended or portrayed as an accident, which would be highly likely with regard to cyberterrorism, is detrimental. Add the lack of spectacular moving images and it is my belief that cyberterrorism, regardless of what you may read in newspapers, see on television, or obtain via other media sources, is not in our near future.

So why then the persistent treatment of cyberterrorism on the part of journalists? Well, in this instance, science fiction-type fears appear to trump rational calculation almost every time. And I haven't even begun to discuss how the media discourse has clearly influenced the pronouncements of policymakers.<sup>c</sup> G

c For more on the issues relating to media coverage of cyberterrorism raised in this column, including analysis of the pronouncements of policymakers in this regard, see “Media, Fear and the Hyperreal: The Construction of Cyberterrorism as the Ultimate Threat to Critical Infrastructures.” In M.D. Cavelti and K.S. Kristensen, Eds., *Securing “The Homeland”: Critical Infrastructure, Risk and (In)Security* (Ashgate, London, 2008), 109–129.

### References

1. Cavelti, M.D. Cyber-Terror: Looming threat or phantom menace? The framing of the U.S. cyber-threat debate. *Journal of Information Technology and Politics* 4, 1 (2007).
2. Denning, D. A view of cyberterrorism five years later. In K. Himma, Ed., *Internet Security: Hacking, Counterhacking, and Society* (Jones and Bartlett Publishers, Sudbury, MA, 2006), 124.
3. Gambetta, D. and Hertog, S. Engineers of Jihad. *Sociology Working Papers, No. 2007–10*, Department of Sociology, University of Oxford, (2007), 8–12; <http://www.nuff.ox.ac.uk/users/gambetta/Engineers%20of%20Jihad.pdf>.
4. Virtual Threat, Real Terror: Cyberterrorism in the 21<sup>st</sup> Century (Serial No. J–108–58), hearing before the Subcommittee on Terrorism, Technology and Homeland Security of the Committee on the Judiciary, United States Senate, 108<sup>th</sup> Congress, Second Session, (Feb. 4, 2004), [http://cip.gmu.edu/archive/157\\_S108VirtualThreathearings.pdf](http://cip.gmu.edu/archive/157_S108VirtualThreathearings.pdf).

**Maura Conway** (maura.conway@dcu.ie) is Lecturer in International Security in the School of Law and Government at Dublin City University in Dublin, Ireland.

# Economic and Business Dimensions Household Demand for Broadband Internet Service

*How much are consumers willing to pay for broadband service?*

**W**HAT IS THE market demand for broadband by U.S. households? It is well known that demand for broadband Internet has grown substantially in the last decade. That growth has driven the demand for computers, routers, fiber-optic cable and much more. Our research puts some statistical numbers behind this widely recognized growth.<sup>2</sup> We estimated consumer willingness-to-pay (WTP) for broadband service in late 2009 and early 2010. Our WTP estimates show a high valuation for broadband for average consumers, and even for inexperienced users.

There are many proposals for increasing the deployment and use of broadband infrastructure, most importantly the FCC's National Broadband Plan (see <http://www.broadband.gov/plan/>). Formal cost-benefit evaluation of these proposals requires, among other things, some understanding of the potential benefits from more widespread access to broadband Internet service. Our study provides more understanding of those benefits.

## Experimental Design

Choice experiments were used to estimate household preferences for the different features that comprise Internet service, including the basic service



**Massachusetts Governor Deval Patrick signing legislation to make high-speed Internet available in the state's 32 communities lacking access to broadband service.**

features speed and reliability, in addition to recent and hypothetical uses of the Internet. A carefully designed choice experiment manipulates the features for a series of hypothetical Internet services to obtain the variation in the data needed to estimate the parameters precisely. Choice experiments also allow us to estimate the marginal utilities (satisfaction) for

service features that are not currently available or are only available in limited geographical areas.

The WTP for Internet service is estimated with data from a nationally representative online survey with more than 6,200 respondents. Respondents face repeated discrete-choice experiments. Respondents choose from a pair of hypothetical Internet service

# ACM Transactions on Accessible Computing



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

[www.acm.org/taccess](http://www.acm.org/taccess)  
[www.acm.org/subscribe](http://www.acm.org/subscribe)



Association for  
Computing Machinery

## Estimated valuation for Internet service (\$ per month).

Features	Basic	Reliable	Premium	Premium Plus
Speed	Fast	Fast	Fast	Fast
Reliability	Less reliable	Very reliable	Very reliable	Very reliable
Priority	No	No	Yes	Yes
Telehealth	No	No	No	Yes
Mobile Laptop	No	No	No	Yes
Videophone	No	No	No	Yes
Movie Rental	No	No	No	Yes
All Users	\$59.10	\$78.98	\$85.35	\$98.09
	(\$0.50)	(\$0.66)	(\$1.09)	(\$1.64)
Inexperienced Users	\$30.74	\$40.80	\$58.69	\$71.00
	(\$2.35)	(\$2.67)	(\$4.63)	(\$8.05)

Note: Standard error of estimated valuation in parentheses.

alternatives, labeled A and B. A follow-up question is then presented that asks respondents to make an additional choice between their hypothetical choice—A or B—and their “status quo” Internet service they now experience at their homes.

The hypothetical alternatives differ by the levels of the three Internet features: cost, speed, and reliability, and *one* of five Internet activities: the ability to designate some uploads and downloads as high priority (labeled “Priority” in the accompanying table); the ability to interact with their health professionals and view their records online (“Telehealth”); the ability to connect remotely regardless of WiFi availability (“Mobile Laptop”); a built-in Skype-like service (“Videophone”); and a video-on-demand movie service

for viewing full-length movies (“Movie Rental”).

We estimate the WTP for a subsample of experienced users, as well as for a subsample of inexperienced users (that is, those with less than 12 months of online experience). This difference provides some indication of valuations for households that have recently connected to the Internet.

## Empirical Results

Speed and reliability are important features of Internet service with consumers willing to pay approximately \$20 per month for more reliable service, \$45 for an improvement in speed from slow to fast, and \$48 for an improvement in speed from slow to very fast. These estimates indicate a very fast service is worth only about \$3 more than fast service. The latter finding requires an explanation. As it turns out, the typical U.S. household is involved in Internet activities and applications at home that do not require very fast download and upload speeds, such as reading and writing email or light Web usage. Households are sensitive to other aspects of their service. They are also willing to pay an additional \$6 per month for Priority, \$4 for Telehealth, \$5 for Videophone, and \$3 for Movie Rental. Mobile Laptop is not valued by respondents.

For comparison with the findings of previous studies, we calculate the own-price elasticity of the demand for broadband Internet of  $-0.44$ , which is less elastic than estimates using older

**The typical U.S. household is involved in Internet activities and applications at home that do not require very fast download and upload speeds.**



data. This means consumers are less willing now to give up broadband service if prices increase because they value it more highly than they had previously. In many households the cost of broadband is treated as a necessary expenditure.

Household valuations for Internet service may vary with the number of years the household has been connected to the Internet. Inexperienced households with slow-speed connections are willing to pay about \$16–\$17 per month for an improvement from slow to fast speed, but they do not value an improvement from fast to very fast speed. Inexperienced households with a high-speed connection are willing to pay approximately \$26–\$27 per month for an improvement from slow to fast speed.

To better illustrate our results, the accompanying table shows four levels of Internet service. Basic service has fast speed and less reliable service. Reliable Internet service has fast speed and very reliable service. Premium service has fast speed, very reliable service, and the ability to designate some downloads as high priority. Premium Plus service has fast speed, very reliable service, plus all other activities bundled into the service. We then assume that household valuation for a dialup-like service: less reliable, slow, and with no other special activities, is \$14 per month.

The estimates shown in the table suggest the representative household would be willing to pay \$59 per month for a Basic service, \$79 for a Reliable service, \$85 for a Premium service, and \$98 for a Premium Plus service. The bottom half of the table shows that an inexperienced household with a slow connection would be willing to pay \$31 per month for a Basic service, \$41 for a Reliable service, \$59 for a Premium service, and \$71 for a Premium Plus service.

These estimates suggest the value of broadband has changed in the last decade. In 2003 the representative household was willing to pay approximately \$46 per month for Reliable broadband service<sup>3</sup> compared to about \$79 in 2010. Given that the price of broadband has not changed much in this period, these estimates suggest experienced households get more for

## Experienced users are more aware of the full range of economic, entertainment, information, and social benefits the Web has to offer.

their money today than in the recent past. Using the language of economics, we would say that monthly consumer surplus per household has increased substantially between 2003 and 2010.

### Policy Implications


Choice experiments show that reliability and speed are important features of Internet service, but that very fast Internet service is not worth much more to households than fast service at this point in time. Using these results, we calculate that a representative household would be willing to pay about \$59 per month for Basic Internet service. In contrast, an inexperienced household with a slow connection would be willing to pay about \$31 per month for Basic Internet service.

One interpretation of these results is that experienced users are more aware of the full range of economic, entertainment, information, and social benefits that the Web has to offer. Inexperienced users may also have less technical ability when using high-technology goods and service. As such, they would be relatively less productive when using the Internet to produce household income and/or savings in time. The large increase in WTP indicates the Internet is not only becoming a much more important part of people's lives, possibly because they are learning more about its capabilities, but also that its capabilities have increased through new

applications and increased reliability.

Our data does not allow us to differentiate two potential explanations—households with a higher WTP for Internet subscribed sooner or experience increases valuation. Likely both contribute to the difference. The FCC's National Broadband Plan discusses a number of proposals for subsidizing broadband. Our findings stress the importance of a well-targeted subsidy program over a poorly targeted one.

If experience causes increased valuation, then correctly targeted private or public programs have the potential to increase overall broadband penetration in the U.S. (see Akerberg et al.<sup>1</sup>). These programs could educate households about the benefits from broadband (for example, digital literacy training), expose households to the broadband experience (for example, public access), and/or directly support the initial take-up of broadband (for example, discounted service and/or hookup fees).

On the other hand, the high WTP for broadband indicates that large and poorly targeted subsidy programs could be quite wasteful. Subsidizing broadband, especially for non-poor households in higher-cost areas, may not be an efficient use of government spending because many such households would have been willing and able to pay market prices for broadband. As a result, the subsidy would have no effect on broadband adoption for these households. 

### References

1. Akerberg, D., Riordan, M., Rosston, G., and Wimmer, B. Low-income demand for local telephone service: Effects of lifeline and linkup. SIEPR Discussion Paper, 08-47. Stanford University, Palo Alto, 2009.
2. Rosston, G.L., Savage, S.J., and Waldman, D.M. Household demand for broadband Internet in 2010. *The B.E. Journal of Economic Analysis & Policy* 10, 1 (Advances), Article 79 (2010).
3. Savage, S. and Waldman, D. United States demand for Internet access. *Review of Network Economics* 3, (2004), 228–247.

**Gregory Rosston** (grosston@stanford.edu) is the deputy director of the Stanford Institute for Economic Policy Research and the deputy director of the Public Policy program at Stanford University, Stanford, CA.

**Scott Savage** (Scott.Savage@colorado.edu) is an associate professor who teaches microeconomics and telecom economics at the University of Colorado at Boulder.

**Donald Waldman** (Waldman@colorado.edu) is a professor and associate chair of the Graduate Program at the University of Colorado at Boulder.

Copyright held by author.

## Inside Risks

# The Growing Harm of Not Teaching Malware

*Revisiting the need to educate professionals to defend against malware in its various guises.*

**A**T THE RISK of sounding a byte alarmist, may I call to your attention the extreme threat to our world posed by cyberwar, cyberterrorism, and cybercrime? Cyberattacks are already numerous and intricate, and the unquestionable trend is up. To grasp the likelihood of these threats, consider the similarities between physical and virtual violence. Before attacking the U.S. on Sept. 11, 2001, terrorists rehearsed their assaults on a smaller scale at the World Trade Center and in several more distant venues.

Since that infamous date, paralleling physical attacks, cyberstrikes of increasing severity have been carried out against many targets. A few small nations have been temporarily shut down. These attacks are proofs of concept waiting to be scaled up. I hope cybersecurity is on governments' front burners. We ought not wait to react until a devastating cyber-onslaught is unleashed upon us.

Six years ago I wrote a *Communications* Inside Risks column urging that viruses, worms, and other malware be taught ("Not Teaching Viruses and Worms Is Harmful," Jan. 2005, p. 144). The goal of that column was to involve future generations of computer professionals in the expanding global malware problem and persuade them to help curb it. Six years later, malware is still not being taught. And the problem is now much worse.

### Malware Evolution

During the first decade of the 21<sup>st</sup> century the malware problem has evolved in two significant ways. Gone are the lethal but simplistic payloads, produced by improvised, amateur scripts. Gone also are the idiots savants who cut-and-pasted such scripts. Carders, script kiddies, spammers, identity thieves, and other low-level miscreants will probably and deplorably never be completely gone. Gangs of much better trained programmers have largely replaced the individual crooks and nuisance makers. These gangs ply their trade for or in behalf of political syndicates, organized crime cartels, and government-sanctioned but unacknowledged dark ops. Some nation-states covertly train and support them.

What began as gross mischief evolved into criminal activity. Rather than erasing a hard disk drive, why not steal the data stored on it? Or encrypt the drive and extort a ransom for de-

crypting it? Or hijack the users' computers? Today's malware is a killer app: obfuscated, often; clumsy, never. A medley of viruses, worms, trojans, and rootkits, it is clever, enigmatic—a sly hybrid. Its bureaucratic components (such as installers and updaters) are examples of automated elegance.

Identity theft, botnetting, and many other forms of trespass and larceny continue. Coupled with negligence by institutions that are supposed to safeguard our privacy, the picture is bleak. Malware launchers seem to be always ahead. And their products are no longer stupid capers but skillful software packages. These are valuable lessons that are not being understood by us, the victims.

Malware perpetrators have clearly mastered these lessons. Trading local pranks for global villainy, the perps are readying their next steps on the international political stage, where cyberspace is a potential war zone in-the-making. Inadequately capable of defending ourselves from being burgled, we are easy targets for evil geniuses plotting fresh hostilities.

We cannot protect ourselves from what we do not know. We must not remain stuck in a weak, purely reactive, defensive mode. New malware should no longer be an unexpected, unpleasant surprise. And we must be embarrassed when anti-malware products cause more problems than they solve. As human beings, we have a duty to

**Today's malware is a killer app: obfuscated, often; clumsy, never.**

make our world a better place. As computer professionals, we must do our fair share to stanch malware and prevent cyberwar.

### Dealing with Malware

The malware problem must be dealt with on many fronts, proactively. Ideally, we should anticipate and be prepared for new malware. On the research front, funding agencies should follow DARPA's example. If synthetic genomics—the fabrication of new genetic material—merits \$50 million in grants per year, so should exploration of new, novel, innovative malware.

University classrooms and laboratories should serve as locations for spreading malware literacy. Understanding is achieved only by doing. The most effective way to comprehend something is to program it. We cannot afford to continue conferring degrees to computer majors who have never seen the source code of viruses, worms, trojans, or rootkits, never reversed any malware binaries, and never programmed their own malware.

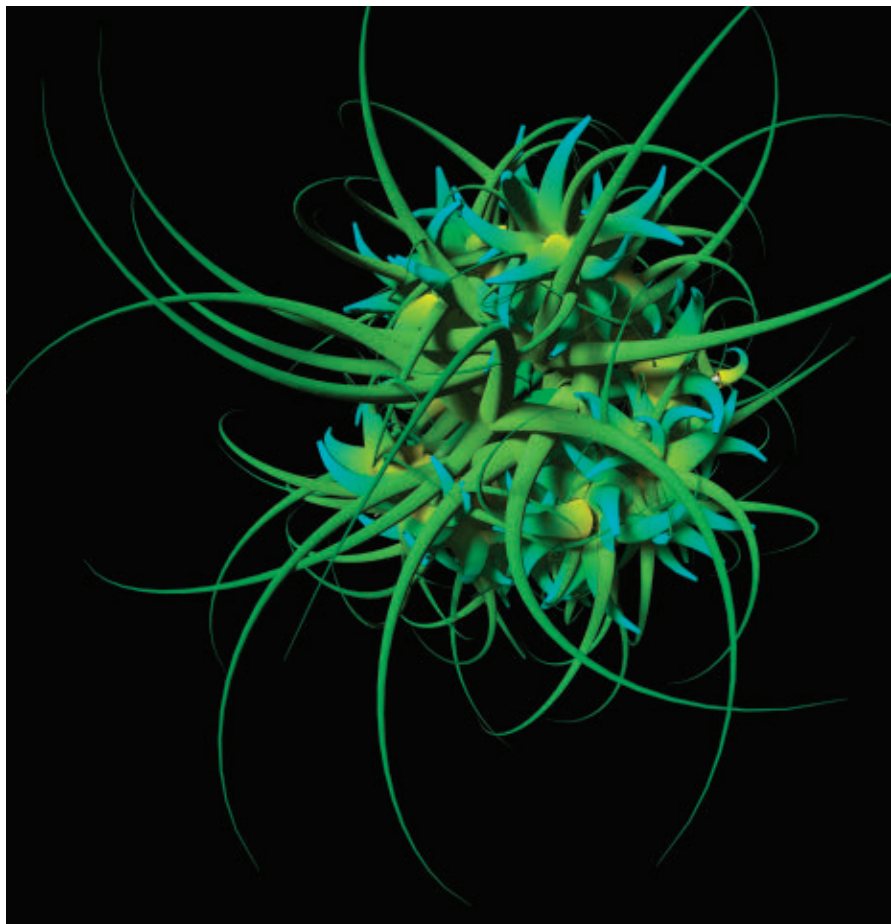
Standard undergraduate computer science curricula offer courses on many disparate topics, such as artificial intelligence and database systems. Students graduating with a degree in computer science are expected to have a solid acquaintance with various subjects that may not be their chosen specialty. Some graduates will dig deeper and become adept at these topics, but the mere fact that these topics are routinely taught to all undergraduate majors is in itself beneficial, because future computer professionals should not be completely ignorant in fields outside their areas of concentration.

Teaching malware will not turn our students into specialists. Malware literacy is not malware expertise. However, unlike artificial intelligence or databases, unfortunately malware is not a standard undergraduate course or even a regular part of an elective computer security course. (Syllabi of computer security courses may pay lip service to diverse issues, including malware, but such courses are overwhelmingly concerned with cryptography.) This means we are matriculating computer scientists whose knowledge of malware is roughly on

a par with that of the general population of amateur computer users.

Six years and many articles, interviews, and blogs later, the question, “Should we teach malware?” still evokes apprehension, trepidation, even dread. The answer, of course, is, “Yes, we should.” Indeed, we must! It would be irresponsible not to have a single course dedicated exclusively to malware, or a course that studies vulnerabilities in general and malware in particular, or some other combina-

tures. How else could aspiring physicians and surgeons learn anatomy? Today, life science majors are not necessarily bacteriologists, parasitologists, or virologists, but all enjoy the benefit of a standard curriculum that offers exposure to microbiology theory and its laboratory practice. This is not the case with computer science majors, whose curricula omit theory and programming of malware. Sadder yet, undergraduates learn sorting, database, and other theories, and carry



Visualization derived from disassembled code of MyDoom worm.

tion, so that students completing the course will gain a deeper understanding of malware.

The apprehension, trepidation, and dread will not go away easily. Spreading viruses, worms, Trojans, and rootkits is dirty business. Programming them may feel like doing something forbidden. Over the past six years, I've heard many concerns about the ethics of teaching malware. Taboos are difficult to dispel. For example, the prohibition of dissecting cadavers held back medicine for cen-

out their corresponding programming assignments, but do not take a similarly rigorous course on malware.

Six years ago, when I proposed that not teaching malware was harmful, I was worried that new malware would attain greater sophistication, become much more complex, and that its force and impact would be felt more widely than those of its predecessors. Well, guess what? It has!

The reason we cannot solve the malware problem is simple: We don't have a theory of malware. There are



textbooks on sorting and searching, on database methods, on computer graphics. These textbooks present algorithms and source code listings. The many different techniques of sorting, for example, are analyzed and their implementations are examined thoroughly. Students are encouraged to explore new approaches to sorting, to improve on what is known, to push the limits of performance. Whereas such explorations are standard practice in areas such as sorting, they do not exist for malware. Malware was absent from nearly all undergraduate curricula six years ago and it is still absent, for essentially the same technical and ideological reasons.

### Technical and Ideological Requirements

On the technical side, teaching malware requires knowing viruses, worms, Trojans, and rootkits, which obligates teachers to have read their source code, which in turn requires them to have the ability to reverse the binaries, and the facility to launch, run, and infect machines on an isolated subnet. Having read a sufficiently large, representative sampling of historic malware source code then leads to formulating various generalizations to build a theory of malware that can be tested by writing derivative malware, new in a shallow sense but not necessarily innovative. These experiences then should culminate in inventing never-before-tried malware to foresee trends in cyberspace.

On the ideological side, arguments range from “moral purity” to “allocation of responsibility.” These arguments are fueled by fear of the un-

**The reason we cannot solve the malware problem is simple: We don't have a theory of malware.**

**Detecting and arresting malware and its launchers won't be easy unless we ramp up on all fronts, especially education.**

known, especially when the unknown is potentially toxic. Having one's reputation ruined by being labeled irresponsible, negligent, reckless, or incompetent is a strong disincentive. It is difficult to imagine computer scientists losing their professional standing or community esteem by demonstrating new multi-core implementations of Batchers sort, especially if it beat all current sorting techniques; but it is not difficult to conjure the poisonous politics of unveiling new malware that would escape detection by all current commercial anti-malware products. Raising the stakes with powerful sorting algorithms is a laudable, honorable endeavor; casting a spell with powerful new malware is considered undignified per se.


That malware should be taught to computer science majors runs into a frequent and bothersome accusation—that we will be granting diplomas to hordes of malicious hackers, aiding and abetting greater misbehavior than is being suffered already. Physicians, surgeons, nurses, pharmacists, and other health professionals have the know-how with which to inflict pain, torture, and death. Every profession may have its “black sheep,” but it is obvious that society benefits by having an absolute majority of responsible and caring professionals.

### Conclusion

I began this column by calling your attention to the forthcoming triple trouble of cyberwar, cyberterrorism, and cybercrime. The last of the three—cybercrime—is abundantly in our midst

already. The other two menaces are works in progress. All three typically deploy via malware. (Human gullibility is, tragically, a contributing factor.) The preferred way thus far has been to exploit overlay networks or saturation-bomb regions of the Internet to build a broad-based infrastructure of illegally tenanted user machines and servers—a large botnet, responsive to peer-to-peer and command and control communications. Such a botnet's unwitting foot soldiers—your and my machines—are powerful weapons in cyberspace, capable of mounting targeted distributed denial-of-service attacks against individual users, institutions, corporations, and governments. Botnets built by worms can remain silent and undergo quiet maintenance and upkeep between bursts of activity. Botnet battles—territorial disputes and turf fights—are vicious confrontations for supremacy, worth billions of dollars and euros. For nation-states, the cyber-arms-race is on: those with the strongest malware will emerge as super-cyber-powers. None of these near-future developments can be wished away. And we continue to harm ourselves by not teaching malware.

May we let thousands of talented young minds lie fallow until our ignorant denial of the problem can no longer be condoned? How much malware damage should we tolerate? Until universal infection is the status quo? How are we to respond to massive but very likely covert malware pandemics? Would our response be capable of restoring and maintaining stability? More importantly, would we be able to verify the effectiveness of such a response?

Detecting and arresting malware and its launchers won't be easy unless we ramp up on all fronts, especially education. Millions of educated professionals are our best defense. Classrooms can be constructive idea generators. Let's not wait another six years for important ideas, such as malware prevention and preemptive interdiction, to be realized. 

**George Ledin, Jr.** ([george.ledin@sonoma.edu](mailto:george.ledin@sonoma.edu)) is a professor of computer science at Sonoma State University and a visiting fellow at SRI International.

Copyright held by author.



# Kode Vicious Forest for the Trees

*Keeping your source trees in order.*

## Dear KV,

I've noticed that you comment a great deal on the cleanliness of people's code, comments, version numbers, and other coding habits, but you've never mentioned one of my pet peeves: people who can't seem to name their source trees correctly. Don't people who tell you, "Oh, that file is in ~my-name/project-foobar" annoy you? I can't imagine that they don't.

### Frustrated by the Trees

## Dear Frustrated,

There are so many things that frustrate me—as these columns have pretty clearly indicated—and so, yes, you are correct. People who don't store their checkouts neatly and in some reasonable fashion annoy me.

I often think that many programmers see their checkouts as they saw their rooms as children: a private domain in which they could do as they pleased until a parent told them to clean things up. With the amount of disk space available to the modern programmer, and the lack of parental supervision in most workplaces, the time to "clean your room!" never comes. Thus, their checkouts grow and accrete files they call temporary but that really should have been given a good home, or removed, long ago.

What happens next is that you're in a meeting or talking with said programmer and you ask, "Hey, where's the source data that you made that graph from?" or "Did you check in that



useful script you wrote last month?" These people will invariably say, "Oh, I meant to, but it's just not that important. You can just go copy it from my tree. It's somewhere in my home directory under my-latest-work-17." "17" is their attempt at a version number, but don't expect them to have any directories labeled 1 to 16—really, just don't. Now you have to find the file, which you

get to do via the excellent, and often slow, `find(1)` command. Hopefully they remembered the name of the file or you'll get to do multiple searches, which is never fun. The only thing that makes this kind of sloppiness worse is when it is done completely in public, in open source projects.

Most, if not all, open source projects allow you to follow them by using one

of the current plethora of source-code control systems to check out their software to your local machine. While providing such a service is a great thing, providing it poorly is much like setting up a library in the middle of town, throwing all the books up in the air, letting them fall where they may, and then labeling some of them with Post-it Notes. Though most projects are not this horrific, I have noticed a tendency toward several sloppy, and therefore maddening, practices. I blame this trend on the recent introduction of distributed version-control systems, such as Mercurial and Git.

KV's first rule of public source-tree maintenance is to label everything clearly. Even if you don't think a tree will last very long, label it: give it a meaning that those who are new to your project can easily understand so they can figure out if that tree is, indeed, of interest to them.

My second rule is to not mix personal developer trees with release trees. A Web page with 100 different possible checkout targets—and you may laugh, but I see this on a regular basis—is not a good way to present your project to users; nor is it a good way to make code available. Keeping developer private source trees separate from trees you intend as real releases is a good way to increase sanity and reduce clutter. If people really need to check out a developer's private tree, they'll likely find it, though you might help them along by setting up a page labeled "Developer Trees."

And lastly, don't use developer trees as release trees. If the code in the developer's tree is good enough to make a release, then have the developer check it in, make a branch, and release it. A developer who is too lazy to do this should not be part of a project. No developer is important or brilliant enough for his or her tree to be the release tree.

**KV**

**Dear KV,**

In my spare time at work I've been adding an embedded language to some of our tools so that other people on my team could more easily script parts of their work. After spending a few weeks doing this, I showed what I had done to my team, and instead of them all

**If the code in the developer's tree is good enough to make a release, then have the developer check it in, make a branch, and release it.**

being happy and welcoming the extra work, their reactions ran from indifferent to hostile. I even used a popular, open source, embeddable language, not something I cooked up on my own. I made their jobs easier. Why wouldn't they be happy?

**Underappreciated**

**Dear Under,**

Are you sure you made their jobs easier? Are you sure you understand their jobs? It is a common belief by engineers that every piece of code they write is somehow a boon to mankind and is helping to drive the entire human race forward, propelling us all into a brave new world. Another thing to consider is that most people do not like surprises, even good ones. Try this experiment. Take a \$20 bill—or if you're in Europe a 10-euro note—and leap into a coworker's cubicle screaming, "Good morning!!!" at the top of your lungs and then loudly slap the bill on the desk. You've just given your coworker money, so surely that coworker will be happy to see you. Please report back your results.

What is more likely is that you found a need that you, yourself, wished to fill and you spent some enjoyable time filling that need. There *is* nothing wrong with working to scratch a technical itch; some of the best innovations come from engineers doing that. There is something wrong with believing that a group of people, who have no idea what you've been doing late at night for the past month, are suddenly going

to look at whatever you've created and say, "Oh, joy! It's just what I wanted!" All but the most obvious of creations need to be socialized. (Yes, I used *socialized* in a technical column.)

If you want your idea to be accepted, you first have to understand whether it is needed by anyone except yourself. Doing this by secretly watching your coworkers and taking notes is a great way to get yourself put on some sort of psych watch list with HR, so I suggest you go about it a bit less subtly than that: by asking them. Ask one or two people you think would want to use your new software if they are actually interested in what you're thinking about building. If they say, "No," that's not a reason to stop; it just tells you that when you're done, you'll have to do a lot more work to get them to see how great your creation is. Just for the record, yelling at them in a meeting and telling them how stupid they are not to see how clever you are is also a losing strategy.

Your best bet is to think about a simple part of your new system that is so useful, and so incontrovertibly a boon to their daily lives, that they will immediately find a use for it. Concentrate on making that useful piece available to them, and you will likely win them over. Or, you could just become management and force them all to do your bidding. Either way.

**KV**

**Q Related articles on [queue.acm.org](http://queue.acm.org)**

**Purpose-Built Languages**

Mike Shapiro

<http://queue.acm.org/detail.cfm?id=1508217>

**Broken Builds**

Kode Vicious

<http://queue.acm.org/detail.cfm?id=1740550>

**George V. Neville-Neil** ([kv@acm.org](mailto: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.





# Education From Science to Engineering

*Exploring the dual nature of computing education research.*

A SERIES OF recent reports claim the U.S. education system is in a very severe crisis; others suggest the crisis is “overblown.”

On the one hand, the National Academies released a report “Rising Above the Gathering Storm, Revisited: Rapidly Approaching Category 5,”<sup>6</sup> which argued that the U.S. economy is at risk because innovation will suffer due to poor-quality science education. The President’s Council of Advisors on Science and Technology (PCAST) stated in its report “Prepare and Inspire: K–12 Education in Science, Technology, Engineering, and Math (STEM) for America’s Future”<sup>8</sup> that there are “troubling signs” about U.S. STEM education. In particular, the Council of Advisors’ report called out the importance of knowing about computing, for example, they say “a basic understanding of technology and engineering is important if our children are to contribute to and compete in a rapidly changing society and an increasingly interconnected global community.”

On the other hand, an essay from Nicholas Lemann in a recent issue of *The New Yorker* referred to the crisis in American education as “overblown.”<sup>3</sup> Lemann points out that the American system of mass higher education is “one of the great achievements of American democracy.” In September, a *New York Times* article pointed to rising unemployment in the technology sector, suggesting that maybe we have *too many* computing graduates.<sup>7</sup>



President Barack Obama meeting with the President’s Council of Advisors on Science and Technology (PCAST).

All of these reports might be right. An explanation that satisfies all these claims is that we are educating large numbers of students, as Lemann suggests, but not well enough to address the needs described in the National Academies and PCAST reports. The unemployed technology workers described by the *New York Times* may not have the right skills or knowledge to get the jobs that will fuel innovation.

Computing education research has a role to play here. If these reports are

right, we need to produce more graduates with a higher level of knowledge and skill. Computing education research can help us figure out where the shortcomings are in the U.S. education system, and how to address them.

## The Sorry State of CS1

The introductory course in computer science in higher education is often referred to as “CS1” from early ACM and IEEE curriculum volumes. One of the first efforts to measure performance in



ACM's *interactions* magazine explores critical relationships between experiences, people, and technology, showcasing emerging innovations and industry leaders from around the world across important applications of design thinking and the broadening field of the interaction design. Our readers represent a growing community of practice that is of increasing and vital global importance.

**interactions**  
<http://www.acm.org/subscribe>



CS1 was in a series of studies by Elliot Soloway and his colleagues at Yale University. They regularly used the same problem, called "The Rainfall Problem": *Write a program that repeatedly reads in positive integers, until it reads the integer 99999. After seeing 99999, it should print out the average.* In one study, only 14% of students in Yale's CS1 could solve this problem correctly.<sup>9</sup> The Rainfall Problem has been used under test conditions and as a take-home programming assignment, and is typically graded so that syntax errors don't count, though adding a negative value or 99999 into the total is an automatic zero. Every study that I've seen (the latest in 2009) that has used the Rainfall Problem has found similar dismal performance, on a problem that seems amazingly simple.

Mike McCracken realized the problem with Soloway's studies, or any similar study, could be that a single campus could get it wrong. Maybe Yale just taught CS1 badly. McCracken wanted to find problems that students might be having *in general* with CS1. He organized a multi-institutional, multi-national (MIMN) study, with student data on the same problem collected from four institutions in three different countries.<sup>5</sup> One place might get it wrong, use the "wrong" language, or use "objects-first" when they ought to do "objects-later" (or some other pedagogical trade-off). Studying a range of schools helps us to describe "traditional" teaching of that subject, and student outcomes from that teaching. McCracken's group asked students to evaluate arithmetic expressions where

**Computing education research can help us figure out where the shortcomings are in the U.S. education system, and how to address them.**

the numbers and operations appeared in a text file (prefix, postfix, or infix—student's choice). 215 CS1 students participated in the study. The average score was 21%. Many of the participants never got past the design part of the problem to write any code at all.

Raymond Lister thought that maybe McCracken's study was asking students to do too much, in that they were designing solutions and implementing programs. He organized another MIMN study, this time where they asked students to read and trace code. 556 students from six institutions across seven countries completed 12 multiple-choice questions involving iteration and array manipulation. The average score was 60%; 23% of the students were only able to get four or fewer problems correct.

The most recent evaluation of CS1 is in the dissertation by Allison Elliott Tew, whom I advised.<sup>10</sup> Elliott Tew has been interested in how we can compare performance between different kinds of CS1, especially where the language varies. She hypothesized that students could take a test written in a pseudocode, especially designed to be easy to read, that would correlate well with how students performed in whatever their "native" CS1 language was. Before she created her test, though, she had to define what we mean by "CS1."

Since Elliott Tew wanted her test to be usable in a variety of different kinds of classes, she tried to define a small subset of what different people saw as "CS1 knowledge." Elliott Tew looked at popular CS1 textbooks to define the intersection set of topics between those, and used the ACM/IEEE curricular volumes to identify only those topics recommended for CS1. In the end, she defined a very small subset of what anyone teaches in CS1 as the "CS1 knowledge" that she would test.

Elliott Tew created an exam with 27 questions in each of MATLAB, Python, Java, and her pseudocode. Each of her 952 subjects, from three institutions in two countries, completed two exams: One in her pseudocode, and one in their "native" language. She found that the correlation was very high between the pseudocode and the "native" language, and additionally, the correlation was very high between the pseudocode and the students' final exam grade in CS1.

Elliott Tew's results make a strong case that pseudocode can be used effectively in a language-independent test for CS1 knowledge and that her test, in particular, is testing the same kinds of things that CS1 teachers are looking for on their final exams. But the relevant finding is that the majority of her 952 test-takers *failed* both of her exams, based on a small subset of what anyone teaches in CS1. The average score on the pseudocode exam was 33.78%, and 48.61% on the "native" language exam.<sup>10</sup>

These four studies<sup>4,5,8,10</sup> paint a picture of a nearly three-decades-long failure to teach CS1 to a level that we would expect. They span decades, a variety of languages, and different teaching methodologies, yet the outcomes are surprisingly similar. Certainly, the majority of students do pass CS1, but maybe they shouldn't be passing. Each of these four efforts to objectively measure student performance in CS1 has ended with the majority of students falling short of what we might reasonably consider passable performance. The last three studies,<sup>4,5,10</sup> in particular, have each attempted to define a smaller and smaller subset of what we might consider to be CS1 knowledge. Yet performance is still dismal. We have not yet found a small enough definition of CS1 learning outcomes such that the majority of students achieve them.

There are a lot of possible explanations for why students perform so badly on these measures. These studies may be flawed. Perhaps students are entering the courses unprepared for CS1. Perhaps our expectations for CS1 are simply too high, and we should not actually expect students to achieve those learning objectives after only a single course. Perhaps we just teach CS1 badly. I argue that, regardless of explanation, these four studies set us up for success.

### From Science to Engineering

In 1985, Halloun and Hestenes published a careful study of their use of the Mechanics Diagnostics Test, later updated as the Force Concept Inventory.<sup>2</sup> The Force Concept Inventory (FCI) gave physics education researchers a valid and reliable yardstick by which to compare different approaches to

## We need to develop better ways of teaching computer science, like the physics educators' interactive-engagement methods.

teaching physics knowledge about force. The traditional approach was clearly failing. Hestenes reported that while "nearly 80% of the students could state Newton's Third Law at the beginning of the course...FCI data showed that less than 15% of them fully understood it at the end."

FCI as a yardstick was the result of physics education research as *science*. Scientists define phenomena and develop instruments for measuring those phenomena. Like computer scientists, education researchers are both scientists and engineers. We not only aim to define and measure learning—we develop methods for changing and improving it.

Physics education researchers defined a set of new methods for teaching physics called *interactive-engagement methods*. These methods move away from traditional lecture, and instead focus on engaging students in working with the physics content. In a study with a stunning 6,000-plus participants interactive-engagement methods were clearly established to be superior to traditional methods for teaching physics.<sup>1</sup> Once the yardstick was created, it was possible to engineer new ways of teaching and compare them with the yardstick.

The demands of the "Rising Above the Gathering Storm" and PCAST reports call on computing education researchers to be engineers, informed by science. These four studies establish a set of measures for CS1 knowledge. There are likely flaws in these measures. More and better measures

can and should be developed. There is much that we do not know about how students learn introductory computing. There is plenty of need for more science.

But even with just these studies, we have significant results showing that our current practice does *not* measure up to our goals. We have four yardsticks to use for measuring progress toward those goals. Even doing better against these yardsticks would be improving our teaching methods of computer science. Our challenge is to do better. We need to develop better ways of teaching computer science, like the physics educators' interactive-engagement methods. We need to publish our better methods and demonstrate successful performance against common yardsticks.

Computing education researchers have proven themselves as scientists. The next challenge is to prove we can also be engineers. Computing education needs to build on the science and show measurable progress toward the needs identified by that science. **G**

### References

1. Hake, R.R. Interactive-engagement vs. traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics* 66 (1998), 64–74.
2. Hestenes, D. et al. Force Concept Inventory. *Physics Teacher* 30 (1992), 141–158.
3. Lemann, N. Schoolwork: The overblown crisis in American education. *The New Yorker* (Sept. 27, 2010); [http://www.newyorker.com/talk/comment/2010/09/27/100927taco\\_talk\\_lemann#ixzz10GexVQJU](http://www.newyorker.com/talk/comment/2010/09/27/100927taco_talk_lemann#ixzz10GexVQJU).
4. Lister, R. et al. A multi-national study of reading and tracing skills in novice programmers. Working group reports from ITiCSE Conference on Innovation and Technology in Computer Science Education. ACM, Leeds, U.K., 2004, 119–150.
5. McCracken, M. et al. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *SIGCSE Bulletin* 33, 4 (2001), 125–180.
6. Prepare and Inspire: K–12 Education in Science, Technology, Engineering, and Math (STEM) for America's Future. Executive Office of the President, Washington, D.C., 2010.
7. Rampell, C. Once a dynamo, the tech sector is slow to hire. *The New York Times* (Sept. 7, 2010); <http://www.nytimes.com/2010/09/07/business/economy/07jobs.html>.
8. Rising Above the Gathering Storm, Revisited: Rapidly Approaching Category 5. National Academies Press, Washington, D.C., 2010.
9. Soloway, E. Cognitive strategies and looping constructs: An empirical study. *Commun. ACM* 26, 11 (Nov. 1983), 853–860.
10. Tew, A.E. Assessing fundamental introductory computing concept knowledge in a language independent manner. Ph.D. in Computer Science: School of Interactive Computing. Georgia Institute of Technology. Atlanta, GA (2010).

**Mark Guzdial** (guzdial@cc.gatech.edu) is a professor in the College of Computing at Georgia Institute of Technology in Atlanta, GA.

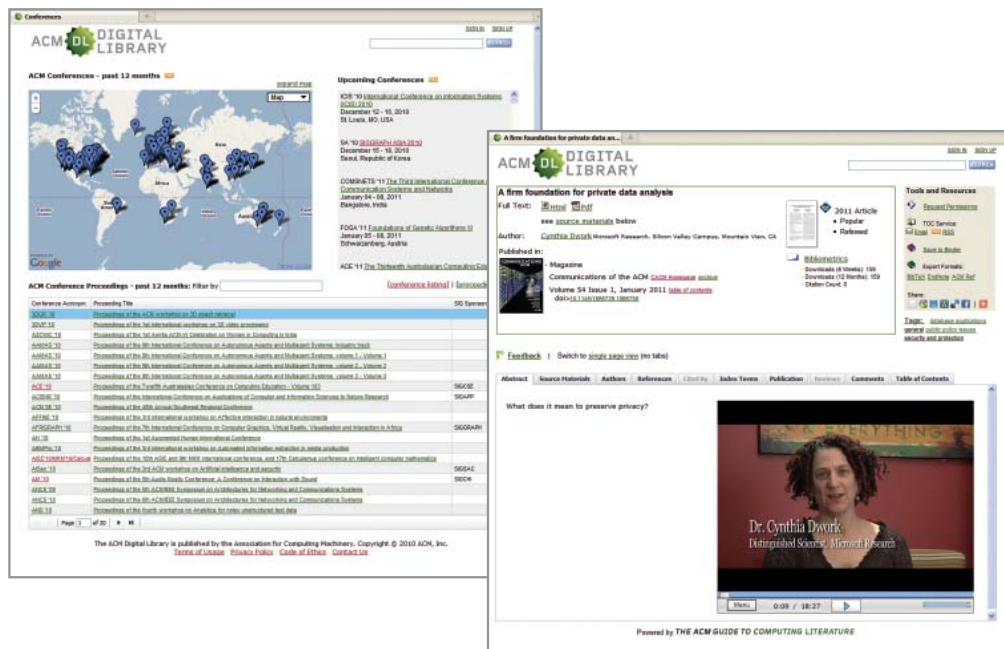
Copyright held by author.



# ACM LAUNCHES ENHANCED DIGITAL LIBRARY



The new DL simplifies usability,  
extends connections, and expands content.



- Broadened citation pages
- Redesigned binders
- Expanded table-of-contents
- Embedded video
- Integrated visualization technology
- Enhanced interactivity tools

Visit the ACM Digital Library at: [dl.acm.org](http://dl.acm.org)

Not a DL Subscriber yet?

Register for a free 3 month personal subscription at:

[dl.acm.org/free3](http://dl.acm.org/free3)

## Viewpoint

# Technology, Conferences, and Community

*Considering the impact and implications of changes in scholarly communication.*

**I**N 2009 AND 2010, over a dozen *Communications* Editor's Letters, Viewpoints, blog entries, reader letters, and articles addressed our conference and journal publication culture. The discussion covered the shift from a traditional emphasis on journals to the current focus on conferences, and the challenges of conference reviewing, but at its heart is our sense of community.<sup>2</sup> One commentary spoke of a "death spiral" of diminishing participation.<sup>1</sup> Several of the contributors joined a plenary panel on peer review at the 2010 Computing Research Association Conference at Snowbird.<sup>4</sup>

In a nutshell, the commentaries note that a focus on conference publication has led to deadline-driven short-term research at the expense of journal publication, a reviewing burden that can drive off prominent researchers, and high rejection rates that favor cautious incremental results over innovative work. Some commentators identify novel approaches to addressing these or other problems, but the dominant view is that we should return to our past practice of regarding journal publication as the locus of quality, which remains the norm in other sciences.

To understand whether this is possible, and I doubt it is, we must understand why computer science in the U.S. shifted to conference publication in the first place. As commentators have noted, it was not simply that computer science requires quick dissemination of results: Conferences did not become focused on quality in Europe or Asia, or

in other competitive, quickly evolving fields such as neuroscience or physics. It is not that U.S. computer science could not expand journal page counts: computer science journals abroad expanded, passing the costs on to libraries. This Viewpoint considers other factors and outlines their implications.

### Technology and a Professional Organization Drove the Shift to Conference Publication

By the early 1980s, the availability of text editing or word processing among computer scientists enabled the relatively inexpensive production of decent-looking proceedings prior to a conference. This was something new. Anticipating that libraries might shelve proceedings, ACM printed many more copies than conferences needed, at a low incremental cost.

ACM also made them available by mail order after a conference at a very low price. Papers in ACM conferences were thus widely distributed and effectively archival. *These are the two features that motivated the creation of journals centuries earlier.*

Proceedings in Europe and Asia rarely had after-conference distribution, so to be archived, work there had to progress to journal publication. The shift to a conference focus did not occur. In 2004, a prominent U.K. researcher wrote about the CHI conference: "*HCI's love of conferences is a fluke of history. We all know this. CS in general has suffered from it, but is steadily moving away. CHI however digs in, with more and more death rattles such as CHI Letters. Being conference centered is bad for any field: bad for its archival material, bad for its conferences,*

**Membership in the top 10 ACM Special Interest Groups in 1990, 2000, and 2010. Currently, only two of 34 SIGs have 3,000 members.**

SIG	1990	2000	2010
PLAN	12,335	4,362	2,323
GRAPH	11,811	4,063	7,216
SOFT	10,824	3,313	2,489
ART	8,955	1,917	1,123
OPS	6,801	2,356	1,828
CHI	5,023	4,950	4,415
ARCH	4,685	1,730	1,384
ADA	4,089	685	292
MOD	3,952	2,941	1,922
MIS	3,082	755	497
(all 30+)	103,489	47,042	41,008

*and worst of all, really bad for the respect that we command with other communities. SIGCHI needs to move away from bolstering up conference publications. It needs to use journals for journal stuff and conferences for conference stuff.”<sup>a</sup>*

He was wrong about the direction of computer science, and at least premature in diagnosing CHI’s expiration. The point, though, is that he saw the problem as an American problem, affecting CHI but not European HCI.

### Knock-on Effects

This change in the complex ecology of scholarly communication was followed by a slow sequence of adjustments. ACM and IEEE had considered conference papers to be ephemeral, and expressly allowed verbatim or minimally revised republication in journals and transactions. With proceedings effectively archived even before digital libraries arrived, this policy was formally ended early in the 1990s.

A significant consequence is that it is increasingly difficult to evolve conference papers into journal articles. Publishers, editors, and reviewers expect considerable new work, even new data, to avoid a charge of self-plagiarism. Republishing the same work is undesirable, but we have inhibited the use of review and revision cycles to clean up conference papers, expand their literature reviews, and engage in the deeper discussions that some feel are being lost.

The pattern extends beyond systems. I edited *ACM Transactions on Computer-Human Interaction* and serve on the editorial boards of *Human-Computer Interaction*, *Interacting with Computers*, and *ACM Computing Surveys*. By my estimation, no more than 15% of the work published in highly selective HCI conferences later appears in journals. Journal publication is not a prerequisite for being hired into leading research universities. Today, the major U.S. HCI journals mostly publish work from Europe and Asia, where conferences are less central.

Now let’s consider reviewing, a primary focus of discussion, before turning to the impact of these changes on our sense of community.

## When conferences became archival, it was natural to focus on quality and selectivity.

### Conference Selectivity and Effects on Reviewing

In other fields, journals focus on identifying and improving research quality; large conferences focus on community building and community maintenance; and workshops or small conferences focus on member support through specialist discussions of work in progress. This reflects Joseph McGrath’s division of group activities into those focused on production, team health, and member support.<sup>3</sup>

When conferences became archival, it was natural to focus on quality and selectivity. Even with authors preparing camera-ready copy, the expense of producing a proceedings was proportional to its page count. Libraries sales were a goal prior to the emergence of digital libraries in the late 1990s. Libraries were more likely to shelve thinner proceedings, and needed to be convinced the work had lasting value. These pressures drove down conference acceptance rates. In my field they dropped from almost 50% to 15% before settling in a range, 20%–25%, that is acceptably selective to academic colleagues yet not brutally discouraging to authors, we hope.

But it is discouraging to have submissions rejected. I know few if any people who submit with no hope of acceptance. In most fields, conferences accept work in progress. It is also discouraging when we see a paper presented and immortalized in the digital library that seems less worthy than a paper that was rejected. Review processes are noisy, and more so as the reviewer pool expands to include graduate students and others. Multidisciplinary fields, with diverse methodologies and priorities, deliver especially random outcomes.

Previous commentaries emphasized

that caution and incrementalism fare better than innovation and significance in conference assessments. An incremental advance has a methodology, a literature review, and a rationale for publication that were bulletproofed in the papers it builds on. We try to channel papers to the most expert reviewers in an area, but to them incremental advances loom larger than they will to others. With pressure to reject ~75% and differing views of what constitutes significant work, the minor flaws or literature omissions that inevitably accompany novel work become grounds for exclusion. And in a zero-sum game where conference publication leads to academic advancement, a novel paper can be a competitive threat to people and paradigms, echoing concerns about journal conservatism in other fields.

Birman and Schneider describe the risk of a “death spiral” when senior people cease to review.<sup>1</sup> Although engaging young researchers as reviewers is great, they often feel more comfortable identifying minor flaws and less comfortable in declaring that work is more or less important. Every year, conferences in my area adjust the review process. But significant change is elusive, given the forces I have described.

### Impact on Community

A leading neuroscientist friend described the profession’s annual meeting as a “must-attend” event “where people find out what is going on.” There are 15,000 presentations and 30,000 attendees. The quality bar is low. It is a community-building effort in a journal-oriented field.

In contrast, despite tremendous growth in many CS specializations, attendance at many of our conferences peaked or plateaued long ago. So has SIG membership, as shown in the accompanying table. Conferences proliferate, dispersing organizational effort and the literature, reducing a sense of larger community.

In my field, CHI once had many vibrant communication channels—a highly regarded newsletter, an interactive email discussion list, passionate debates in the halls and business meetings of conferences, discussants for paper sessions, and in the late 1990s an active Web forum. All of them disap-

<sup>a</sup> Gilbert Cockton, email communication, Jan. 22, 2004.



peared. The CHI conference gets more submissions, but attendance peaked years ago. When a small, relatively polished subset of work is accepted, what is there to confer about?

High rejection rates undermine community in several ways. People don't retain quite the same warm feeling when their work is rejected. Without a paper to give, some do not receive funding to attend. Rejected work is revised and submitted to other conferences, feeding conference proliferation, diverting travel funding, and dispersing volunteer efforts in conference management and reviewing. In addition, high selectivity makes it difficult for people in related fields to break in—especially researchers from journal-oriented fields or countries who are not used to polishing conference submissions to our level.

A further consequence is that computer scientists do not develop the skills needed to navigate large, community-building conferences. At our conferences, paper quality is relatively uniform and the number of parallel sessions small, so we can quickly choose what to attend. In contrast, randomly sampling sessions at a huge conference with 80% acceptance leads us to conclude that it is a junk conference. Yet with a couple hours of preparation, combing the many parallel sessions for topics of particular interest, speakers of recognized esteem, and best paper nominations, and then planning meetings during some sessions, one can easily have as good an experience as at a selective conference. But it took me a few tries to discover this.

Courtesy of Moore's Law, our field enjoys a constant flow of novelty. If existing venues do not rapidly shift to accommodate new directions, other outlets will appear. Discontinuities can be abrupt. Our premier conference for many years, the National Computer Conference, collapsed suddenly two decades ago, bringing down the American Federation of Information Processing Societies (AFIPS), then the parent organization of ACM and IEEE. Over half of all ACM A.M. Turing Award winners published in the AFIPS conferences. Most of those published single-authored papers. Yet the AFIPS conference proceedings disappeared, until they were recently added to the ACM Digital Library. The field moved on—renewal is part of our heritage. But perhaps we can smooth the process.

## Possible Directions

Having turned our conferences into journals, we must find new ways to strengthen community. Rolling back the clock to the good old heyday of journals, ignoring changes wrought by technology and time, seems unlikely to happen. For one thing, it would undermine careers built on conference publication. More to the point, computer science in the U.S. responded first to technologies that enable broad dissemination and archiving. Other countries are now following; other disciplines will also adapt, one way or another. Instead of looking back, we can develop new processes and technologies to address challenges that emerged from exploiting the technologies of the 1980s.

With storage costs evaporating, we could separate quality determination from participation by accepting most conference submissions for presentation and online access, while distinguishing ~25% as "Best Paper Nominations." Making a major conference more inclusive could pull participation back from spin-off conferences.

A more radical possibility is inspired by the revision history and discussion pages of Wikipedia articles. Authors could maintain the history of a project as it progresses through workshop, conference, and journal or other higher-level accreditation processes. Challenges would have to be overcome, but such an approach might ameliorate reviewer load and multiple publication burdens—or might not.

We are probably not approaching the bottom of a "death spiral." But when AFIPS and the National Computer Conference collapsed, the transition from profitable success to catastrophe was remarkably rapid. Let's continue this discussion and keep history from repeating. **C**

## References

1. Birman, K. and Schneider, F.B. Program committee overload in systems. *Commun. ACM* 52, 5 (May 2009), 34–37.
2. Fortnow, L. Time for computer science to grow up. *Commun. ACM* 52, 8 (Aug. 2009), 33–35.
3. McGrath, J.E. Time, interaction, and performance (TIP): A theory of groups. *Small Group Research* 22, 2 (1991), 147–174.
4. Peer review in computing research. CRA Conference at Snowbird, July 19, 2010; <http://www.cra.org/events/snowbird-2010/>.

**Jonathan Grudin** ([jgrudin@microsoft.com](mailto:jgrudin@microsoft.com)) is a member of the Adaptive Systems and Interaction Group at Microsoft Research in Redmond, WA.

Copyright held by author.

# Calendar of Events

## February 15–17

9<sup>th</sup> USENIX Conference on File and Storage Technologies, San Jose, CA, Contact: John Wilkes, Email: [john@e-wilkes.com](mailto:john@e-wilkes.com)

## February 18–20

Symposium on Interactive 3D Graphics and Games, San Francisco, CA, Sponsored: SIGGRAPH, Contact: Michael Garland, Email: [mjgarland@gmail.com](mailto:mjgarland@gmail.com)

## February 21–23

First ACM Conference on Data and Application Security and Privacy, San Antonio, TX, Sponsored: SIGSAC, Contact: Ravinderpal S. Sandhu, Email: [ulfar@yahoo.com](mailto:ulfar@yahoo.com)

## February 23–25

Multimedia Systems Conference, Santa Clara, CA, Sponsored: SIGMM, Contact: Wu-Chi Feng, Email: [wuchi@cs.pdx.edu](mailto:wuchi@cs.pdx.edu)

## February 24–27

Indian Software Engineering Conference, Thiruvananthapuram, Kerala India, Contact: Arun G. Bahulkar, Email: [arun.bahulkar@tcs.com](mailto:arun.bahulkar@tcs.com)

## February 25–26

International Conference & Workshop on Emerging Trends in Technology, Mumbai, Maharashtra India, Contact: Poorva G. Waingankar, Email: [poorva.waingankar@thakureducation.org](mailto:poorva.waingankar@thakureducation.org)

## March 9–12

The 42<sup>nd</sup> ACM Technical Symposium on Computer Science Education, Dallas, TX, Sponsored: SIGCSE, Contact: Thomas Contina, Email: [tcontina@cs.cmu.edu](mailto:tcontina@cs.cmu.edu)

## March 14–16

Second Joint WOSP/SIPEW International Conference on Performance Engineering, Karlsruhe, Germany, Sponsored: SIGMETRICS, SIGSOFT, Contact: Samuel Kounev, Email: [skounev@ipd.uni-karlsruhe.de](mailto:skounev@ipd.uni-karlsruhe.de)

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

## Automated usability tests can be valuable companions to in-person tests.

BY JULIAN HARTY

# Finding Usability Bugs with Automated Tests

IDEALLY, ALL SOFTWARE should be easy to use and accessible for a wide range of people. However, even software that appears to be modern and intuitive often falls short of the most basic usability and accessibility goals. Why does this happen? One reason is that sometimes our designs *look* appealing so we skip the step of testing their usability and accessibility—all in the interest of speed, reducing costs, and competitive advantage.

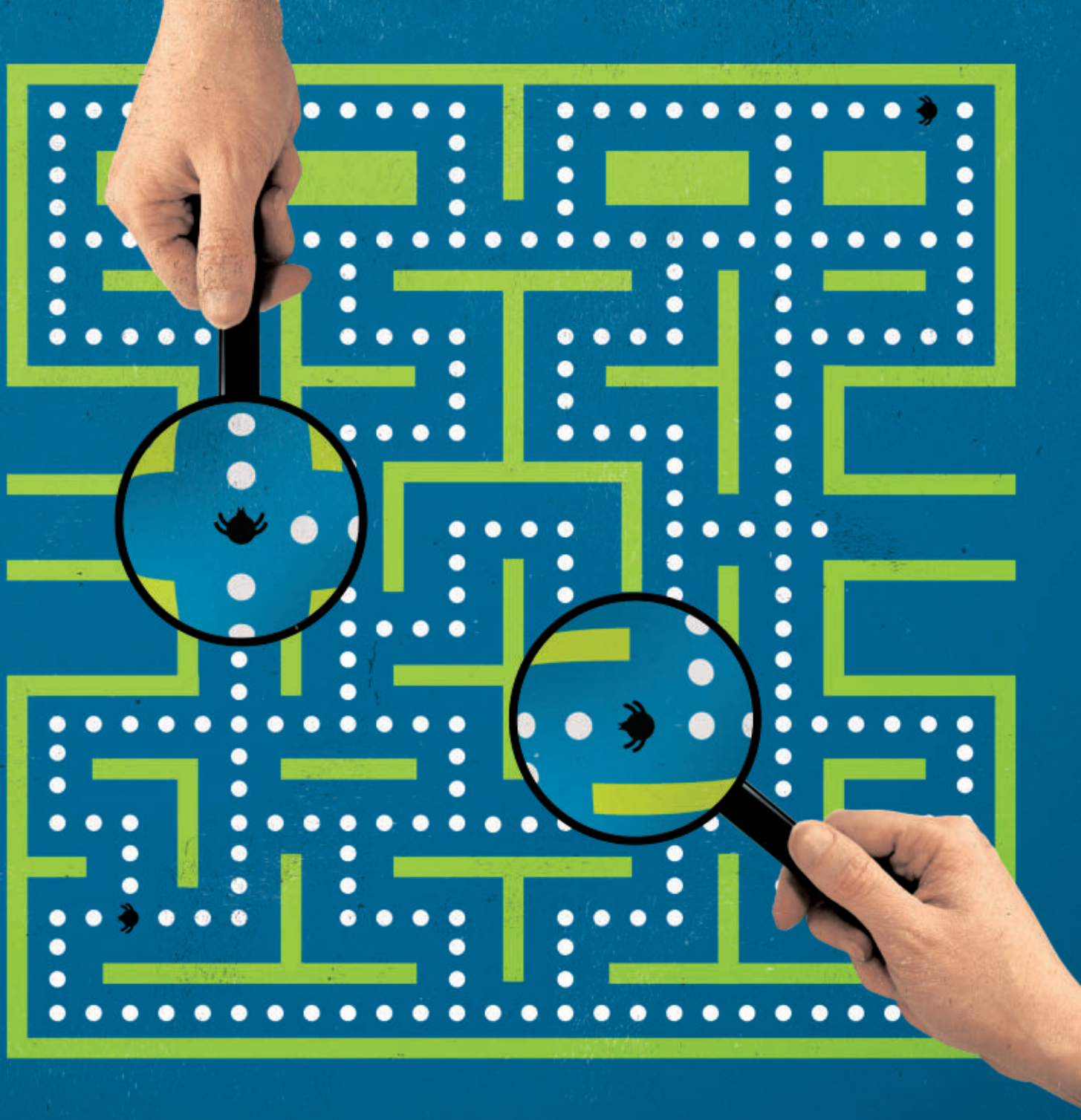
Even many large-scale applications from Internet companies present fundamental hurdles for some groups of users, and smaller sites are no better. We therefore need ways to help us discover these

usability and accessibility problems efficiently and effectively.

Usability and accessibility are two ways of measuring software quality. This article covers several ways in which automated tests can help identify problems and limitations in Web-based applications, where fixing them makes the software more usable and/or accessible. The work complements, rather than replaces, other human usability testing. No matter how valuable in-person testing is, effective automation is able to increase the value of overall testing by extending its reach and range. Automated tests that are run with minimal human intervention across a vast set of Web pages would be impractical to conduct in person. Conversely, people are capable of spotting many issues that are difficult to program a computer to detect.

Many organizations do not do any usability or accessibility testing at all; often it's seen as too expensive, too specialized, or something to address after testing all the "functionality" (which is seldom completed because of time and other resource constraints). For these organizations, good test automation can help in several ways. Automated tests can guide and inform the software development process by providing information about the software as it is being written. This testing helps the creators of the software fix problems quickly (because they have fast, visible feedback) and to experiment with greater confidence. It can also help identify potential issues in the various internal releases by assessing each release quickly and consistently.

Some usability experts find the idea of incorporating automated tests into their work alien, uncomfortable, or even unnecessary. Some may already be using static analysis tools such as Hera and Bobby to check for compliance with WCAG (Web Content Accessibility Guidelines; <http://www.w3.org/TR/WCAG20/>) and Section 508 (<http://www.access-board.gov/sec508/guide/1194.22.htm>), but



not yet using dynamic test automation tools. As a result, they catch some problems but miss others (which was the case for several of the examples given later in this article).

One aim of this article is to encourage readers simply to *try* applying some automated tests to see if they help uncover issues that may be worth fixing.

### **Why is Usability and Accessibility Testing Difficult?**

It's clear that companies today are not

doing enough usability and accessibility testing, and part of the reason is that it can be difficult to accomplish. Here are a few reasons why.

It's often difficult to understand users' frustrations when interacting with software, especially when their needs differ from ours. For example, if I'm a 20-something with good eyesight and mobility, an immensely detailed multilayered user interface might suit me well. But what about users with different abilities? Will they find it frustrating or possibly even

unusable? Finding ways to accommodate a wide range of users is challenging, particularly for corporations that have problems accepting offers of free help from groups such as blind users and people with motor impairments. Although rejecting such help may seem illogical, the logistics of preparation, transport, adapting the office environment to accommodate the visitors, and so forth, may discourage those who already have lots of demands on their limited time and resources.



There are a range of devices that bridge the gap between user and application—from mice and keyboards, to screen readers, to specialized equipment that adapts the user interface for people with severe impairments. Unless we have had experience with these tools, it is difficult to conceive of how they work in practice and how they affect the user's experience with *our* application. Furthermore, developers of UI tools seldom provide functional interfaces to support test automation or screen readers—making both challenging to implement.

In my view, usability testing is not inherently difficult, but it tends to be time consuming and hard to scale when it requires human observation of the people using the software being assessed. Also, because software developers seem to find the work fiddly or extraneous, they may fail at the first hurdle: deciding whether the testing effort is worth the investment.

### Difficulties in Test Automation

In addition to the basic challenges in usability and accessibility testing, there are also challenges in developing good automated testing frameworks. Although pockets of interesting academic research exist that focus on automated testing, as an industry we've found it difficult to translate

academic work into practical value. Sharing of knowledge and tools from academia has been limited, as companies need first to pay for access to many of the papers and then translate the formal structure of those papers into something that makes sense to them. These solutions must also address the expectations implicit in the question, "Will this solve some of my immediate issues today?" Complicating things is that commercial automated-testing tool providers tend to guard their tests and methods, as they perceive this to be to their competitive advantage.

Many test-automation frameworks are not used regularly, to the extent that practitioners actually run the automated tests again and again. Furthermore, in my experience many aren't even used by their author(s). They are written once—perhaps so the authors get good performance reviews from their managers, rather than providing practical value to their projects—and then fall into disuse, as no one sees the value in the work or wants to support it. Longer-term test automation tends to suffer from fragility, poor maintainability, and inadequate software engineering practices. Good test automation is like good software development, requiring similar skills, practices, and pas-

sion to create and maintain it.

Another difficulty in test automation is finding bugs that "bug" people to the extent they are deemed worth fixing versus bugs that will be discounted because users are unlikely to encounter them or because developers don't see the value of fixing them.

Still another challenge is integrating all the various Web browsers with the test-automation software. Each browser is distinct and needs special code so it can be used with automated tests. This code can be fairly complex and challenging to write. Developers have started various projects using a single browser only to discover that the overhead of trying to extend their work to additional browsers is significantly more complex and time consuming than they are prepared for.

Finally, many test-automation tools still require their users to have technical and programming skills (for example, Java, Maven, JUnit, IDEs, among others) to write the tests. For open source projects the initial learning curve may be too steep to get the software to run on your computer. Some companies try to dumb down the test automation so people without a programming background can write tests, but these attempts often cause more harm than good.

### Examples of Automated Testing

In 2009, I helped test several global software applications at Google. They were constructed using GWT (Google Web Toolkit), a very powerful development framework that allows developers to create cross-browser Web applications and hides many of the complexities from the developers. The resulting applications looked sleek and modern, and each had a user base of millions of people.

Development teams estimated GWT saved person-years of work in getting their applications into production. The way in which the teams were using GWT, however, resulted in several side effects that ended up generating usability and accessibility problems, such as broken keyboard navigation and poor support for screen readers. We discovered similar issues for other applications that used other frameworks and approaches, indicating these problems

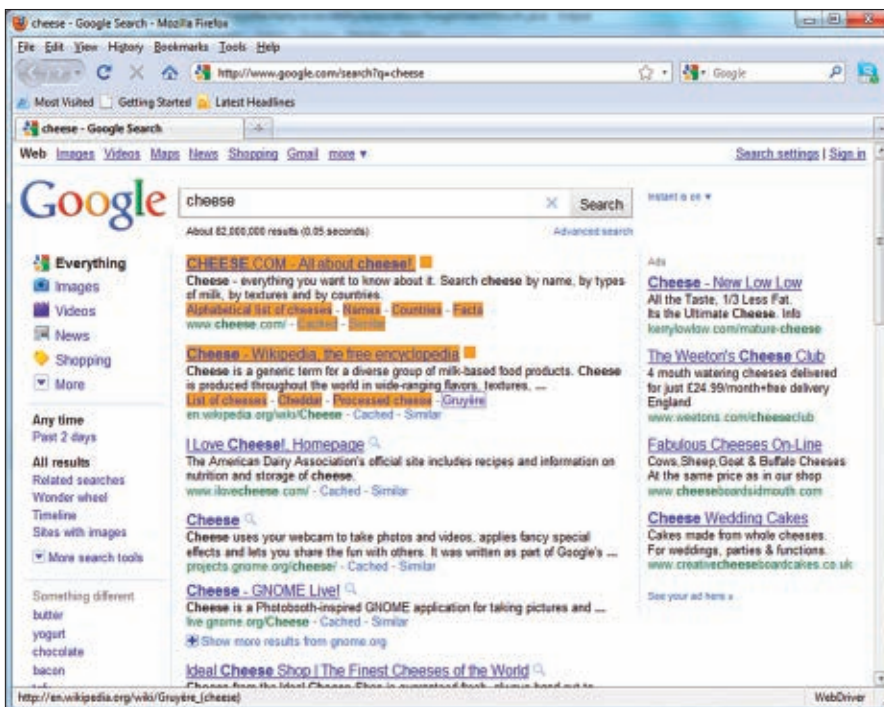


Figure 1. Screenshot of Google Search keyboard navigation test.

might be widespread and prevalent throughout the industry.

My main goal was to determine whether we could create automated tests that would help identify potential problems that may affect *quality-in-use* for groups of users in terms of dynamic use of the software. As mentioned earlier in this article, several standards (for example, Section 508) and guidelines (for example, WCAG) aim to help address basic problems with accessibility, and a plethora of software tools are available to test for Section 508 and WCAG compliance. None, however, seemed to focus on quality-in-use of the applications.


Furthermore, my work needed to provide positive ROI (return on investment), as well as be practical and useful.

### Testing Keyboard Navigation


One facet of usability and accessibility testing is keyboard input and navigation (as opposed to relying on a mouse or a touch screen). I decided to focus on finding ways to test keyboard navigation using automated software tools. The work started with a simple but effective heuristic: when we tab through a user interface, we should eventually return to where we started—typically, either the address bar in the Web browser or the input field that had the initial focus (for example, the search box for Google's Web search).

The initial test consisted of about 50 lines of Java code. It provided a highly visible indicator of the navigation by setting the background of each visited element to orange; each element was also assigned an ascending number representing the number of tabs required to reach that point. The screenshot in Figure 1 shows an example of navigating through the Google Search results. The tab order first works through the main search results; next, it tabs through the ads on the right, and then the column on the left; the final element is the Advanced Search link, which is arrived at after approximately 130 tabs! The code tracks the number of tabs, and if they exceed a specified value, the test fails; this prevents the test from running indefinitely.

This test helped highlight several



**One aim of this article is to encourage readers simply to try applying some automated tests to see if they help uncover issues that may be worth fixing.**



key issues such as *black holes*, Web elements that “swallow” all keystrokes. It also helped identify Web elements that were unreachable by tabbing through the page. Our success was measured by the percentage of bugs fixed and the reduction in keystrokes needed to navigate a user interface.

Several GWT frameworks include custom elements such as buttons. Developers can attach a JavaScript handler to these elements to handle specific keystrokes. We discovered a serious bug whereby JavaScript mistakenly discarded all the other keystrokes (apart from the specific ones it was designed to handle). This meant that once a user navigated to that button, he or she was unable to leave using the keyboard (tab characters were being silently discarded).

We used a heuristic in the automated test that assumed that if a user pressed the Tab key enough times, the user should eventually return to where he or she started in the user interface. The test included a “maximum number of tabs” parameter. We set this to three times the number of Web elements on the Web page as a balance between making sure we didn't “run out” of tabs before reaching the end of a legitimate page and the test continuing forever if we didn't cap the number of keystrokes. If the test failed to return to the initial element, it failed. This test was able to detect the black hole caused by the erroneous JavaScript. The problem was finally fixed by changing the code in the underlying custom GWT framework.

The second problem we discovered was a “new message” button that was unreachable using the keyboard. This was embarrassing for the development team, as they prided themselves on developing a “power-user” interface for their novel application. One aspect of the test was that it set the background color of each Web element it visited to orange. We were able to spot the problem by watching the tests running interactively and seeing that the “new message” button was never highlighted. We were able to spot similar problems by looking at screenshots saved by the test automation code (which saved both an image of the page and the DOM (document object model) so we could visualize

the underlying HTML content).

The third problem was more insidious and initially more difficult to detect. GWT used a hidden IFRAME in the Web page to store the history of Web pages visited by the user (so the user could navigate with the browser navigation controls such as “Back”). We discovered, however, that one of the initial tab characters was directed to the hidden IFRAME. This was confusing for users, because the cursor disappeared, and it was also mildly annoying, as they had to press an additional Tab character to get to where they wanted in the user interface. Once the problem was uncovered, the fix was easy: add a TABINDEX=“-1” attribute to the hidden IFRAME.

The next heuristic we considered was that the sum of the number of tabs should be identical for both forward (Tab) and reverse (Shift+Tab) keystrokes. The first part of the test used the same code as that used for the initial heuristic, where the count of tabs issued is incremented for each element visited. Once the test reached

the initially selected element, it started generating the Shift+Tab keyboard combination, which caused the navigation to go in reverse. Again, the number of Shift+Tab keystrokes was counted. Each time an element was visited, the value set in the title property of that element was added to the current value of the counter. The sum of the tab-orders should be identical for every element visited. If not, there is a hysteresis loop in the user interface indicating a potential issue worth investigating. Figures 2 and 3 show the tab counts for each element visited. We can see that each pair of values for a given Web element add up to nine (for example, Button B’s counts are:  $4 + 5 = 9$ ; and Button A’s counts are  $6 + 3 = 9$ ). So this test passes. [Note: the figures do not include extra tabs required for the browser’s address bar, etc.]

The final heuristic was that the flow of navigation should match a regular pattern such as down a logical column of input fields and then right and up to the top of the next column.

Figure 4 shows two typical flows.

Here we can detect whether the expected pattern is being followed by obtaining the  $(x,y)$  location of each element on the Web page. The pattern may be explicit (if we know what we want or expect) or implicit (for example, based on how similar Web pages behave). A tolerance may be used to allow slight variations in alignment (where we consider these to be acceptable).

Our automated tests rely on WebDriver, now part of the open source Selenium test-automation project and known as Selenium 2.0. WebDriver aims to interact with a given Web browser as a person would; for example, keystrokes and mouse clicks are generated at the operating-system level rather than being synthesized in the Web browser. We describe this as generating native events. Sometimes WebDriver cannot generate native events because of the technical limitations of a particular operating system or Web browser, and it must compensate by using alternative in-

Figure 2. Testing for forward tab counts.

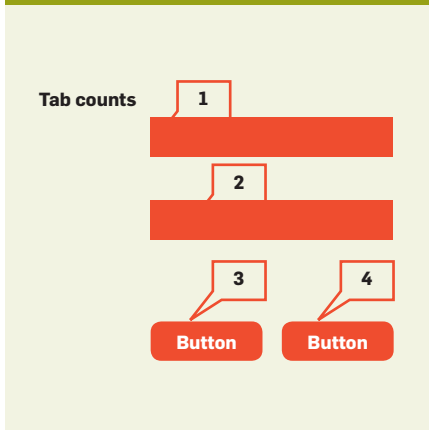


Figure 3. Testing for reverse tab counts.

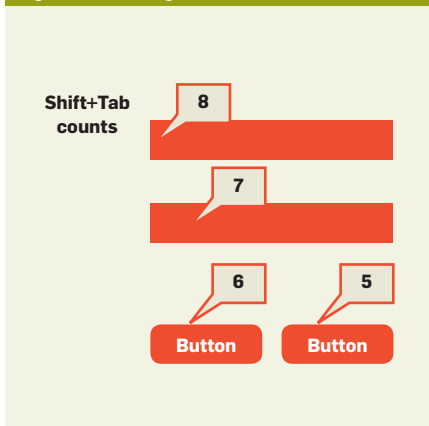
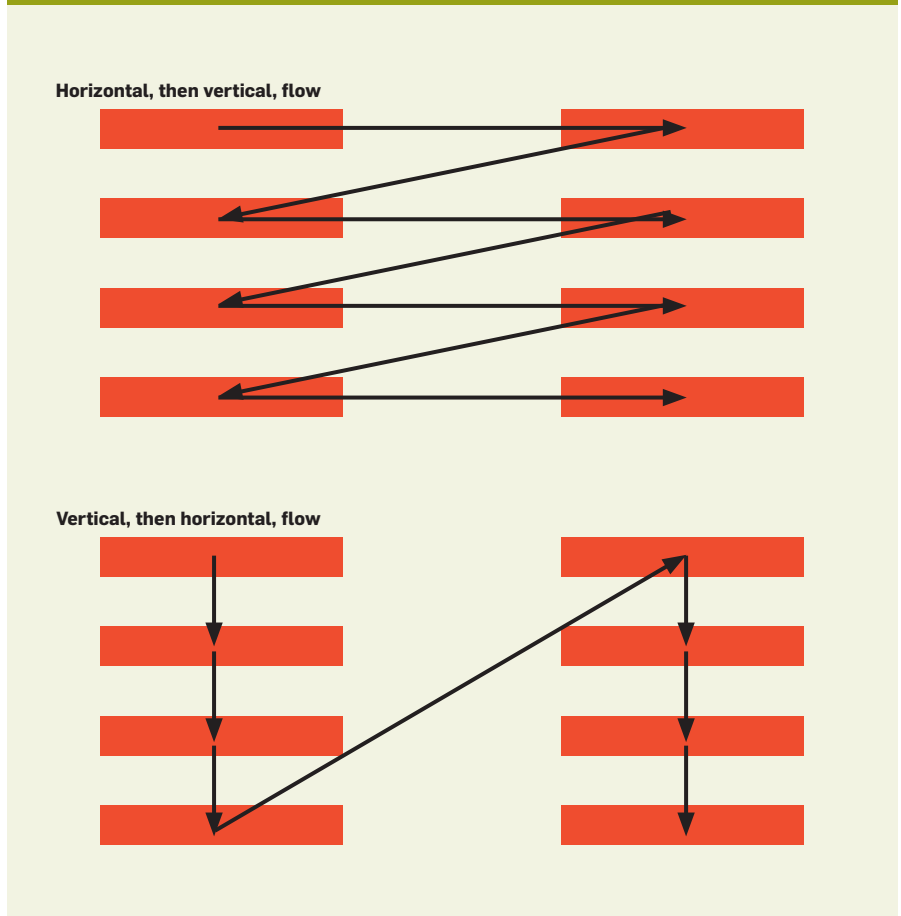


Figure 4. Typical navigation flows.





put methods. For the keyboard navigation tests, though, generating native events is essential to establishing the fidelity of the tests.

WebDriver works with the majority of popular desktop Web browsers, such as Firefox, Internet Explorer, Opera, and even includes the Web browsers on Android, iPhone, and BlackBerry devices. This broad reach means we can run our tests on the most popular browsers, which helps increase the usefulness of the tests.

### Finding Layout Issues

Layout problems are an area that can adversely affect a user's perception of an application and may indirectly reduce its usability by distracting or frustrating users. There are numerous classes of problems that can cause poor layout, including quirks in a particular Web browser, mistakes made by the developers and designers, and poor tools and libraries. Localizing an application from English to languages such as German, where the text is typically more voluminous, is a reliable trigger for some layout issues. Many of these problems have been challenging to detect automatically, and traditionally we have relied on humans to spot and report them.

This changed in 2009 when I met Michael Tamm, who created an innovative approach that enables several types of layout bugs to be detected automatically and simply. For example, one of his tests programmatically toggles the color of the text on a page to white and then black, taking a screenshot in both cases. The difference between the two images is generated, which helps identify the text on the page. Various algorithms then detect the horizontal and vertical edges on the Web page, which typically represent elements such as text boxes and input fields. The difference of the text is then effectively superimposed on the pattern of edges to see if the text meets, or even overlaps, the edges. If so, there is a potential usability issue worth further investigation. The tests capture and annotate screenshots; this allows someone to review the potential issues quickly and decide if they are serious.

For existing tests written in WebDriver, the layout tests were enabled

by adding a couple of lines of source code. For new automated tests, some code needs to be written to navigate to the Web page to be tested before running the tests. (See <http://code.google.com/p/fighting-layout-bugs> for more information, including a video of Tamm explaining his work, sample code, and so on.)

### Looking Ahead

Our work to date has been useful, and I expect to continue implementing test automation to support additional heuristics related to dynamic aspects of Web applications. WebDriver includes support for touch events and for testing on popular mobile phone platforms such as iPhone, Android, and BlackBerry. WebDriver is likely to need some additional work to support the matrix of tests across the various mobile platforms, particularly as they are frequently updated.

We are also considering writing our tests to run interactively in Web browsers; in 2009, a colleague created a proof of concept for Google's Chrome browser. This work would reduce the burden of technical knowledge to run the tests. The final area of interest is to add tests for WAI-ARIA (Web Accessibility Initiative-Accessible Rich Internet Applications; <http://www.w3.org/WAI/intro/aria.php>) and for the tests described at <http://openajax-dev.jongund.webfactional.com/rules/>.

We are actively encouraging sharing of knowledge and tools by making the work open source, and others are welcome to contribute additional tests and examples.

### Conclusion

Automated testing can help catch many types of problems, especially when several techniques and approaches are used in combination. It's good to keep this in mind so we know where these automated tests fit within our overall testing approach.


With regard to the automated tests we conducted on the Google sites, the ROI for the amount of code written has justified the work. Running the tests discovered bugs that were fixed in several frontline Google properties and tools. Conservatively, the page-weight of many millions of Web re-

quests has been reduced because of problems discovered and fixed using this test automation. Keyboard navigation has also been improved for those who need or prefer using it.

Test automation is imperfect and limited, yet it can be useful in catching various problems that would trip up some of your users. The work complements other forms of testing and helps inform the project team and usability experts of potential issues quickly, cost effectively, and reliably.

### Acknowledgments

Thank you to Google for allowing the original work to be open sourced, to eBay for supporting the ongoing work, to Jonas Klink for his contributions, and to various people who contributed to the article and offered ideas. Please contact the author if you are interested in contributing to the project at [julianharty@gmail.com](mailto:julianharty@gmail.com).

Steve Krugg's work is an excellent complement to automated tests. He has written two books on the topic: *Rocket Surgery Made Easy* (<http://www.sensible.com/rocketsurgery/index.html>) and *Don't Make Me Think*, of which three chapters on user testing are available to download for free from <http://www.sensible.com/secondedition/index.html>. 

### Related articles on [queue.acm.org](http://queue.acm.org)

#### Case Study: UX Design and Agile: A Natural Fit?

<http://queue.acm.org/detail.cfm?id=1891739>

#### Orchestrating an Automated Test Lab

Michael Donat

<http://queue.acm.org/detail.cfm?id=1046946>

#### Too Darned Big to Test

Keith Stobie

<http://queue.acm.org/detail.cfm?id=1046944>

**Julian Harty** is the tester at large at eBay, where he is working to increase the effectiveness and efficiency of testing within the organization. He is passionate about finding ways to adapt technology to work for users, rather than forcing users to adapt to (poor) technology. Much of his material is available online. He is a frequent speaker and writes about a range of topics related to technology, software testing and accessibility, among others.

## What can software vendors do to make the lives of system administrators a little easier?

BY THOMAS A. LIMONCELLI

# A Plea from Sysadmins to Software Vendors: 10 Do's and Don'ts

A FRIEND OF mine is a grease monkey: the kind of auto enthusiast who rebuilds engines for fun on a Saturday night. He explained to me that certain brands of automobiles were designed in ways to make the mechanic's job easier. Others, however, were designed as if the company had a pact with the aspirin industry to make sure there are plenty of mechanics with headaches. He said those car companies hate mechanics. I understood completely because, as a system administrator (sysadmin), I can tell when

software vendors hate me. It shows in their products.

A panel discussion at the Computer-Human Interaction for Management of Information Technology (CHIMIT) 2009 conference focused on a number of do's and don'ts for software vendors looking to make software that is easy to install, maintain, and upgrade. This article highlights some of the issues uncovered at that meeting. CHIMIT is a conference that focuses on computer-human interaction for IT workers—the opposite of most CHI research, which is about the *users* of the systems that IT workers maintain. This panel turned the microscope around and gave sysadmins a forum to share how they felt about the speakers who were analyzing them.

Here are some highlights:

**1. DO have a “silent install” option.** One panelist recounted automating the installation of a software package on 2,000 desktop PCs, except for one point in the installation when a window popped up and the user had to click OK. All other interactions could be programmatically eliminated through a “defaults file.” Linux/Unix tools such as Puppet and Cfengine should be able to automate not just installation, but also configuration. Deinstallation procedures should not delete configuration data, but there should be a “leave no trace” option that removes everything except user data.

**2. DON'T make the administrative interface a GUI.** Sysadmins need a command-line tool for constructing repeatable processes. Procedures are best documented by providing commands we can copy and paste from the procedure document to the command line. We cannot achieve the same repeatability when the instructions are: “Checkmark the 3rd and 5th options, but not the 2nd option, then click OK.” Sysadmins do not want a GUI that requires 25 clicks for each new user. We want to craft the commands to be exe-

cuted in a text editor or generate them via Perl, Python, or PowerShell.

### **3. DO create an API so the system can be remotely administered.**

An API gives us the ability to do things with your product you didn't think about. That's a good thing. Sysadmins strive to automate, and automate to thrive. The right API lets me provision a service automatically as part of the new employee account creation system. The right API lets me write a chat bot that hangs out in a chat room to make hourly announcements of system performance. The right API lets me integrate your product with a USB-controlled toy missile launcher. Your other customers may be satisfied with a "beep" to get their attention; I like my way better (<http://www.kleargear.com/5004.html>).

### **4. DO have a configuration file that is an ASCII file, not a binary blob.**

This way the files can be checked into a source-code control system. When the system is misconfigured it becomes important to be able to "diff" against previous versions. If the file cannot be uploaded back into the system to recreate the same configuration, then we can not trust that you are giving us all the data. This prevents us from cloning configurations for mass deployment or disaster recovery. If the file can be edited and uploaded back into the system, then we can automate the creation of configurations. Archives of configuration backups make for interesting historical analysis.<sup>1</sup>

**5. DO include a clearly defined method to restore all user data, a single user's data, and individual items (for example, one email message).** The method to make backups is a prerequisite, obviously, but we care primarily about the restore procedures.

**6. DO instrument the system so we can monitor more than just, "Is it up or down?"** We need to be able to determine latency, capacity, and utilization, and we must be able to collect this

data. Don't graph it yourself. Let us collect and analyze the raw data so we can make the "pretty picture" graphs that our nontechnical management will understand. If you are not sure what to instrument, imagine the system being completely overloaded and slow: what parameters would we need to be able to find and fix the problem?

### **7. DO tell us about security issues.**

Announce them publicly. Put them in an RSS feed. Tell us even if you don't have a fix yet; we need to manage risk. Your public relations department does not understand this, and that's OK. It is your job to tell them to go away.

### **8. DO use the built-in system logging mechanism (Unix syslog or Windows Event Logs).**

This allows us to leverage preexisting tools that collect, centralize, and search the logs. Similarly, use the operating system's built-in authentication system and standard I/O systems.

### **9. DON'T scribble all over the disk.**

Put binaries in one place, configuration files in another, data someplace else. That's it. Don't hide a configuration file in /etc and another one in /var. Don't hide things in \Windows. If possible, let me choose the path prefix at install time.


### **10. DO publish documentation electronically on your Web site.**

It should be available, linkable, and findable on the Web. If someone blogs about a solution to a problem, they should be able to link directly to the relevant documentation. Providing a PDF is painfully counterproductive. Keep all old versions online. The disaster recovery procedure for a five-year-old, unsupported, pathetically outdated installation might hinge on being able to find the manual for that version on the Web.

Software is not just bits to us. It has a complicated life cycle: procurement, installation, use, maintenance, up-

grades, deinstallation. Often vendors think only about the use (and some seem to think only about the procurement). Features that make software more installable, maintainable, and upgradable are usually afterthoughts. To be done correctly, these things must be part of the design from the beginning, not bolted on later.

Be good to the sysadmins of the world. As one panelist said, "The inability to rapidly deploy your product affects my ability to rapidly purchase your products."

I should point out this topic was not the main point of the CHIMIT panel. It was a very productive tangent. When I suggested that each panelist name his or her single biggest "don't," I noticed the entire audience literally leaned forward in anticipation. I was pleasantly surprised to see software developers and product managers alike take an interest. Maybe there's hope, after all. 

#### Related articles on [queue.acm.org](http://queue.acm.org)

##### **Error Messages: What's the Problem?**

Paul P. Maglio, Eser Kandogan  
<http://queue.acm.org/detail.cfm?id=1036499>

##### **Facing the Strain**

Kode Vicious  
<http://queue.acm.org/detail.cfm?id=1160442>

##### **A Conversation with Phil Smoot**

<http://queue.acm.org/detail.cfm?id=1113332>

#### **Reference**

1. Plonka, D., Tack, A. J. An analysis of network configuration artifacts. In *Proceedings of the 23rd Large Installation System Administration Conference* (Nov. 2009), 79–91.

**Thomas A. Limoncelli** is an author, speaker, and system administrator. His books include *The Practice of System and Network Administration* (Addison-Wesley) and *Time Management for System Administrators* (O'Reilly). He works at Google in New York City.

#### **Acknowledgments**

I would like to thank the members of the panel: Daniel Boyd, Google; Aileen Frisch, Exponential Consulting and author; Joseph Kern, Delaware Department of Education; and David Blank-Edelman, Northeastern University and author. I was the panel organizer and moderator. I would also like to thank readers of my blog, [www.EverythingSysadmin.com](http://www.EverythingSysadmin.com), for contributing their suggestions.

© 2011 ACM 0001-0782/11/0200 \$10.00



## How can system administrators reduce stress and conflict in the workplace?

BY CHRISTINA LEAR

# System Administration Soft Skills

SYSTEM ADMINISTRATION CAN be both stressful and rewarding. Stress generally comes from outside factors such as conflict between system administrators (SAs) and their colleagues, a lack of resources, a high-interrupt environment, conflicting priorities, and SAs being held responsible for failures outside their control.

What can SAs and their managers do to alleviate the stress? There are some well-known interpersonal and time-management techniques that can help, but these can be forgotten in times of crisis or just through force of habit. The purpose of this article is to restate these maxims and remind readers of these important soft skills, particularly as they apply to SAs.

**Conflicts with colleagues.** SAs often feel their efforts are not appreciated and their department is the butt of jokes or a source of frustration for the rest of the company. The sources of these conflicts can be varied. The attitude that the SAs project and how they

are perceived by their colleagues, how they prioritize their workloads, how they follow through, the first impressions they make on their colleagues, and poor communication skills are all pieces of the puzzle. The conflict is often exaggerated in an engineering environment where technology-savvy employees have different needs and expectations of their computing environment.

**Attitude.** One of the greatest causes of conflict is the attitude that SAs present to their colleagues. SAs are sometimes perceived as unfriendly, unhelpful, or slow to respond. How people perceive you is directly related to the attitude you project.

The number-one attitude problem among SAs is a blatant disrespect for the people they are hired to support. End users are not “lusers” or “pests with requests.” Often a change of vocabulary helps. Refer to these end users as *colleagues*<sup>5</sup> or *customers*.<sup>4</sup> This terminology is a reminder that SAs and end users are on the same team and that SAs are in a service industry, supporting the needs of the end users.

Beware, however, of thinking “the customer is always right.” Part of an SA’s job is to (politely) say no when appropriate. Remember that just because you can do something doesn’t mean you should. Teach your colleagues how to do things themselves, provide documentation, and make sure they do not need to ask time and time again how to do something. It is also the responsibility of an SA to politely reject requests that are against policy. As a general rule, if you are comfortable with colleagues performing certain tasks (for example, they can’t break anything and it’s not against policy), then you should enable those colleagues to do it themselves.

Another attitude problem that SAs can develop is avoiding colleagues who bring them nothing but complaints or problems. You need to think of each problem as a challenge to solve and a way to demonstrate your expertise. Be glad that someone found

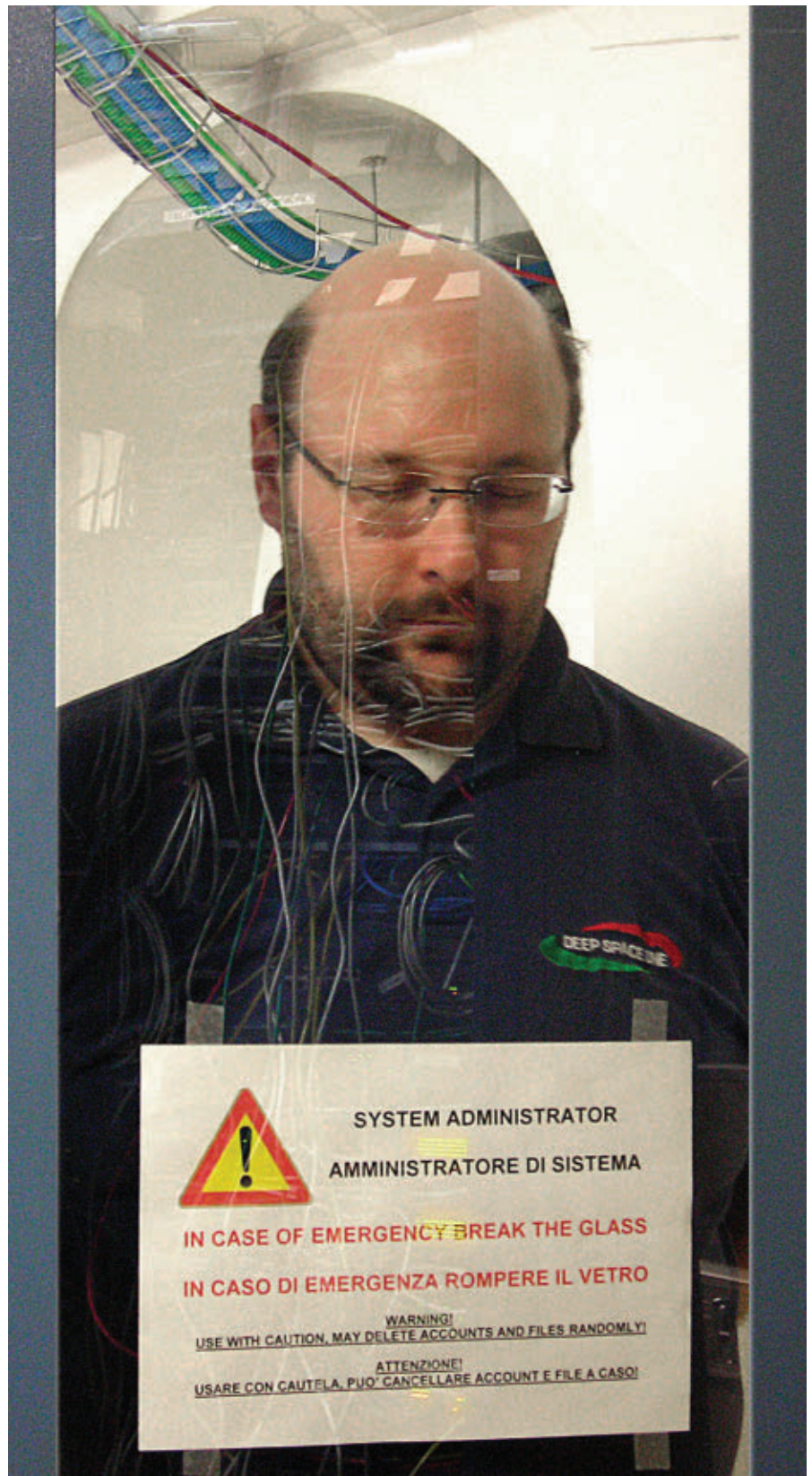
the problem and took the time to tell you about it. This gives you a chance to get the whole system working better.

**Align priorities with colleagues' expectations.** As an SA, the way you prioritize your tasks influences how your colleagues perceive your effectiveness. You can make your colleagues a lot happier and avoid conflicts if your priorities match their expectations.

People expect small things to happen quickly and big things to take a reasonable amount of time. The definition of *big* and *small* is theirs and is based on their perception of your job. For example, resetting a password, which is perceived as taking a minute or two, should happen quickly. Installing a new computer is perceived as a bigger process, and it is reasonable to take a day or two. When a critical server is down, your colleagues expect you to be working on nothing but the emergency.

Prioritize requests so that emergencies are handled first, then quick requests, followed by longer requests. This will result in higher customer satisfaction than if the same tasks were done in a different order. Sometimes what your colleagues think is a big task is actually a small one. This is not a problem—if you perform the task more quickly than expected, then everyone is happy. But if they perceive a big task as small, this is a potential source of conflict. For larger jobs it is a good idea to ask your colleagues what their expectations are, explain what is involved in performing the tasks, and, where appropriate, reset their expectations to more realistic levels.

**Follow through on commitments.** Dropping customer requests and not following through on commitments are sure ways to create or exacerbate conflict with work colleagues. The first step toward following through on all commitments is to keep track of them all in one place, either on paper or electronically, not in your head. A trouble-ticket system is a useful tool for keeping track of requests and other tasks. From your master list of tasks





and commitments you should create daily prioritized to-do lists. The section on time management later in this article discusses managing these daily to-do lists in more detail.

A daily to-do list is an effective reminder of your commitments. If you see that you are overloaded and will miss a deadline, try to negotiate a new deadline. Then you will know whether it is so critical to complete the task today that you must reprioritize, or work late, or whether it can wait a day or two. The sooner you communicate with your colleagues, the easier it is to reprioritize your work and the easier it is for them to adjust their work to the new schedule.

**Make a good first impression.** The first impression that you make with colleagues dominates all future interactions with them. If you get off on the wrong foot, it is difficult to recover, and you will experience more conflicts. On the other hand, if you make a good first impression, the occasional mistake will be forgiven and forgotten.

Be on time or early for appointments, polite, friendly, and willing to listen. Most people are visual beings, so appearance and facial expression are two things they notice first. Appropriate attire is different at different companies, so tailor your appearance to your environment. At one company an SA wore baggy overalls and dyed her hair bright pink. The group she supported was very respectful and enthusiastic about her. She gained quick acceptance from her customers because they figured that anyone who dressed that way and got away with it must be really good at what she does. She retained their respect by being good at her job. On the other hand, at another company an SA dressed provocatively because she felt that when her heterosexual male colleagues found her attractive, they were nicer to her. Being treated nicely and being respected, however, are two different things.

Most importantly, listen to what your colleagues are saying and take notes. There is nothing more frustrating than trying to explain something to someone who is not really listening and just assumes that he or she knows what you want. Specific techniques

for improving your listening and communication skills are covered in more detail in the next section.

Making a good first impression on new hires begins before their first day at work. They are motivated and want to be productive right away, so you need to make sure that when they arrive, they will find their computers in their offices, already configured, accounts created, and everything working properly. SAs need to know who is starting, on what date, and what the computing needs are. On the employee's first day, the friendliest member of your SA team should visit the person to do some in-person orientation, answer questions, and personally deliver a printed "welcome to our network" guide.

**Improve your communication skills.** Conflicts often arise through simple miscommunication. There are a few well-known techniques for improving communication skills that SAs can apply successfully.

When you have a problem, you need to make sure you are being heard. Do this using "I statements." This is a tool to help you make your point and communicate your feelings. The general form is: "I feel [emotion] when you [action]." This makes people aware of the effect of their actions. Express soft emotions—sadness or fear—rather than hard emotions—anger. Anger makes people defensive, whereas soft emotions inspire people to help.

When someone brings a problem to you, you need to make sure that you are hearing that person properly. "Active listening" is a technique that ensures complete communication. You should seek to understand what was said before replying, and your next statement should mirror what you just heard with an accurate but shorter statement. It's like verifying a packet checksum before using the data. A mirror statement could begin with "I hear you saying that...." A summary statement is a form of mirroring but covers more material. It is often useful toward the end of a meeting or after a person has completed several long points and you want to make sure that you heard everything—and heard it correctly.

"Reflection" is a technique that assures people their emotions are being

recognized. This is particularly important when the person you are dealing with is angry or upset, and it is more effective than becoming defensive. For example, if someone yells at you about something, try saying, "Wow! You are really upset about this!" Acknowledging the other person's emotion gives him or her a chance to calm down, at which point you can have a rational discussion using the active listening techniques.

Finally, pay attention to how much technical jargon you use, and tailor your style to suit your audience. Some of your colleagues will want to hear the jargon just to be reassured that you know what you're doing. Others are confused or intimidated by it and just want plain language. If you are not sure where to aim, start with a mix of jargon and plain language and watch for hints (facial expression, body language, or comments) about whether you should simplify further or just stick with the jargon. Asking "How much detail do you want?" is a good way of letting the other person set the level of the conversation, without sounding condescending.

**Be a system advocate.** Conflicts can also arise because of the perceived role of SAs at a given company. Your colleagues perceive SAs as being somewhere between clerks who reactively perform menial tasks and advocates who proactively solve their problems and lobby for their needs. For a lone SA, it is better to be seen as an advocate than a clerk. A large SA group needs the whole spectrum, with the advocates mentoring the more junior clerks, but the group as a whole must be seen as advocates.

To understand how being seen as an advocate can help prevent conflict, consider the following scenario: A non-SA colleague identifies a piece of software that he needs to do his job. If he views the SA as a clerk, he orders the software and says nothing until it arrives, at which point he expects it to be installed immediately. The SA then discovers that the colleague has licensed it to a random machine (perhaps his desktop) rather than the license server, that the machine it needs to run on doesn't have enough CPU capacity, memory, or disk space, that the operating-system version is




not supported by the software, that the machine has insufficient graphics capability, that he is one of several who independently ordered the software, or any number of other issues. All of these issues take time, and often money, to resolve. The colleague's expectations of a quick install are not met, and both he and the SA are frustrated, leading to conflict.

When the SA is seen as an advocate, on the other hand, he or she is involved from the beginning, compiles a set of requirements, finds out who else might need the software, and so on. Then the software licenses are linked to the license server, there are sufficient licenses for all who need it, all the hardware and related issues are resolved in advance, and everyone has a realistic expectation for when the new software will be available.


Advocates are proactive, identifying potential problems and solving them before they arise. They use extensive monitoring to track peak loads and usage trends. Using this data the advocate upgrades networks and services before they get overloaded and slow people down.

Advocates interact with customers on a regular basis and are aware of future requirements. Their colleagues know to involve them in the planning phases for new endeavors. One result of such a team effort is a seamless, smooth-running network that meets the end users' needs. Another is having colleagues who are more invested in the evolution of the network.

**Engineering environment.** The conflict between SAs and their colleagues is often worse in an engineering environment, where technology-savvy engineers want more control over their own machines in order to be able to work more efficiently. SAs know, however, that unfettered root access for people outside the SA group leads to randomly configured systems, more failures, and more support calls. This struggle for control and the importance of finding the right balance for each environment has been discussed elsewhere.<sup>3</sup> The key point to remember is that SAs need to foster trust and build good relationships with their colleagues. The end users really just want to be able to do their jobs efficiently and effectively. They need to



**The number-one attitude problem among SAs is a blatant disrespect for the people they are hired to support. End users are not “lusers” or “pests with requests.”**



trust that the SAs will be enablers and not roadblocks. The techniques already described in this article will also help to foster trust and reduce conflict in an engineering environment.

Another conflict that can arise in an engineering environment is between the SA's desire for high reliability and the engineer's desire for cutting-edge technologies that may not yet be ready for primetime. In some environments SAs need to sacrifice uptime in order to deliver the best overall service. This is particularly true where the engineers in question are the ones developing the new technologies.

### **Lack of Resources**

Most SA groups are pressed for time and money. The first thing to do in this situation is to make the most of what you have using automation, time management, and organizational structures. Once that is done, the managers can lobby for more resources by improving the perception and visibility of the SA group.

**Automation.** A good way to address a lack of resources is to create additional time for the SAs by automating the most time-consuming tasks. Automation saves time both by getting tasks done more quickly and by ensuring consistency, thus reducing support calls.

Start with a script that outputs the commands that would do the task. The SA can review the commands for correctness, edit them for special cases, and then paste them to the command line. Writing such scripts is usually easier than automating the entire process and can be a stepping-stone to further automation of the process.

A simple script that assists with the common case may be more valuable than a large system that automates every possible aspect of a task. Automate the 80% that is easy and save the special cases for the next version. Document which cases require manual handling, and what needs to be done.


Look for vendor-supplied automation tools for tasks such as operating-system installs, and use them. Figure out how to automate customizations for your environment, too. Where possible, automate tasks that are common requests from customers and create a Web page to make these

requests self-service. This approach saves time for both the SAs and the customers, and it increases customer satisfaction.


**Time management.** Another way to make the most of available resources is through the application of various well-known time-management techniques. Time management means using time wisely—working smarter, not harder. The topic of how SAs can better manage their time is a book in itself.<sup>1</sup> Time management can be particularly difficult for SAs because their job is typically interrupt-driven. To be more productive, it is important to break this cycle. You can deflect an interruption by writing the request into your personal to-do list and telling the person that you will get to it later. If you are unable to write it down, then politely ask the person to send you an email message or trouble-ticket request. Make it easier for the person by suggesting the wording that would be most useful to you.

Often the most productive, least interrupted time of day is the first hour in the morning, so don't waste it reading email. Quickly check the monitoring system for problems, and your email for items tagged "Urgent." Then edit and prioritize your daily to-do list, rescheduling some items for another day if there is too much. Then schedule your day with a granularity that works for you (for example, in half-hour, one-hour, two-hour, or half-day increments). Daily prioritized to-do lists make the "what next?" decision easier and quicker. Spend the rest of that first hour working on your highest-priority item. At the end of the day copy the items that remain unfinished on your to-do list to the next day's list.

Handle each piece of paper or email once. Don't even look at something if you don't have time to deal with it. Process each item completely the first time, rather than sorting into piles and then having to reread it later. As you touch each item, examine it and decide whether you are going to throw it away without reading it, read it and throw it away, deal with it and then throw it away, respond to it and then throw it away, or file it. Sometimes, dealing with an item means recording it in your to-do list. Other times, you can quickly reply to an email or write



**The key point to remember is that SAs need to foster trust and build good relationships with their colleagues. The end users really just want to be able to do their jobs efficiently and effectively. They need to trust that the SAs will be enablers and not roadblocks.**



your response in the margin of a paper document and send it back to the party who sent it. File as little as possible. When in doubt, throw it out.

Automate as much of your email processing as possible. For example, automate sorting of email into folders per email list or forum, notifications from social networking sites, blog posts, and non-spam coupons, updates, and special offers from vendors. Then decide how often you want to scan those folders for items of interest, and even set up automated deletions after a certain number of days. Have folders for storing information for set periods of time (for example, one week, one month, two months, six months, one year) and auto-delete items older than the specified time. Keep refining and updating your automation.

Stay focused. A clean desk, a clean computer desktop with virtual screens for each task, and a clean email box reduce distractions and help you maintain focus. Disabling email alerts also helps. Schedule time for checking email rather than looking at each message as it arrives. Merlin Mann, author of *Inbox Zero* (<http://inboxzero.com/>) has several tips for emptying your inbox and keeping it that way.

Look for ways to reduce the time each task takes. Automation is one way to do that. Another way is to precompile decisions, such as deciding always to make a backup copy of a file before editing it, to bring your PDA, to write down requests, to do small tasks sooner rather than later, or when to restock printers with paper. Sometimes, the solution is as simple as keeping spare toner cartridges near the printer so that the time to install them is not dominated by the time to walk to a distant supply room.

**Organizational structures.** Making SAs more productive and less interrupt-driven yields the best use of the limited resources available. Switching from one job to another takes time. The more context switches an SA has to perform, the more time is wasted and the less effective the SA can be. Controlling the number of interrupts an SA experiences during the day is probably the most effective method of reducing stress and increasing productivity.

Controlling interrupts can be achieved through changing the structure of the SA group. Divide the SA team so that front-line support people perform requests that customers expect to see done quickly. Requests that will take more time can be passed on to second-tier personnel. Senior SAs can be in charge of larger projects, such as the creation of services. This division of labor ensures your priorities are aligned with your colleagues' expectations and shelters people who are working on long-term projects from continual interruptions. This may sound like something only a large SA team can afford to do, but even a team of two SAs can benefit from this technique. One SA can shield the other from interruptions in the morning and vice versa in the afternoon. This is called the mutual-interruption shield technique.<sup>2</sup>

**Perception and visibility.** Many of the stress factors that SAs face, including a lack of resources, can result from problems with perception and visibility.

► *Perception* is how people see you; it is a measure of quality.

► *Visibility* is how much people see of you; it is a measure of quantity.

The importance of being perceived well is clear. The importance of being visible, perhaps less so. When SAs are not visible, they may be assumed not to be contributing, not to be busy, to be overstaffed or overfunded, or to be otherwise unnecessary. This can result in underfunding and understaffing, leading to worse perceptions and poorer visibility.

Most of the techniques discussed here deal with improving how SAs are perceived. Be aware that if you are poorly perceived, it takes a lot of time and effort to turn things around. SAs can do a lot to improve the visibility of their work, but they should try to improve visibility only if they are actually doing a good job. In other words, don't publicize a bad product.

For example, to increase your visibility, create a system-status Web page that puts you in front of customers' eyes daily. Make it a page that also has other useful information and links so that it becomes a home page. Meet regularly with managers to help them understand what you do and help you

maintain focus on their highest priorities.

Pay attention to your office locations. Customer-facing people should be in the more visible locations. Hold town hall and user group meetings where every idea, request, or criticism is written down without judgment or objections. Be clear that writing it down is a commitment to consider the issue, but not necessarily to implement something.

Newsletters are often produced by SA groups but rarely read by customers. They are a lot of work to produce and too easy to ignore. Having lunch and social functions with customers is a simple way of maintaining interaction and is usually more effective and less time consuming than a newsletter.

Take responsibility for your, and your team's, positive visibility by improving the perception and visibility of the group. SA managers can use their teams' positive visibility to argue for more resources.

### Conflicting Priorities

SAs can end up with a number of conflicting high-priority requests, resulting in more stress. Try to resolve conflicting priorities by talking to your affected colleagues, or perhaps their manager, to persuade them to decide among themselves what the priorities are. If more than one group is involved, get the managers in a room together and let them figure it out. If one of the tasks is something that affects the SA group, get your manager in on the discussion. You may feel that you have enough information about business priorities to make the decision, but it is often better to involve your colleagues, so that they have a better idea of what you are working on and why their requests are being delayed. This approach can also aid you and your manager when you request more resources in the next budget.

### End-to-End Responsibility

SAs are often held accountable for every failure, regardless of whether or not they have control over the component that failed. SAs are the central clearinghouse for all problems. Embrace that role rather than fight it. Don't expect your colleagues to know

that the internal Web server has failed because of content that another group placed there rather than an operating-system, network, or hardware issue.

If it is not something you can fix (for example, by removing the offending content or rolling back a software release), then use your system-status page, trouble-ticket system, or phone messages to let people know that the problem is being worked on and whom they can talk to for more information.

Everyone has experienced the finger-pointing phenomenon at some point, where for every problem someone is pointing a finger at someone else. With a complex system, the easiest way to "get rid of someone" is to tell that person to talk to someone else, that it's not your problem. Don't fall into that trap. Rather than trying to duck the problem, act as the clearinghouse. Get everyone together to figure out what the problem is and get the right people working on the solution. If your trouble-ticket system is used to track statistics for how long calls take to resolve, make sure there is a way to mark the ticket as waiting for input from another source. Don't just punt the problem and forget about it. Keep checking for solutions. Set up automated reminders that you need to check for a solution.

### Physical Well-Being

An important part of managing stress, often neglected by SAs, is taking care of one's body. Physical exercise is an excellent form of stress relief and has the added benefit of improving mental alertness and stamina. It should be scheduled as part of your weekly routine, so that you don't have to decide when to "make time"; you just go when it is time. For example, decide to exercise every Monday, Wednesday, and Friday before work, or at lunchtime, or at 6 P.M., but not vaguely "after work" or "after I get home." If the time is nebulous, it is not part of a routine and it will get skipped.

Getting enough sleep and eating properly are also important components of physical well-being. Someone who has not had enough sleep makes poor decisions, cannot concentrate, makes mistakes, and works more slowly. Be at your best at work by



taking care of yourself outside work and by having enough personal time. Vacations are important. Use them to give your mind a break so that you come back refreshed. Disconnect. Be out of touch. Trust your colleagues to survive without you.

### The Future

An SA's job is constantly changing because of new technologies and the growing sophistication of the customer base. How do those changes alter which soft skills are required, or do they?

When my mother started out as an SA (actually an "operator"), only SAs had access to the machines, and they fed their colleagues' programs into the card reader and handed back the results when the program was finished. Her colleagues' expectations of how quickly tasks got done were vastly different from what we are accustomed to today. Her customer base was much smaller and universally tech savvy but was not as reliant on computers for everything. Her work was not as interrupt-driven as today, she had no email to handle, and her day consisted predominantly of project work rather than many small tasks for many different people. The soft skills required were inherently different from those discussed in this article.

Over the past four decades the soft skills that SAs need have changed substantially, and I expect that in the next four decades there will be more significant changes, many of which will be driven by changes in technology that are impossible to predict so far in advance. We can anticipate, however, that in the coming decades the soft skills discussed in this article are going to become increasingly important for SAs.

The connected population continues to grow, as do the ways to be connected and to (re)connect with others, yielding ever increasing electronic communications. There is an increasing "always-on" expectation that if you are online (and why would you not be?), you can respond to any message immediately. Electronic communications cannot always create interrupts. Time management and taking control of electronic communications, rather

than letting them control us, are going to become increasingly important for everyone, but especially for interrupt-prone SAs.

Given the inevitable continued increases in the quantity of electronic communications that we all receive, we need to look at the quality and quantity of what we send. SAs need to learn to be brief and to the point, without being rude. It saves you time while writing and saves your colleagues time while reading.

The SA's customer base is going to become increasingly remote and mobile. Telecommuting will become feasible for more people, as will working while commuting. Outsourcing and international offices are going to continue to be factors, placing SAs in locations that are remote from their customers. When SAs are not physically close to their colleagues, they need to be sure to address the issues of perception and visibility mentioned here. It is also worth noting that it is often better to pick up the phone rather than deal with remote colleagues by email or instant messaging. It is more personal, issues can be resolved more quickly, and there is less chance for misunderstanding. It also gives you an opportunity to use the communication skills described earlier.


Mobile computing devices are only going to become more common—integral to everyone's productivity—with all their attendant technological challenges. Don't fight it, but recognize it, embrace the challenge, and manage your colleagues' support expectations. The same goes for future technologies. Be the early adopter. Look for how the latest thing can become useful to everyone and what changes are needed to make it so.

As computing continues to become more ubiquitous, previously independent systems become integrated into the computers and networks that SAs support. The expectation that things will "just work" grows, as does the stress when something stops working.

Some of your colleagues will become more sophisticated in their requirements, and some of your less sophisticated colleagues will become reliant on systems you support. You need to develop the ability to communicate effectively with and support

customers at every level of technological sophistication.

### Conclusion

The role of SA can be stressful, but once you recognize what some of the stress factors are, you can alleviate much of that stress and turn the job into the rewarding position that it should be. There are various methods for reducing conflicts with colleagues, methods for coping with a lack of resources and an interrupt-driven environment, resolving conflicting priorities, and embracing the fact that SAs are held responsible for every failure. The discussion in this article was necessarily brief, but for those who would like more detail, all of these topics are described in greater depth in *The Practice of System and Network Administration* (Addison-Wesley, 2007).<sup>2</sup> 

#### Related articles on queue.acm.org

##### Collaboration in System Administration

Eben M. Haber, Eser Kandogan, Paul Maglio  
<http://queue.acm.org/detail.cfm?id=1898149>

##### Avoiding Obsolescence

Kode Vicious  
<http://queue.acm.org/detail.cfm?id=1781175>

##### Beyond Instant Messaging

John C. Tang, James "Bo" Begole  
<http://queue.acm.org/detail.cfm?id=966718>

#### References

1. Limoncelli, T.A. *Time Management for System Administrators*. O'Reilly, 2005.
2. Limoncelli, T.A., Hogan, C., Chalup, S.R. *The Practice of System and Network Administration*, 2nd edition. Addison-Wesley, Reading, PA, 2007.
3. Owen, H. The problem of PORCMOLSULB. *login: The USENIX Magazine* 27, 4 (Aug. 2002).
4. Smallwood, K.C. SAGE views: Whither the customer? *login: The USENIX Magazine* 17, 5 (Sept. 1992).
5. Zwicky, E.D. SAGE views: The customer isn't always right; the customer isn't always even a customer. *login: The USENIX Magazine* 17, 6 (Nov. 1992).

**Christina Lear** worked for more than 10 years as a system administrator. She is a member of SAGE (the Usenix Special Interest Group for Sysadmins) and has been heavily involved in the Usenix LISA (large installation system administration) community. She co-authored *The Practice of System and Network Administration* with Thomas Limoncelli, for which they received the SAGE Outstanding Achievement Award in 2005. More recently she obtained a Ph.D. in aeronautical engineering and has worked as an aerodynamicist for a Formula 1 racing team.

*Introducing:*

# XRDS

The ACM Magazine for Students

*XRDS* delivers the tools, resources, knowledge, and connections that computer science students need to succeed in their academic and professional careers!

The **All-New *XRDS: Crossroads*** is the official magazine for ACM student members featuring:

- › Breaking ideas from top researchers and PhD students
- › Career advice from professors, HR managers, entrepreneurs, and others
- › Interviews and profiles of the biggest names in the field
- › First-hand stories from interns at internationally acclaimed research labs
- › Up-to-date information on the latest conferences, contests, and submission deadlines for grants, scholarships, fellowships, and more!



**Also available**

**The All-New *XRDS.acm.org***

***XRDS.acm.org*** is the new online hub of *XRDS* magazine where you can read the latest news and event announcements, comment on articles, plus share what's happening at your ACM chapter, and more. Get involved by visiting today!

***XRDS.acm.org***



Association for  
Computing Machinery

*Advancing Computing as a Science & Profession*

DOI:10.1145/1897816.1897838

**Body posture and finger pointing are a natural modality for human-machine interaction, but first the system must know what it's seeing.**

**BY JUAN PABLO WACHS, MATHIAS KÖLSCH,  
HELMAN STERN, AND YAEL EDAN**

# Vision-Based Hand-Gesture Applications

THERE IS STRONG evidence that future human-computer interfaces will enable more natural, intuitive communication between people and all kinds of sensor-based devices, thus more closely resembling human-human communication. Progress in the field of human-computer interaction has introduced innovative technologies that empower users to interact with computer systems in increasingly natural and intuitive ways; systems adopting them show increased efficiency, speed, power, and realism. However, users comfortable with traditional interaction methods like mice and keyboards are often unwilling to embrace new, alternative interfaces. Ideally, new interface technologies should be more accessible without requiring long periods of learning and adaptation. They should also provide more natural human-machine communication. As described in Myron Krueger's pioneering 1991 book *Artificial Reality*,<sup>27</sup> "natural interaction" means voice

and gesture. Pursuing this vision requires tools and features that mimic the principles of human communication. Employing hand-gesture communication, such interfaces have been studied and developed by many researchers over the past 30 years in multiple application areas. It is thus worthwhile to review these efforts and identify the requirements needed to win general social acceptance.

Here, we describe the requirements of hand-gesture interfaces and the challenges in meeting the needs of various application types. System requirements vary depending on the scope of the application; for example, an entertainment system does not need the gesture-recognition accuracy required of a surgical system.

We divide these applications into four main classes—medical systems and assistive technologies; crisis management and disaster relief; entertainment; and human-robot interaction—illustrating them through a set of examples. For each, we present the human factors and usability considerations needed to motivate use. Some techniques are simple, often lacking robustness in cluttered or dynamic scenarios, indicating the potential for further improvement. In each, the raw data is real-time video streams of hand gestures (vision-based), requiring effective methods for capturing and processing images. (Not covered is the literature related to voice recognition and gaze-tracking control.)

## » key insights

- Gestures are useful for computer interaction since they are the most primary and expressive form of human communication.
- Gesture interfaces for gaming based on hand/body gesture technology must be designed to achieve social and commercial success.
- No single method for automatic hand-gesture recognition is suitable for every application; each gesture-recognition algorithm depends on user cultural background, application domain, and environment.





## Basic Communication Form

We humans use gestures to interact with our environment during the earliest stages of our development. We also communicate using such gestures as body movement, facial expression, and finger pointing. Though much has been written about gesture interfaces, interface technology rarely adopts this media; consequently, expressiveness and naturalness elements are missing from most user interfaces. Hand-gesture applications provide three main advantages over conventional human-machine interaction systems:

*Accessing information while maintaining total sterility.* Touchless interfaces are especially useful in health-care environments;

*Overcoming physical handicaps.* Control of home devices and appliances for people with physical handicaps and/or elderly users with impaired mobility; and

*Exploring big data.* Exploration of large complex data volumes and manipulation of high-quality images through intuitive actions benefit from 3D interaction, rather than constrained traditional 2D methods.

Human-robot interaction is another application where the main motivation for gesture-based systems is to have this communication resemble natural human dialogue as much as possible. For example, imagine how intuitive it could be to use hand gestures to tell a robot what to do or where to go. Pointing to a dust spot to indicate “Clean that spot,” users would be able to tell a Roomba robotic vacuum cleaner what to do next. Finally, gestures provide a source of expressiveness when immersed in realistic video games. Some notable technologies (such as Microsoft Kinect, Sony PSP, and Nintendo DS and Wii) include gesture recognition in their consoles. Unfortunately, only dynamic gestures (such as waving and fist hitting) are recognized so far. Dynamic hand-shape recognition, as in American Sign Language, remains a challenge.

## Costs/Benefits

The appeal of gesture interfaces derives partly from their flexibility and customizability. Still, many requirements as to their functionality and performance are the same throughout

most classes of use. As devices and hand-gesture interfaces proliferate as a result of inexpensive cameras and computational power, questions concerning market acceptance also become more frequent. Here are the basic requirements, though they are likely to vary depending on application:

**Price.** Better camera quality, frame-rate, distortion, and auto-shutter speed yield better performance but higher cost. Some inexpensive methods for achieving 3D reconstruction (such as flashing IR LED illuminators from multiple angles) can replace stereo cameras. But the sum of the prices for discrete hardware components can add up for the typical consumer, as well as for a manufacturer. The cost of more advanced sensors and sensor setups must be weighed against any potential performance benefit.

**Challenges.** Given a fixed budget, the challenge for the developer is to decide how the development budget should be spent and, especially, which hardware the system cannot do without.

**Responsiveness.** The system should be able to perform real-time gesture recognition. If slow, the system will be unacceptable for practical purposes. In 1963, Sheridan and Ferrell<sup>43</sup> found maximum latency between “event occurrence” and “system response” of 45ms was experienced by most of their human test subjects as “no delay.” Starting at 300ms, an interface feels sluggish, possibly provoking oscillations and causing a symptom known as “move and wait.”

**Challenges.** Simple, computationally efficient features are of great interest to machine-vision researchers, though more effective techniques must still be developed.

**User adaptability and feedback.** Some systems are able to recognize only a fixed number of gestures selected by the system designer; others adapt to a nuanced spectrum of user-selected gestures. The type of gesture selected depends on the application; for example, in video games, learning gestures is part of a gratifying experience playing the game. In either case, feedback indicating the correctness of the gesture performed is necessary for successful interaction.

**Challenges.** Most hand-gesture systems have a core algorithm trained

offline (not in real time). Training a classifier online requires a fast, flexible online learning algorithm capable of generalizing from a few training samples. Presenting feedback to the user without increasing cognitive load is an additional problem.

**Learnability.** Gesture patterns (the lexicon) used to control applications must be easy to perform and remember. These factors are strongly associated with learning rate and “memorability” indices, as reported by Wachs.<sup>51</sup>

**Challenges.** The learning rate depends on task, user experience, and user cognitive skills. Hardly any literature exists on user performance as a function of gesture vocabulary size or user experience. Two exceptions are by Nielsen<sup>34</sup> and by Kela et al.<sup>23</sup> focusing on acceleration-based gestures. A possible solution is to adopt gestures that are natural and intuitive to the user; users are also more likely to remember them.

**Accuracy (detection, tracking, and recognition).** Among these three main criteria affecting the performance of hand-gesture systems, detection describes whether a hand is in the camera’s view. Tracking describes the ability to follow the hand from frame to frame. And recognition is based on how close the hand’s trajectories are to learned templates, based on distance metrics, and indicates the level of confusion of the given gesture with other gestures. For this article, we limit ourselves to performance measures for per-frame-analysis as opposed to activity-recognition systems where more complex performance measures are considered.<sup>32</sup>

**Challenges.** The main challenges for the three performance measures are at the forefront of research in machine vision. Detection is an extremely complex problem due to hand shape, variable lighting conditions, skin color, and hand size. Tracking complications arise from occlusions, cluttered environments, and rapid motions causing motion blur. Addressing these challenges allows good recognition accuracy to follow.

**Low mental load.** Having to recall gesture trajectories, finger configurations, and associated actions is likely to add to a user’s mental load. Another source of mental (and physical) load



is when users' hands cover the display, preventing them from seeing the graphics being guided.


**Challenges.** The gestures should be simple, temporally short, and natural. For a given set of tasks, users should have to remember at most only a few postures. Iconic representations of gesture-command associations may also help relieve users' mental load.

**Intuitiveness.** The gesture types selected by interface developers should have a clear cognitive association with the functions they perform. For example, an open palm can represent a "stop" command, a closed fist with thumb up can represent "OK," and a pointing finger can represent the direction to move an object. Few users are able to remember complex shapes and unnatural finger configurations. Intuitiveness is associated with other usability terms (such as learnability and "easy to remember"). Other factors affecting user-gesture choices are general knowledge, cultural environment, and linguistic capability.<sup>51</sup>


**Challenges.** Intuitiveness is strongly associated with cultural background and experience. A gesture natural to one user may be unnatural to others. Moreover, Stern et al.<sup>46</sup> showed there is no consensus among users regarding gesture-function associations. This problem can be overcome by letting users decide which gesture best represents their intentions. The "Wizard of Oz" paradigm<sup>34</sup> and analytical structured approaches<sup>51</sup> help achieve this representation.

**Comfort.** Lexicon design should avoid gestures that require intense muscle tension over long periods, a syndrome commonly called "Gorilla arm." Gestures must be concise and comfortable while minimizing stress on the hand. Awkward, repetitive postures can strain tissues and result in pressure within the carpal tunnel. Two types of muscular stress are found: static, the effort required to maintain a posture for a fixed amount of time, and dynamic, the effort required to move a hand through a trajectory.

**Challenges.** Measuring stress produced by hand gestures is very difficult. For stress-index measures, experiments vary from subjective questionnaires to electronic devices (such as electromyograms) that measure



**Lexicon design should avoid gestures that require intense muscle tension over long periods, a syndrome commonly called "Gorilla arm."**



muscle activity. The main obstacle with physiological methods is that muscle potentials are highly variable within subjects and depend on external factors like positioning, temperature, and physiologic state. Instead, analytical approaches help assess stress based on the dynamics of musculoskeletal models.

**Lexicon size and multi-hand systems.** For sign languages (such as American Sign Language), hand-gesture-recognition systems must be able to recognize a large lexicon of both single-handed and two-handed gestures. For multi-touch systems, lexicon size plays a minor role. In either case, the challenge is to detect (and recognize) as many hands as possible.

**Challenges.** The different types of gestures to be recognized must be weighed against the system's robustness. A classifier that recognizes a small number of gestures generally outperforms the same system trained on more gestures. The challenge for the vision algorithm is to select robust features and classifiers such that the system's performance is barely affected by lexicon size. Multi-hand systems pose additional challenges (such as disambiguation of mutual hand occlusions and correctly associating hands and people).

**Come as you are.**<sup>48</sup> This phrase refers to an HCI design that poses no requirement on the user to wear markers, gloves, or long sleeves, fix the background, or choose a particular illumination. Many methods encumber the user in order to track and recognize gestures by standardizing the appearance of the hands (markers, gloves, long sleeves) but make interaction cumbersome. The challenge for the vision algorithm is to recognize hand gestures without requiring the user wear additional aids or being wired to a device.

**Challenges.** This flexibility constraint suggests a machine-vision-based solution that is not invasive. The drawback reveals itself with varied environments and user appearance. Assumptions about user characteristics and illumination affect system robustness. Near-IR illuminators can help. Far-IR cameras, ultrasonic, IR laser scanners, and capacitive imagers are also possible approaches for maintaining a system that lets users come as you are.



**Reconfigurability.** Hand-gesture systems are used by many different types of users, and related hand-gesture interfaces are not “one size fits all.” Location, anthropometric characteristics, and type and number of gestures are some of the most common features that vary among users.

**Challenges.** This requirement is not technically challenging; the main problem is the choice of functionalities within the interface that can change and those that cannot. The designer should avoid overwhelming the user by offering infinite tunable parameters and menus. On the other hand, users should have enough flexibility that they can freely set up the system when a major component is replaced or extended.

**Interaction space.** Most systems assume users are standing in a fixed place with hands extended (limited

by a virtual interaction envelope) and within the envelope recognize gestures. But these assumptions do not hold for mobile ubiquitous hand-gesture-recognition systems where the interaction envelope surrounds only the mobile device.

**Challenges.** Recognition of 3D body-arm configurations is usually achieved through at least two cameras with stereo vision, a setup requiring previous calibration and usually slower response than single-camera-based systems. Monocular vision can be used to disambiguate 3D location using accurate anthropomorphic models of the body, but fitting such a model to the image is computationally expensive.

**Gesture spotting and the immersion syndrome.** Gesture spotting consists of distinguishing useful gestures from unintentional movement related to the immersion-syndrome phenomenon,<sup>2</sup> where unintended movement is interpreted against the user’s will. Unintended gestures are usually evoked when the user interacts simultaneously with other people and devices or just resting the hands.

**Challenges.** The main challenge here is cue selection to determine the temporal landmarks where gesture interaction starts and ends; for example, hand tension can be used to find the “peak” of the gesture temporal trajectory, or “stroke,” while voice can be used to mark the beginning and culmination of the interaction. However, recognition alone is not a reliable measure when the start and end of a gesture are unknown, since irrelevant activities often occur during the gesture period. One solution is to assume that relevant gestures are associated with activities that produce some kind of sound; audio-signal analysis can therefore aid the recognition task.<sup>52</sup>

While responsiveness, accuracy, intuitiveness, come as you are, and gesture spotting apply to all classes of gesture interface, other requirements are more specific to the context of the application. For mobile environments in particular, ubiquity and wearability represent special requirements:

**Ubiquity and wearability.** For mobile hand-gesture interfaces, these requirements should be incorporated into every aspect of daily activity in every location and every context; for ex-

ample, small cameras attached to the body or distributed, networked sensors can be used to access information when the user is mobile.

**Challenges.** Hand-gesture systems that are spatially versatile and adaptable to changing environments and users require self-calibration. Small programmable sensors are expensive, and cross-platform environments have yet to be developed.

In a literature review we undertook as we wrote this article, we found that the requirements outlined here are acknowledged by only a few scientists, including Baudel and Beaudouin-Lafon<sup>2</sup> and Triesch and Malsburg.<sup>48</sup>

## Hand-Gesture Recognition

Hand gestures can be captured through a variety of sensors, including “data gloves” that precisely record every digit’s flex and abduction angles, and electromagnetic or optical position and orientation sensors for the wrist. Yet wearing gloves or trackers, as well as associated tethers, is uncomfortable and increases the “time-to-interface,” or setup time. Conversely, computer-vision-based interfaces offer unencumbered interaction, providing several notable advantages:

- ▶ Computer vision is nonintrusive;
- ▶ Sensing is passive, silent, possibly stealthy;
- ▶ Installed camera systems can perform other tasks aside from hand-gesture interfaces; and
- ▶ Sensing and processing hardware is commercially available at low cost.

However, vision-based systems usually require application-specific algorithm development, programming, and machine learning. Deploying them in everyday environments is a challenge, particularly for achieving the robustness necessary for user-interface acceptability: robustness for camera sensor and lens characteristics, scene and background details, lighting conditions, and user differences. Here, we look at methods employed in systems that have overcome these difficulties, first discussing feature-extraction methods (aimed at gaining information about gesture position, orientation, posture, and temporal progression), then briefly covering popular approaches to feature classification (see Figure 1).



Figure 1. Head and hand detection using depth from stereo, illumination-specific color segmentation, and knowledge of typical body characteristics.<sup>17</sup>

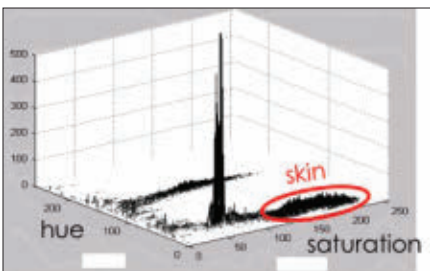


Figure 2. Hue-Saturation histogram of skin color. The circled region contains the hand pixels in the photo; the high spike is caused by grayish and white pixels.

**Motion.** Frame-to-frame comparison against a learned background model is an effective and computationally efficient method for finding foreground objects and for observing their position and movement. This comparison requires several assumptions (such as a stationary camera or image pre-processing to stabilize the video) and a static background; for example, Kang et al.<sup>21</sup> employed the Lucas-Kanade tracking method.<sup>29</sup>

**Depth.** Range data from a calibrated camera pair<sup>40</sup> or direct range sensors (such as LiDAR) is a particularly useful cue if the user is expected to face the camera(s) and the hands are considered the closest object. Depth from stereo is usually coarse-grain and rather noisy, so it is often combined with other image cues (such as color<sup>17,22,33</sup>). Well-calibrated stereo cameras are costly, and depth can be calculated accurately only if the scene contains sufficient texture. If texture is lacking, artificial texture can be projected into the scene through a digital light projector injecting structured light patterns.<sup>39</sup>

**Color.** Heads and hands are found with reasonable accuracy based purely on their color.<sup>24,40</sup> Skin color occupies a rather well-defined area in color spaces (such as Hue, Saturation, and Intensity,  $L^*a^*b^*$ , and YIQ) so can be used for segmentation (see Figure 2 and Hasanuzzaman et al.,<sup>19</sup> Rogalla et al.,<sup>41</sup> and Yin and Zhu<sup>53</sup>). Combined histogram-matching and blob-tracking with Camshift<sup>7</sup> or the Viterbi algorithm<sup>54</sup> is a popular approach due to its speed, ease of implementation, and performance. Shortcomings stem from confusion with similar-colored objects in the background and limitations with respect to posture recognition. Better optics and sensors often improve color saturation, therefore color-based algorithms; another accuracy boost can be achieved through user-worn markers (such as colored gloves and bright LEDs). While simplifying the interface implementation, these aids do not permit users to “come as you are,” so IR illumination can be used instead of markers. The IR light source illuminates users’ hands, allowing an IR camera to capture the images of the illuminated parts.<sup>44</sup> In addition, reflective material affixed to a body part can increase the part’s re-

flection properties.

**Shape.** Many objects can be distinguished by their shape, or silhouette. Different object orientations are often also revealed based on shape alone. Shape is available if the object is clearly segmented from the background scenery, achievable in controlled environments (such as with chroma keying), often for stationary-camera systems (using a background model) and a bit less reliably with a good hand-color model.<sup>53</sup> Popular methods include statistical moments,<sup>13</sup> rule-based methods (see Figure 3 and Kawarazaki et al.<sup>22</sup> and Yin and Zhu<sup>53</sup>), active shape models,<sup>12</sup> and shape context.<sup>4</sup>

**Appearance.** Methods that consider the intensity and/or color values across a region of interest are more powerful and robust than methods that consider shape alone. Since they do not rely on segmentation, they are generally able to handle situations with no intensity/color distinction between foreground and background. The theoretical upper bound on lexicon size is much greater for appearance-based methods than for purely depth- and shape-based methods. The drawback is increased computational cost during training and recognition; for example, detecting heads in all possible orientations or hands in all possible configurations is not currently possible at interactive frame rates. Examples of appearance-based methods (such as by Viola and Jones<sup>49</sup>) have been employed for various vision-based interfaces, including those reported by Hasanuzzaman.<sup>19</sup>

**Multi-cue.** Rather than rely on a single image cue, a number of schemes combine information from multiple cues. Motion-based region-of-interest designation, combined with appearance-based hand, face, or body detection, improves speed and accuracy. Appearance and color for detection and motion cues, together with color, were used for hand-gesture interfaces (see Figure 4) by Kölsch et al.<sup>24</sup> and Rauschert et al.<sup>40</sup> Removal of any of these cues was shown to hurt performance. Methods that segment a gesture in an image based on color, then classify the shape, do not fall into this multi-cue category, since the cues are used sequentially, not cooperatively; that is, if the first cue fails, the second cue is useless. True multi-cue systems

face similar difficulties with data combination as classic sensor fusion: intra-cue confidence is often unavailable for weighting; the data domains and/or ranges are often distinct; and the combination function may be highly non-linear.

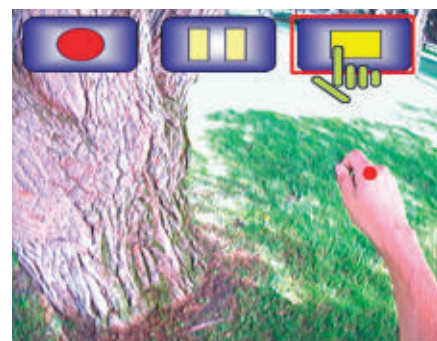
The extracted features are then subjected to various classifiers, from generic support vector machines<sup>10</sup> to highly customized shape classifiers, as in Yin and Zhu.<sup>53</sup> Some features perform classification implicitly; for example, the Lucas-Kanade-based tracker discards “unreliable” patches, and Camshift<sup>7</sup> determines a decision boundary in space and color histograms.

Classification is sometimes externally combined with feature extraction, as in the boosting approach involving a combination of weak detectors.<sup>49</sup> Other methods involve a distinct translation step into feature space and subsequent classification; for example, consider the motion track of a hand gesture, with its spatial location over time serving as feature vector and a hidden Markov model classifying hand trajectory into various temporal/dynamic gestures<sup>26,33,40,42</sup> (see Figure 5).

As with speech recognition, dy-



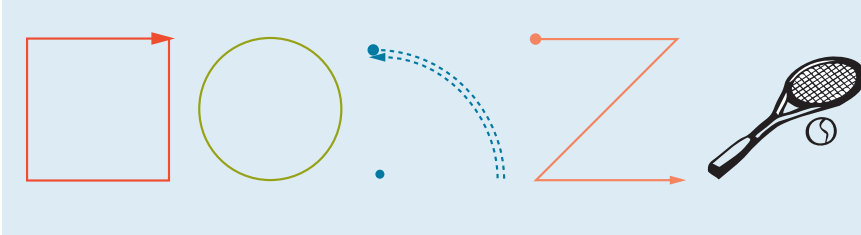
**Figure 3.** Hand-gesture recognition using color segmentation, conversion into polar coordinates, and maxima detection to identify and count fingers.



**Figure 4.** Multi-cue hand tracking and posture recognition.<sup>24</sup>



**Figure 5.** Motions reliably distinguished by hidden Markov models,<sup>42</sup> from moving the WiiMote in a square to swinging an arm, as in serving with a tennis racquet. The third gesture from the left describes a 90-degree roll angle around the z-axis (back and forth).



dynamic gesture segmentation (when the gesture starts and ends) is a challenge; in gesture research, such temporal segmentation is often called “gesture spotting” (see the section on requirements and challenges). Spotting is not only difficult but necessitates a lag between gesture start and finish (or later), limiting the responsiveness of the user interface. Other successful classification methods are dynamic time warping, Hough transforms, mean-shift and Camshift, and Bayesian approaches.

### Applications

The first application of hand-gesture control we review—medical systems and assistive technologies—provides the user sterility needed to help avoid the spread of infection. The second—entertainment—involves naturalness of the interface as part of the user experience. The next—crisis management and disaster relief—involves a per-

formed task requiring quick user feedback. Finally, human-robot interaction must be natural and intuitive for the personal robot of the future. Here, we cover hand-gesture control interfaces for each category and discuss the related design considerations.

**Medical systems and assistive technologies.** Gestures can be used to control the distribution of resources in hospitals, interact with medical instrumentation, control visualization displays, and help handicapped users as part of their rehabilitation therapy.<sup>35,50</sup> Some of these concepts have been exploited to improve medical procedures and systems; for example, Face MOUSE<sup>35</sup> satisfied the “come as you are” requirement, where surgeons control the motion of a laparoscope by making appropriate facial gestures without hand or foot switches or voice input. Graetzel et al.<sup>16</sup> covered ways to incorporate hand gestures into doc-

tor-computer interfaces, describing a computer-vision system that enables surgeons to perform standard mouse functions, including pointer movement and button presses, with hand gestures that satisfy the “intuitiveness” requirement. Wachs et al.<sup>50</sup> developed a hand-gesture-tracking device called Gestix that allows surgeons to browse MRI images in an operating room (see Figure 6), using a natural interface to satisfy both “come as you are” and “intuitiveness.”

A European Community Project called WearIT@work<sup>30</sup> satisfies the “comfort” requirement by encouraging physicians to use a wrist-mounted RFID reader to identify the patient and interact through gestures with the hospital information system to document exams and write prescriptions, helping ensure sterility. However, since this is an encumbered interface, the “come as you are” requirement is violated. We expect to see some of these new technologies (based on “smart instruments”) introduced directly into the operating room, where the direction/activation of a robotic end effector could be performed through gesture recognition.<sup>35</sup>

For the impaired, the critical requirements of a hand-gesture interface system are “user adaptability and feedback” and “come as you are.” In this context, wheelchairs, as mobility aids, have been enhanced through robotic/intelligent vehicles able to recognize hand-gesture commands (such as in Kuno et al.<sup>28</sup>). The Gesture Pendant<sup>44</sup> is a wearable gesture-recognition system used to control home devices and provide additional functionality as a medical diagnostic tool. The Staying Alive<sup>3</sup> virtual-reality-imagery-and-relaxation tool satisfies the “user adaptability and feedback” requirement, allowing cancer patients to navigate through a virtual scene using 18 traditional T'ai Chi gestures. In the same vein, a tele-rehabilitation system<sup>18</sup> for kinesthetic therapy—treatment of patients with arm-motion coordination disorders—uses force-feedback of patient gestures. Force-feedback was also used by Patel and Roy<sup>36</sup> to guide an attachable interface for individuals with severely dysarthric speech. Also, a hand-worn haptic glove was used to help rehabilitate post-stroke patients in the chronic phase by Boian et al.<sup>5</sup> These systems



**Figure 6.** Surgeon using Gestix to browse medical images.




illustrate how medical systems and rehabilitative procedures promise to provide a rich environment for the potential exploitation of hand-gesture systems. Still, additional research and evaluation procedures are needed to encourage system adoption.


**Entertainment.** Computer games are a particularly technologically promising and commercially rewarding arena for innovative interfaces due to the entertaining nature of the interaction. Users are eager to try new interface paradigms since they are likely immersed in a challenging game-like environment.<sup>45</sup> In a multi-touch device, control is delivered through the user's fingertips. Which finger touches the screen is irrelevant; most important is where the touch is made and the number of fingers used.

In computer-vision-based, hand-gesture-controlled games,<sup>13</sup> the system must respond quickly to user gestures, the “fast-response” requirement. In games, computer-vision algorithms must be robust and efficient, as opposed to applications (such as inspection systems) with no real-time requirement, and where recognition performance is the highest priority. Research efforts should thus focus on tracking and gesture/posture recognition with high-frame-rate image processing (>10 fps).

Another challenge is “gesture spotting and immersion syndrome,” aiming to distinguish useful gestures from unintentional movement. One approach is to select a particular gesture to mark the “start” of a sequence of gestures, as in the “push to talk” approach in radio-based communication where users press a button to start talking. In touchscreen mobile phones, the user evokes a “swipe” gesture to start operating the device. To “end” the interaction, the user may evoke the “ending” gesture or just “rest” the hands on the side of the body. This multi-gesture routine may be preferable to purely gaze-based interaction where signaling the end of the interaction is a difficult problem, since users cannot turn off their eyes. The problem of discriminating between intentional gestures and unintentional movement is also known as the Midas Touch problem (<http://www.diku.dk/hjemmesider/ansatte/panic/eyegaze/node27.html>).



**For sign languages (such as American Sign Language), hand-gesture-recognition systems must be able to recognize a large lexicon of single-handed and two-handed gestures.**



In the Mind-Warping augmented-reality fighting game,<sup>45</sup> where users interact with virtual opponents through hand gestures, gesture spotting is solved through voice recognition. The start and end of a temporal gesture is “marked” by voice—the start and end of a Kung Fu yell; Kanget al.<sup>21</sup> addressed the problem of gesture spotting in the first-person-shooter *Quake II*. Such games use contextual information like gesture velocity and curvature to extract meaningful gestures from a video sequence. Bannach et al.<sup>41</sup> addressed gesture spotting through a sliding window and bottom-up approach in a mixed-reality parking game. Schlömer et al.<sup>42</sup> addressed accelerometer-based gesture recognition for drawing and browsing operations in a computer game. Gesture spotting in many Nintendo Wii games is overcome by pressing a button on the WiiMote control through the “push to talk” analogy.

Intuitiveness is another important requirement in entertainment systems. In the commercial arena, most Nintendo Wii games are designed to mimic actual human motions in sports games (such as golf, tennis, and bowling). Wii games easily meet the requirement of “intuitiveness,” even as they violate the “come as you are” requirement, since users must hold the WiiMote, instead of using a bare hand. Sony's EyeToy for the Playstation and the Kinect sensor for Microsoft's Xbox360 overcome this limitation while achieving the same level of immersion through natural gestures for interaction. These interfaces use hand-body gesture recognition (also voice recognition in Kinect) to augment the gaming experience.

In the research arena, the intuitive aspect of hand-gesture vocabulary is addressed in a children's action game called QuiQui's Giant Bounce<sup>20</sup> where control gestures are selected through a “Wizard of Oz” paradigm in which a player interacts with a computer application controlled by an unseen subject, with five full-body gestures detected through a low-cost USB Web camera.


“User adaptability and feedback” is the most remarkable requirement addressed in these applications. In entertainment systems, users profit from having to learn the gesture vocabularies employed by the games. A

training session is usually required to teach them how the gestures should be performed, including speed, trajectory, finger configuration, and body posture. While beginners need time to learn the gesture-related functions, experienced users navigate through the games at least as quickly as if they were using a mechanical-control device or attached sensors.<sup>8,37</sup>


Intelligent user interfaces that rely on hand/body gesture technology face special challenges that must be addressed before future commercial systems are able to gain popularity. Aside from technical obstacles like reliability, speed, and low-cost implementation, hand-gesture interaction must also address intuitiveness and gesture spotting.

**Crisis management and disaster relief.** Command-and-control systems help manage public response to natural disasters (such as tornados, floods, wildfires, and epidemic diseases) and to human-caused disasters (such as terrorist attacks and toxic spills). In them, the emergency response must be planned and coordinated by teams of experts with access to large volumes of complex data, in most cases through traditional human-computer interfaces. One such system, the “Command Post of the Future,”<sup>47</sup> uses pen-based gestures.<sup>11</sup> Such hand-gesture interface systems must reflect the requirements of “fast learning,” “intuitiveness,” “lexicon size and number of hands,” and “interaction space” to achieve satisfactory performance.<sup>26</sup> The first two involve natural interaction with geospatial information (easy to remember and common gestures); the last two involve the system’s support of collaborative decision making among individuals. Multimodality (speech and gesture), an additional requirement for crisis-management systems, is not part of our original list of requirements since it includes modalities other than gestures. The pioneering work was Richard A. Bolt’s “Put-That-There” system,<sup>6</sup> providing multimodal voice input plus gesture to manipulate objects on a large display.

DAVE\_G,<sup>40</sup> a multimodal, multi-user geographical information system (GIS), has an interface that supports decision making based on geospatial data to be shown on a large-screen dis-



**Aside from technical obstacles like reliability, speed, and low-cost implementation, hand-gesture interaction must also address intuitiveness and gesture spotting.**



play. Potential users are detected as soon as they enter the room (the “come as you are” requirement) through a face-detection algorithm; the detected facial region helps create a skin-color model applied to images to help track the hands and face. Motion cues are combined with color information to increase the robustness of the tracking module. Spatial information is conveyed using “here” and “there” manipulative gestures that are, in turn, recognized through a hidden Markov model. The system was extended to operate with multiple users in the “XISM” system at Pennsylvania State University<sup>26</sup> where users simultaneously interface with the GIS, allowing a realistic decision-making process; however, Krahnstoeber et al.<sup>26</sup> provided no detail as to how the system disambiguates tracking information of the different users.

Other approaches to multi-user hand-gesture interfaces have adopted multi-touch control through off-the-shelf technology,<sup>15,31</sup> allowing designers to focus on collaborative user performance rather than on hand-gesture-recognition algorithms. These systems give multiple users a rich hand-gesture vocabulary for image manipulation, including zoom, pan, line drawing, and defining regions of interest, satisfying the “lexicon size and number of hands” requirement. Spatial information about objects on the GIS can be obtained by clicking (touching) the appropriate object.

These applications combine collaborative hand-gesture interaction with large visual displays. Their main advantage is user-to-user communication, rather than human-computer interaction, so the subjects use their usual gestures without having to learn new vocabularies; for example, sweeping the desk can be used to clean the surface.

**Human-robot interaction.** Hand-gesture recognition is a critical aspect of fixed and mobile robots, as suggested by Kortenkamp et al.<sup>25</sup> Most important, gestures can be combined with voice commands to improve robustness or provide redundancy and deal with “gesture spotting.” Second, hand gestures involve valuable geometric properties for navigational robot tasks; for example, the pointing gesture can symbolize the “go there” command for

mobile robots. For a robotic arm, human users may use the “put it there” command while pointing to the object and then the place. Hand actions can be used to manipulate operations (such as grasp and release), since a human hand is able to simulate the form of the robot gripper. All these aspects of robot interaction help satisfy the “intuitiveness” requirement. Third, people with physical handicaps are able to control robots through gestures when other channels of interaction are limited or impossible without special keyboards and teach-pendants, or robot controls, satisfying the “come as you are” requirement. Fourth, such an interface brings operability to beginners who find it difficult to use sophisticated controls to command robots. Hand-gesture control of robots faces several constraints specific to this category of interfaces, including “fast,” “intuitive,” “accuracy,” “interaction space,” and “reconfigurability.” While most systems succeed to some extent in overcoming the technical requirements (“accuracy”), the interaction aspects of these systems involve many unsolved challenges.

Using stereo vision to develop a cooperative work system, Kawarazaki<sup>22</sup> combined robotic manipulators and human users with hand-gesture instructions to recognize four static gestures; when users point at an object on a table with their forefinger the robot must be able to detect it. Chen and Tseng<sup>10</sup> described human-robot interaction for game playing in which three static gestures at multiple angles and scales are recognized by a computer-vision algorithm with 95% accuracy, satisfying the “accuracy” requirement.

Using Sony’s AIBO entertainment robot, Hasanuzzaman<sup>19</sup> achieved interaction by combining eight hand gestures and face detection to identify two nodding gestures and the hand (left or right) being used, allowing for a larger lexicon than hand gestures alone.

Rogalla et al.<sup>41</sup> developed a robotic-assistant interaction system using both gesture recognition and voice that first tracks gestures, then combines voice and gesture recognition to evoke a command. Once the hand is segmented, six gestures are trained using a hand contour as the main feature of each gesture. Since the user

and robot interact with objects on a table, the interaction space is large enough to include both user and objects. Rogella et al.<sup>41</sup> reported 95.9% recognition accuracy.

Nickel and Stiefelhagen<sup>33</sup> developed a system that recognizes dynamic pointing gestures that rely on head and arm orientation for human-robot interaction. The system uses a hidden Markov model to recognize trajectories of the segmented hands and up to 210 gestures, satisfying the requirement of “lexicon size and number of hands.”

Yin and Zhu<sup>53</sup> implemented a programming-by-demonstration approach in which the robot learns gestures from a human user (the instructor), satisfying the requirement of “user adaptability and feedback.” The system uses eight static gestures to control a hybrid service robot system called HARO-1. Calinon and Billard<sup>9</sup> also used a programming-by-demonstration paradigm, allowing users to help the robot reproduce a gesture through kinesthetic teaching; in it, the user teaches the robot 10 dynamic gestures acquired through sensors attached to the torso and upper and lower arm, hence violating the “come as you are” and “comfort” requirements.

Most approaches we’ve reviewed here employ a stereo camera to acquire hand gestures. Some systems also add voice detection, thereby solving the “gesture spotting” problem and improving recognition accuracy. Most of them detect static hand gestures but are not robust enough to recognize more than 10 gestures, so do not satisfy the requirement of “lexicon size and number of hands.” Two-handed dynamic-gesture multimodal interaction is thus a promising area for future research.

## Conclusion

Hand-gesture implementation involves significant usability challenges, including fast response time, high recognition accuracy, quick to learn, and user satisfaction, helping explain why few vision-based gesture systems have matured beyond prototypes or made it to the commercial market for human-computer devices. Nevertheless, multi-touchscreens and non-joystick and -keyboard interaction methods have found a home in the game-console

market, commercial appeal suggesting that hand-gesture-based interactive applications could yet become important players in next-generation interface systems due to their ease of access and naturalness of control.

Four recommended guidelines help evaluate future hand-gesture interfaces to increase the likelihood of their widespread commercial/social acceptance:

**Validation.** Rigorous statistical validation procedures for gesture-based systems on public, standard test sets. A system’s performance can be demonstrated through several statistical measures<sup>32</sup>: sensitivity/recall, precision/positive predictive value, specificity, negative predictive value, f-measure, likelihood ratio, and accuracy;

**User independence.** User independence while permitting customizability enhances acceptability;

**Usability criteria.** Provide usability criteria to evaluate learnability, efficiency, ease of remembering, likelihood of errors, and user satisfaction; performance can be evaluated through task completion time and subjective workload assessment through, say, the NASA Task Load Index (<http://human-systems.arc.nasa.gov/groups/TLX/>) and the Subjective Workload Assessment Technique<sup>38</sup>; and

**Qualitative/quantitative assessment.** Provide qualitative and quantitative assessments of this modality compared to other modalities (such as voice recognition); for example, user performance when using alternative modalities can be compared with the metrics outlined in the guideline concerning usability criteria.

**Questions.** Reviewing the HCI literature as we wrote this article revealed increasing adoption of certain principles and heuristics that contribute to the design of hand-gesture-recognition systems:

**Context support in hand-gesture recognition.** Gestures are context-dependent. Gestures and their types and uses are determined by the context in which they are applied. Task domain analysis helps identify users’ intended actions, goals, and means. Previously, HCI researchers adopted task analysis to help determine suitable features for natural HCI.<sup>26,40</sup>

The trade-off between increasing the number of gestures to be recog-




nized and the performance of the recognition is a well-known obstacle in the design of gesture-based interfaces. The more freely a system allows users to express themselves, the less accurate it gets; conversely, the greater the rigor in specifying gestures, the greater the likelihood the system will perform accurately.


A common approach toward achieving this trade-off is to create a set of specific grammars or vocabularies for different contexts. The system dynamically activates different subsets of vocabularies and grammars according to the context, instead of maintaining a single large lexicon. This built-in feature reduces complexity in gesture-recognition systems, as separate gesture-recognition algorithms are used for smaller gesture subsets. Context is captured in many ways, including hand position, interaction log, task, type of gesture, and how the user interacts with devices in the environment.

*Methods for hand-gesture recognition.* No single algorithm for hand-gesture recognition favors every application. The suitability of each approach depends on application, domain, and physical environment. Nevertheless, integration of multiple methods lends robustness to hand-tracking algorithms; for example, when a tracker loses track of a hand due to occlusion, a different tracker using a different tracking paradigm can still be active. Occlusion is usually disambiguated through the stereo cameras to create depth maps of the environment. Common approaches for hand-gesture tracking use color and motion cues. Human skin color is distinctive and serves to distinguish the human face and hand from other objects. Trackers sensitive to skin color and motion can achieve a high degree of robustness.<sup>40</sup>

Regarding classification, gestures are the outcome of stochastic processes. Thus, defining discrete representations for patterns of spatio-temporal gesture motion is a complicated process. Gesture templates can be determined by clustering gesture training sets to produce classification methods with accurate recognition performance; Kang et al.<sup>21</sup> described examples of such methods, including hidden Markov models, dynamic time warping, and finite state machines.




## Two-handed dynamic-gesture multimodal interaction is thus a promising area for future research.



Finally, Kang et al.<sup>21</sup> also addressed the problem of gesture spotting through sliding windows, distinguishing intentional gestures from captured gestures through recognition accuracy of the observed gestures.

*Intuitive gestures (selection and teaching) in interface design.* Ideally, gestures in HCI should be intuitive and spontaneous. Psycholinguistics and cognitive sciences have produced a significant body of work involving human-to-human communication that can help find intuitive means of interaction for HCI systems. A widely accepted solution for identifying intuitive gestures was suggested by Baudel et al.,<sup>2</sup> and in Höysniemi et al.'s "Wizard-of-Oz" experiment, an external observer interprets user hand movement and simulates the system's response.<sup>20</sup> Called "teaching by demonstration," it is widely used for gesture learning. Rather than pick the gestures during the design stage of the interface, they are selected during real-time operation while interacting with the user, thus mimicking the process of parents teaching gestures to a toddler.<sup>9</sup> First, the parents show the toddler a gesture, then assist the toddler to imitate the gesture by moving the toddler's own hands. The toddler learns the skill of producing the gesture by focusing on his or her own active body parts. Hand gestures play an important role in human-human communication. Analysis of these gestures based on experimental sociology and learning methodologies will lead to more robust, natural, intuitive interfaces.

### Acknowledgments

This research was performed while the first author held a National Research Council Research Associateship Award at the Naval Postgraduate School, Monterey, CA. It was partially supported by the Paul Ivanier Center for Robotics Research & Production Management at Ben-Gurion University of the Negev. 

### References

1. Bannach, D., Amft, O., Kunze, K.S., Heinz, E.A., Tröster, G., and Lukowicz, P. Waving real-hand gestures recorded by wearable motion sensors to a virtual car and driver in a mixed-reality parking game. In *Proceedings of the Second IEEE Symposium on Computational Intelligence and Games* (Honolulu, Apr. 1–5, 2007), 32–39.
2. Baudel, T. and Beaudouin-Lafon, M. Charade: Remote control of objects using FreeHand gestures. *Commun. ACM* 36, 7 (July 1993), 28–35.

3. Becker, D.A. and Pentland, T. Staying alive: A virtual reality visualization tool for cancer patients. In *Proceedings of the AAAI Workshop on Entertainment and Alife/AI*. AAAI Technical Report WS-96-03, 1996.
4. Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape contexts. *Transactions on Pattern Analysis and Machine Intelligence* 24, 24 (Apr. 2002), 509–522.
5. Boian, R., Sharma, R., Han, C., Merians, A., Burdea, G., Adamovich, S., Recce, M., Tremaine, M., and Poizner, H. Virtual reality-based post-stroke hand rehabilitation. *Studies in Health and Technology Information* (2002), 64–70.
6. Bolt, R.A. 'Put-That-There': Voice and gesture at the graphics interface. In *Proceedings of the Seventh International Conference on Computer Graphics and Interactive Techniques*. ACM Press, New York, 1980, 262–270.
7. Bradski, G.R. Computer-vision face tracking for use in a perceptual user interface. *Intel Technology Journal* (Q2 1998).
8. Brashear, H., Henderson, V., Park, K., Hamilton, H., Lee, S., and Starner, T. American Sign Language recognition in game development for deaf children. In *Proceedings of ACM SIGACCESS Conference on Assistive Technologies* (Portland, OR, Oct. 23–25). ACM Press, New York, 2006, 79–86.
9. Calinon, S. and Billard, A. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction* (Arlington, VA, 2007), 255–262.
10. Chen, Y.T. and Tseng, K.T. Developing a multiple-angle hand-gesture-recognition system for human-machine interactions. In *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society* (Taipei, Nov. 5–8, 2007), 489–492.
11. Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. QuickSet: Multimodal interaction for distributed applications. In *Proceedings of the Fifth ACM international Conference on Multimedia* (Seattle, WA, Nov. 9–13). ACM Press, New York, 1997, 10–13.
12. Cootes, T.F. and Taylor, C.J. Active shape models: 'smart snakes'. In *Proceedings of the British Machine-Vision Conference* (Leeds, Sept. 22–24). Springer, Berlin, 1992, 266–275.
13. Freeman, W. and Roth, M. Orientation histograms for hand-gesture recognition. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition* (Zurich, June 1995).
14. Freeman, W.T., Tanaka, K., Ohta, J., and Kyuma, K. Computer vision for computer games. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition* (Zurich, June 1995), 296–301.
15. Fruijtjer, S., Dulk, P.D., and Dias, E. Collaborative interaction and integrated spatial information and services in disaster management. In *Proceedings of the 2008 IEEE International Workshop on Horizontal Interactive Human Computer System* (Amsterdam, Oct. 1–3, 2008), 43–45.
16. Graetzl, C., Fong, T.W., Grange, C., and Baur, C. A non-contact mouse for surgeon-computer interaction. *Technology and Health Care* 12, 3 (Aug. 24, 2004), 245–257.
17. Grange, S., Fong, T., and Baur, C. M/ORIS: A medical/operating room interaction system. In *Proceedings of the ACM International Conference on Multimodal Interfaces* (State College, PA, 2004). ACM Press, New York, 2004, 159–166.
18. Gutierrez, M., Lemoine, P., Thalmann, D., and Vexo, F. Telerehabilitation: Controlling haptic virtual environments through handheld interfaces. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology* (Hong Kong, Nov. 10–12), ACM Press, New York, 2004, 195–200.
19. Hasanuzzaman, M., Ampornaramveth, V., Zhang, T., Bhuiyan, M.A., Shirai, Y., and Ueno, H. Real-time vision-based gesture recognition for human-robot interaction. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics* (Shenyang, China, Aug. 22–26, 2004), 413–418.
20. Höysniemi, J., Hämmäläinen, P., Turkki, L., and Rouvi, T. Children's intuitive gestures in vision-based action games. *Commun. ACM* 48, 1 (Jan. 2005), 44–50.
21. Kang, H., Lee, C., and Jung, K. Recognition-based gesture spotting in video games. *Pattern Recognition Letters* 25, 15 (Nov. 2004), 1701–1714.
22. Kawarazaki, N., Hoya, I., Nishihara, K., and Yoshidome, T. Cooperative welfare robot system using hand-gesture instructions. *Lecture Notes in Control and Information Sciences* 306, Springer, Berlin, 2004, 143–153.
23. Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., and Marca, D. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing* 10, 5 (July 2006), 285–299.
24. Kölsch, M., Turk, M., and Höllerer, T. Vision-based interfaces for mobility. In *Proceedings of the International Conference on Mobile and Ubiquitous Systems* (Boston, Aug. 22–26, 2004), 86–94.
25. Kortenkamp, D., Huber, E., and Bonasso, R. Recognizing and interpreting gestures on a mobile robot. In *Proceedings of the 13th Conference on Artificial Intelligence* (Portland, OR, Aug. 4–8, 1996), 915–921.
26. Krahnstoeber, N., Kettebekov, S., Yasini, M., and Sharma, R. A real-time framework for natural multimodal interaction with large-screen displays. In *Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces* (Pittsburgh, PA, Oct. 14–16). IEEE Computer Society, Washington, D.C., 2002, 349.
27. Krueger, M.W. *Artificial Reality, Second Ed.* Addison-Wesley, Redwood City, CA, 1991.
28. Kuno, Y., Murashima, T., Shimada, N., and Shirai, Y. Intelligent wheelchair remotely controlled by interactive gestures. In *Proceedings of 15th International Conference on Pattern Recognition* (Barcelona, Sept. 3–7, 2000), 672–675.
29. Lucas, B.D. and Kanade, T. An iterative image-registration technique with an application to stereo vision. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., San Francisco, 1981, 674–679.
30. Lukowicz, P., Timm-Giel, A., Lawo, M., and Herzog, O. WearIT@Work: Toward real-world industrial wearable computing. *IEEE Pervasive Computing* 6, 4 (Oct. 2007), 8–13.
31. Micire, M.J. and Yanco, H.A. Improving disaster response with multi-touch technologies. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, Oct. 29–Nov. 2, 2007), 2567–2568.
32. Minnen, D., Westeyn, T., and Starner, T. Performance metrics and evaluation issues for continuous activity recognition. In *Proceedings of Performance Metrics in Intelligent Systems Workshop* (Aug. 21–23). NIST, Gaithersburg, MD, 2008.
33. Nickel, K. and Stiefelhagen, R. Visual recognition of pointing gestures for human-robot interaction. *Image and Vision Computing* 25, 12 (Dec. 2007), 1875–1884.
34. Nielsen, M., Starring, M., Moeslund, T.B., and Granum, E. *A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for Man-Machine Interaction*. Technical Report CVMT 03-01. Aalborg University, Aalborg, Denmark, Mar. 2003.
35. Nishikawa, A., Hosoi, T., Koara, K., Negoro, D., Hikita, A., Asano, S., Kakutani, H., Miyazaki, F., Sekimoto, M., Yasui, M., Miyake, Y., Takiguchi, S., and Monden, M. FAcE MOUSE: A novel human-machine interface for controlling the position of a laparoscope. *IEEE Transactions on Robotics and Automation* 19, 5 (Oct. 2003), 825–841.
36. Patel, R. and Roy, D. Teachable interfaces for individuals with dysarthric speech and severe physical disabilities. In *Proceedings of the AAAI Workshop on Integrating Artificial Intelligence and Assistive Technology* (Madison, WI, July 26–30, 1998), 40–47.
37. Pentland, A. and Becker, D. *Sensei: A Real-Time Recognition, Feedback, and Training System for Tai Chi Gestures*. Masters Thesis. Harvard University, Cambridge, MA, 1997.
38. Potter, S.S. and Bressler, J.R. *Subjective Workload Assessment Technique (SWAT): A User's Guide*. Interim Report, 1998.
39. Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of SIGGRAPH* (Orlando, FL, July 19–24). ACM Press, New York, 1998, 179–188.
40. Rauschert, I., Agrawal, P., Sharma, R., Fuhrmann, S., Brewer, I., and MacEachren, A.M. Designing a human-centered, multimodal GIS interface to support emergency management. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems* (McLean, VA, Nov. 8–9). ACM Press, New York, 2002, 119–124.
41. Rogalla, O., Ehrenmann, M., Zöllner, R., Becher, R., and Dillmann, R. Using gesture and speech control for commanding a robot assistant. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication* (Berlin, Sept. 25–27, 2002), 454–459.
42. Schlömer, T., Poppinga, B., Henze, N., and Boll, S. Gesture recognition with a Wii controller. In *Proceedings of the Second International Conference on Tangible and Embedded Interaction* (Bonn, Germany, Feb. 18–20). ACM Press, New York, 2008, 11–14.
43. Sheridan, T. and Ferrell, W. Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics* 4, 1 (1963), 25–29.
44. Starner, T., Auxier, J., Ashbrook, D., and Gandy, M. The gesture pendant: A self-illuminating, wearable, infrared computer-vision system for home-automation control and medical monitoring. In *Proceedings of the Fourth International Symposium on Wearable Computers* (Atlanta, Oct. 2000), 87–94.
45. Starner, T., Leibe, B., Singletary, B., and Pair, J. Mind-warping: Towards creating a compelling collaborative augmented reality game. In *Proceedings of the Fifth International Conference on Intelligent User Interfaces* (New Orleans, Jan. 9–12). ACM Press, New York, 2000, 256–259.
46. Stern, H.I., Wachs, J.P., and Edan, Y. Designing hand-gesture vocabularies for natural interaction by combining psycho-physiological and recognition factors (special issue on gesture in multimodal systems). *International Journal of Semantic Computing* 2, 1 (Mar. 2008), 137–160.
47. Thomas, J.J. and Cook, K.A., Eds. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE CS Press, 2005.
48. Triesch, J. and Malsburg, C.V.D. Robotic gesture recognition by cue combination. *Gesture and Sign Language in Human-Computer Interaction*. Lecture Notes in Computer Science. Springer, Berlin, 1998, 233–244.
49. Viola, P. and Jones, M. Robust real-time object detection. *International Journal of Computer Vision* 57, 2 (May 2004), 137–154.
50. Wachs, J., Stern, H., Edan, Y., Gillam, M., Feied, C., Smith, M., and Handler, J. A hand-gesture sterile tool for browsing MRI images in the OR. *Journal of the American Medical Informatics Association* 15, 3 (May–June 2008), 321–323.
51. Wachs, J.P. *Optimal Hand-Gesture Vocabulary Design Methodology for Virtual Robotic Control*. Ph.D. Thesis, Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er She'eva, Israel, 2007.
52. Ward, J.A., Lukowicz, P., Troster, G., and Starner, T.E. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 10 (Oct. 2006), 1553–1567.
53. Yin, X. and Zhu, X. Hand-posture recognition in gesture-based human-robot interaction. In *Proceedings of the IEEE Conference on Industrial Electronics and Applications* (Singapore, May 24–26, 2006), 1–6.
54. Yuan, Q., Sclaroff, S., and Athitsos, V. Automatic 2D Hand tracking in video sequences. In *Proceedings of the IEEE Workshop on Applications of Computer Vision* (Breckenridge, CO, Jan. 5–7). IEEE Computer Society Press, 2005, 250–256.

**Juan Pablo Wachs** (jpwachs@purdue.edu) is an assistant professor in the School of Industrial Engineering of Purdue University, West Lafayette, IN.

**Mathias Kölsch** (kolsch@nps.edu) is an assistant professor in the Computer Science Department and Modeling, Virtual Environments, and Simulation Institute of the Naval Postgraduate School, Monterey, CA.

**Helman Stern** (helman@bgu.ac.il) is Professor Emeritus in the Department of Industrial Engineering and Management of Ben-Gurion University of the Negev, Be'er She'eva, Israel.

**Yael Edan** (yael@bgu.ac.il) is a professor in the Department of Industrial Engineering and Management of Ben-Gurion University of the Negev, Be'er She'eva, Israel.

DOI:10.1145/1897816.1897839

**Google's WebTables and Deep Web Crawler identify and deliver this otherwise inaccessible resource directly to end users.**

**BY MICHAEL J. CAFARELLA, ALON HALEVY,  
AND JAYANT MADHAVAN**

# Structured Data on the Web

THOUGH THE WEB is best known as a vast repository of shared documents, it also contains a significant amount of structured data covering a complete range of topics, from product to financial, public-record, scientific, hobby-related, and government. Structured data on the Web shares many similarities with the kind of data traditionally managed by commercial database systems but also reflects some unusual characteristics of its own; for example, it is embedded in textual Web pages and must be extracted prior to use; there is no centralized data design as there is in a traditional database; and, unlike traditional databases that focus on a single domain, it covers everything. Existing data-management systems do not address these challenges and assume their data is modeled within a well-defined domain.

This article discusses the nature of Web-embedded structured data and the challenges of managing it. To begin, we present two relevant research projects

developed at Google over the past five years. The first, WebTables, compiles a huge collection of databases by crawling the Web to find small relational databases expressed using the HTML table tag. By performing data mining on the resulting extracted information, WebTables is able to introduce new data-centric applications (such as schema completion and synonym finding). The second, the Google Deep Web Crawler, attempts to surface information from the Deep Web, referring to data on the Web available only by filling out Web forms, so cannot be crawled by traditional crawlers. We describe how this data is crawled by automatically submitting relevant queries to a vast number of Web forms. The two projects are just the first steps toward exposing and managing structured Web data largely ignored by Web search engines.

## Web Data

Structured data on the Web exists in several forms, including HTML tables, HTML lists, and back-end Deep Web databases (such as the books sold on Amazon.com). We estimate in excess of one billion data sets as of February 2011. More than 150 million sources come from a subset of all English-language HTML tables,<sup>4,5</sup> while Elmeleegy et al<sup>11</sup> suggested an equal number from HTML lists, a total that does not account for the non-English Web. Finally, our experience at Google

## » key insights

- Because data on the Web is about everything, any approach that attempts to leverage it cannot rely on building a model of the data ahead of time but on domain-independent methods instead.
- The sheer quantity and heterogeneity of structured data on the Web enables new approaches to problems involving data integration from multiple sources.
- While the content of structured data is typically different from what is found in text on the Web, each content collection can be leveraged to better understand other collections.



suggests the Deep Web alone can generate more than one billion pages of valuable structured data. The result is an astounding number of distinct structured data sets, most still waiting to be exposed more effectively to users.

This structured data differs from data stored in traditional relational databases in several ways:

**Data in “page context” must be extracted.** Consider a database embedded in an HTML table (such as local coffeehouses in Seattle and the U.S. presidents in Figure 1). To the user the data set appears to be structured, but a computer program must be able to automatically distinguish it from, say, a site’s navigational bar that also uses an HTML table. Similarly, a Web form that gives access to an interesting Deep Web database, perhaps containing all Starbucks locations in the world, is not that different from a form offering simple mailing-list signup. The computer program might also have to automatically extract schema information in the form of column labels sometimes appearing in the first row of an HTML table but that sometimes do not exist at all. Moreover, the subject of a table may be described in the surrounding text, making it difficult to extract. There is nothing akin to traditional relational metadata that leaves no doubt as to how many tables there are and the relevant schema information for each table.

**No centralized data design or data-quality control.** In a traditional database, the relational schema provides a topic-specific design that must be observed by all data elements. The database and the schema may also enforce certain quality controls (such as observing type consistency within a column, disallowing empty cells, and constraining data values to a certain legal range). For example, the set of coffeehouses may have a column called *year-founded* containing integers constrained to a relatively small range. Neither data design nor quality control exists for Web data; for





example, if a year-founded string is in the first row, there is nothing to prevent the string *macchiatone* from appearing beneath it. Any useful application making use of Web data must also be able to address uncertain data design and quality.

**Vast number of topics.** A tradi-

tional database typically focuses on a particular domain (such as products or proteins) and therefore can be modeled in a coherent schema. On the Web, data covers everything, and is also one of its appeals. The breadth and cultural variations of data on the Web make it inconceivable that any

manual effort would be able to create a clean model of all of it.

Before addressing the challenges associated with accessing structured data on the Web, it is important to ask what users might do with such data. Our work is inspired by the following example benefits:

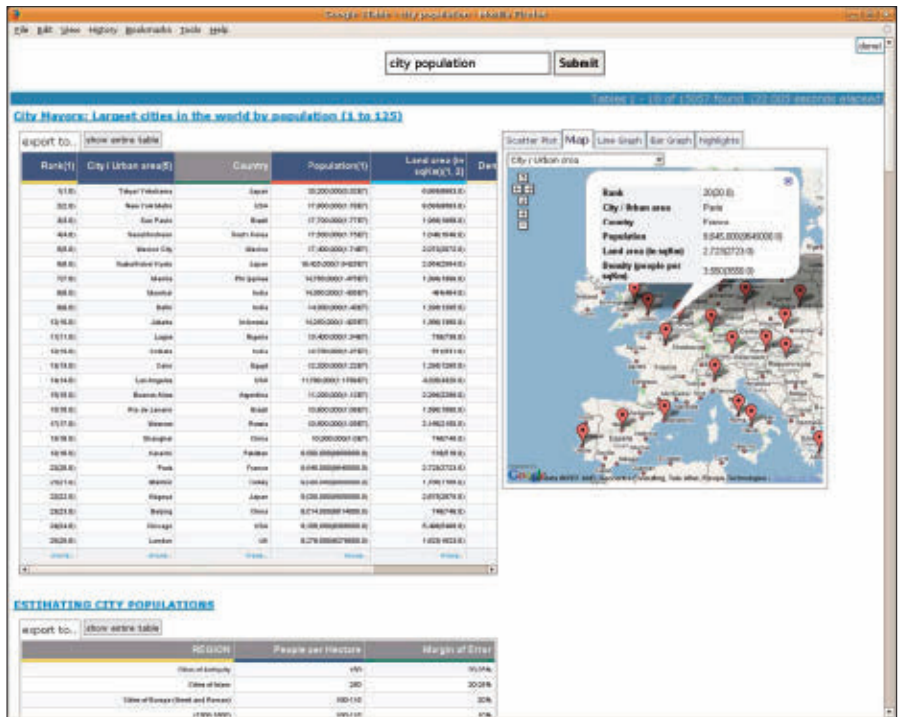
**Improve Web search.** Structured Web data can help improve Web search in a number of ways; for example, Deep Web databases are not generally available to search engines, and, by surfacing this data, a Deep Web exploration tool can expand the scope and quality of the Web-search index. Moreover, the layout structure can be used as a relevance signal to the search ranker; for example, an HTML table-embedded database with a column *calories* and a row *latte*, should be ranked fairly high in response to the user query *latte calories*. Traditionally, search engines use the proximity of terms on a page as a signal of relatedness; in this case, the two terms are highly related, even though they may be distant from each other on the page.

**Enable question answering.** A long-standing goal for Web search is to return answers in the form of facts; for example, in the *latte calories* query, rather than return a URL a search engine might return an actual numerical value extracted from the HTML table. Web search engines return actual answers for very specific query domains (such as weather and flight conditions), but doing so in a domain-independent way is a much greater challenge.

**Enable data integration from multiple Web sources.** With all the data sets available on the Web, the idea of combining and integrating them in ad hoc ways is immensely appealing. In a traditional database setting, this task is called data integration; on the Web, combining two disparate data sets is often called a “mashup.” While a traditional database administrator might integrate two employee databases with great precision and at great cost, most combinations of Web data should be akin to Web search—relatively imprecise and inexpensive; for example, a user might combine the set of coffeehouses with a database of WiFi hotspots, where speed



**Figure 1.** Typical use of the `table` tag to describe relational data that has structure never explicitly declared by the author, including metadata consisting of several typed and labeled columns, but that is obvious to human observers. The navigation bars at the top of the page are also implemented through the `table` tag but do not contain relational-style data.



**Figure 2.** Results of a keyword query search for “city population,” returning a relevance-ranked list of databases. The top result contains a row for each of the most populous 125 cities and columns for “City/Urban Area,” “Country,” “Population,” and “rank” (by population among all the cities in the world). The system automatically generated the image at right, showing the result of clicking on the “Paris” row. The title (“City Mayors...”) links to the page where the original HTML table is located.

is more important than flawless accuracy. Unlike most existing mashup tools, we do not want users to be limited to data that has been prepared for integration (such as already available in XML).

The Web is home to many kinds of structured data, including embedded in text, socially created objects, HTML tables, and Deep Web databases. We have developed systems that focus on HTML tables and Deep Web databases. WebTables extracts relational data from crawled HTML tables, thereby creating a collection of structured databases several orders of magnitude larger than any other we know of. The other project surfaces data obtained from the Deep Web, almost all hidden behind Web forms and thus inaccessible. We have also constructed a tool (not discussed here) called Octopus that allows users to extract, clean, and integrate Web-embedded data.<sup>3</sup> Finally, we built a third system, called Google Fusion Tables,<sup>13</sup> a cloud-based service that facilitates creation and publication of structured data on the Web, therefore complementing the two other projects.

### WebTables

The WebTables system<sup>4,5</sup> is designed to extract relational-style data from the Web expressed using the HTML table tag. Figure 1 is a table listing American presidents (<http://www.enchantedlearning.com/history/us/pres/list.shtml>) with four columns, each with topic-specific label and type (such as **President** and **Term as President**) as a date range; also included is a tuple of data for each row. Although most of the structured-data metadata is implicit, this Web page essentially contains a small relational database anyone can crawl.

Not all table tags carry relational data. Many are used for page layout, calendars, and other nonrelational purposes; for example, in Figure 1, the top of the page contains a table tag used to lay out a navigation bar with the letters A–Z. Based on a human-judged sample of raw tables, we estimate up to 200 million true relational databases in English alone on the Web. In general, less than 1% of the content embedded in the HTML table tags represents good tables. In-



**Any useful application making use of Web data must also be able to address uncertain data design and quality.**



deed, the relational databases in the WebTables corpus form the largest database corpus we know of, by five orders of decimal magnitude.<sup>a</sup>

WebTables focuses on two main problems surrounding these databases: One, perhaps more obvious, is how to extract them from the Web in the first place, given that 98.9% of tables carry no relational data. Once we address this problem, we can move to the second—what to do with the resulting huge collection of databases.

**Table extraction.** The WebTables table-extraction process involves two steps: First is an attempt to filter out all the nonrelational tables. Unfortunately, automatically distinguishing a relational table from a nonrelational table can be difficult. To do so, the system uses a combination of handwritten and statistically trained classifiers that use topic-independent features of each table; for example, high-quality data tables often have relatively few empty cells. Another useful feature is whether each column contains a uniform data type (such as all dates or all integers). Google Research has found that finding a column toward the left side of the table with values drawn from the same semantic type (such as country, species, and institution) is a valuable signal for identifying high-quality relational tables.

The second step is to recover metadata for each table passing through the first filter. Metadata is information that describes the data in the database (such as number of columns, types, and names). In the case of the presidents, the metadata contains the column labels *President*, *Party*, and so on. For coffeehouses, it might contain *Name*, *Speciality*, and *Roaster*. Although metadata for a traditional relational database can be complex, the goal for WebTables metadata is modest—determine whether or not the first row of the ta-

<sup>a</sup> The second-largest collection we know is due to Wang and Hu,<sup>22</sup> who also tried to gather data from Web pages but with a relatively small and focused set of input pages. Other research on table extraction has not focused on large collections.<sup>10,12,23</sup> Our discussion here refers to the number of distinct databases, not the number of tuples. Limaye et al<sup>16</sup> described techniques for mapping entities and columns in tables to an ontology.




ble includes labels for each column. When inspecting tables by hand, we found 70% of good relational-style tables contain such a metadata row. As with relational filtering, we used a set of trained classifiers to automatically determine whether or not the schema row is present.


The two techniques together allowed WebTables to recover 125 million high-quality databases from a large general Web crawl (several billion Web pages). The tables in this corpus contained more than 2.6 million unique “schemas,” or unique sets of attribute strings. This enormous data set is a unique resource we explore in the following paragraphs.

**Leveraging extracted data.** Aggregating data over the extracted WebTables data, we can create new applications previously difficult or impossible through other techniques. One such application is structured data search. Traditional search engines are tuned to return relevant documents, not data sets, so users searching for data are generally ill-served. Using the extracted WebTables data, we implemented a search engine that takes a keyword query and returns a ranked list of databases instead of URLs; Figure 2 is a screenshot of the prototype system. Because WebTables extracted structural information for each object in the search engine’s index, the results page can be more interesting than in a standard search engine. Here, the page of search results contains an automatically drawn map reflecting the cities listed in the data set; imagine the system being used by knowledge workers who want to find data to add to a spreadsheet.

In addition to the data in the tables, we found significant value in the collection of the tabular schemata we collected. We created the Attribute Correlation Statistics Database (ACSDb) consisting of simple frequency counts for each unique piece of metadata WebTables extracts; for example, the database of presidents mentioned earlier adds a single count to the four-element set `president`, `party`, `term-as-president`, `vice-president`. By summing individual attribute counts over all entries in the ACSDb, WebTables is able to compute various attribute probabilities, given a



**An important lesson we learned is there is significant value in analyzing collections of metadata on the Web, in addition to the data itself.**



randomly chosen database; for example, the probability of seeing the name attribute is far higher than seeing the roaster attribute.

WebTables also computes conditional probabilities, so, for example, we learn that  $p(\text{roaster} \mid \text{house-blend})$  is much higher than  $p(\text{roaster} \mid \text{album-title})$ . It makes sense that two coffee-related attributes occur together much more often than a combination of a coffee-related attribute and, say, a music-related attribute. Using these probabilities in different ways, we can build interesting new applications, including these two:

*Schema autocomplete.* The database schema auto-complete application is designed to assist novice database designers. Like the tab-complete feature in word processors, schema autocomplete takes a few sample attributes from the user and suggests additional attributes to complete the table; for example, if a user types `roaster` and `house-blend`, the auto-complete feature might suggest `speciality`, `opening-time` and other attributes to complete the `coffeehouse` schema. Table 1 lists example outputs from our auto-complete tool, which is also useful in scenarios where users should be encouraged to reuse existing terminologies in their schemas.

The auto-complete algorithm is easily implemented with probabilities from the ACSDb. The algorithm repeatedly emits the attribute from the ACSDb to yield the highest probability, when conditioned on the attributes the user (or algorithm) has already suggested. The algorithm terminates when the attribute yielding the highest probability is below a tunable threshold.

*Synonym finding.* The WebTables synonym-finding application uses ACSDb probabilities to automatically detect likely attribute synonyms; for example, `phone-number` and `phone-#` are two attribute labels that are semantically equivalent. Synonyms play a key role in data integration. When we merge two databases on the same topic created by different people, we first need to reconcile the different attribute names used in the two databases. Finding these synonyms is generally done by the application de-

signer or drawn automatically from a pre-compiled linguistic resource (such as a thesaurus). However, the task of synonym finding is complicated by the fact that attribute names are often acronyms or word combinations, and their meanings are highly contextual. Unfortunately, manually computing a set of synonyms is burdensome and error-prone.

WebTables uses probabilities from the ACSDB to encode three observations about good synonyms:

- ▶ Two synonyms should not appear together in any known schema, as it would be repetitive on the part of the database designer;

- ▶ Two synonyms should share common co-attributes; for example, phone-number and phone-# should both appear along with name and address; and

- ▶ The most accurate synonyms are popular in real-world use cases.

WebTables can encode each of these observations in terms of attribute probabilities using ACSDB data. Combining them, we obtain a formula for a synonym-quality score WebTables uses to sort and rank every possible attribute pair; Table 2 lists a series of input domains and the output pairs of the synonym-finding system.

## Deep Web Databases

Not all structured data on the Web is published in easily accessible HTML tables. Large volumes of data stored in back-end databases are often made available to Web users only through HTML form interfaces; for example, a large chain of coffeehouses might have a database of store locations that are retrieved by zip code using the HTML form on the company's Web site, and users retrieve data by performing valid form submissions. On the back-end, HTML forms are processed by either posing structured queries over relational databases or sending keyword queries over text databases. The retrieved content is published on Web pages in structured templates, often including HTML tables.

While WebTables-harvested tables are potentially reachable by users posing keyword queries on search engines, the content behind HTML forms was for a long time believed to be beyond the reach of search en-

gines; few hyperlinks point to Web pages resulting from form submissions, and Web crawlers did not have the ability to automatically fill out forms. Hence, the names “Deep,” “Hidden,” and “Invisible Web” have all been used to refer to the content accessible only through forms. Bergman<sup>2</sup> and He et al<sup>14</sup> have speculated that the data in the Deep Web far exceeds the data indexed by contemporary search engines. We estimate at least 10 million potentially useful distinct forms<sup>18</sup>; our previous work<sup>17</sup> has a more thorough discussion of the Deep Web literature and its relation to the projects described here.

The goal of Google's Deep Web Crawl Project is to make Deep Web content accessible to search-engine users. There are two complementary approaches to offering access to it: create vertical search engines for specific topics (such as coffee, presidents, cars, books, and real estate) and surface Deep Web content. In the first, for each vertical, a designer must create a mediated schema visible to users and create semantic mappings from the Web sources to the mediated schema. However, at Web scale, this approach suffers from several drawbacks:

- ▶ A human must spend time and effort building and maintaining each mapping;

- ▶ When dealing with thousands of domains, identifying the topic relevant to an arbitrary keyword query is extremely difficult; and

- ▶ Data on the Web reflects every topic in existence, and topic boundaries are not always clear.

The Deep Web Crawl project followed the second approach to surface DeepWeb content, pre-computing the most relevant form submissions for all interesting HTML forms. The URLs resulting from these submissions can then be added to the crawl of a search engine and indexed like any other HTML page. This approach leverages the existing search-engine infrastructure, allowing the seamless inclusion of Deep Web pages into Web-search results. The system currently surfaces content for several million Deep Web databases spanning more than 50 languages and several hundred domains, and the surfaced pages contribute results to more than 1,000 Web-search queries per second on Google.com. For example, as of the writing of this article, a search query for `citibank atm 94043` will return in the first position a parameterized URL surfacing

**Table 1. Sample output from the schema autocomplete tool. To the left is a user's input attribute; to the right are sample schemas.**

Input attribute	Auto-completer output
name	name, size, last-modified, type
instructor	instructor, time, title, days, room, course
elected	elected, party, district, incumbent, status, opponent, description
ab	ab, h, r, bb, so, rbi, avg, lob, hr, pos, batters
sqft	sqft, price, baths, beds, year, type, lot-sqft, days-on-market, stories

**Table 2. Sample output from the synonym-finding tool. To the left are the input context attributes; to the right are synonymous pairs generated by the system.**

Input context	Synonym-finder outputs
name	e-mail email, phone telephone, e-mail address email address, date last-modified
instructor	course-title title, day days, course course-#, course-name course-title
elected	candidate name, presiding-officer speaker
ab	k so, h hits, avg ba, name player
sqft	bath baths, list list-price, bed beds, price rent

results from a database of ATM locations—a very useful search result that would not have appeared otherwise.

Pre-computing the set of relevant form submissions for any given form is the primary difficulty with surfacing; for example, a field with label roaster should not be filled in with value toyota. Given the scale of a Deep Web crawl, it is crucial there be no human involvement in the process of pre-computing form submissions. Hence, previous work that either addressed the problem by constructing mediator systems one domain at a time<sup>8,9,21</sup> or needed site-specific wrappers or extractors to extract documents from text databases<sup>1,19</sup> could not be applied.

Surfacing Deep Web content involves two main technical challenges:

- Values must be selected for each input in the form; value selection is trivial for select menus but very challenging for text boxes; and

- Forms have multiple inputs, and using a simple strategy of enumerating all possible form submissions can be wasteful; for example, the search form on cars.com has five inputs, and a cross product will yield more than 200 million URLs, even though cars.com lists only 650,000 cars for sale.<sup>7</sup>

The full details on how we addressed these challenges are in Madhavan et al.<sup>18</sup> Here, we outline how we approach the two problems:

**Selecting input values.** A large number of forms have text-box inputs and require valid input values for the retrieval of any data. The system must therefore choose a good set of values to submit in order to surface useful result pages. Interestingly, we found it is not necessary to have a complete understanding of the semantics of the form to determine good candidate text inputs. To understand why, first note that text inputs fall into one of two categories: generic search inputs that accept most keywords and typed text inputs that accept only values in a particular topic area.

For search boxes, the system predicts an initial set of candidate keywords by analyzing text from the form site, using the text to bootstrap an iterative probing process. The system submits the form with candidate keywords; when valid form submissions

result, the system extracts more keywords from the resulting pages. This iterative process continues until either there are no new candidate keywords or the system reaches a pre-specified target number of results. The set of all candidate keywords can then be pruned, choosing a small number that ensures diversity of the exposed database content. Similar iterative probing approaches have been used to extract text documents from specific databases.<sup>1,6,15,19,20</sup>

For typed text boxes, the system attempts to match the type of the text box against a library of types common across topics (such as U.S. zip codes). Note that probing with values of the wrong type results in invalid submissions or pages with no results. We found even a library of just a few types can cover a significant number of text boxes.

**Selecting input combinations.** For HTML forms with more than one input, a simple strategy of enumerating the entire cross-product of all possible values for each input will result in a huge number of output URLs. Crawling too many URLs drains the resources of a search engine Web crawler while posing an unreasonable load on Web servers hosting the HTML forms. Choosing a subset of the cross-product that yields results that are nonempty, useful, and distinct is an algorithmic challenge.<sup>18</sup> The system incrementally traverses the search space of all possible subsets of inputs. For a given subset, it tests whether it is informative, or capable of generating URLs with sufficient diversity in their content. As we showed in Madhavan et al,<sup>18</sup> only a small fraction of possible input sets must be tested, and, for each subset, the content of only a sample of generated URLs must be examined. Our algorithm is able to extract large fractions of underlying Deep Web databases without human supervision, using only a small number of form submissions. Furthermore, the number of form submissions the system generates is proportional to the size of the database underlying the form site, rather than the number of inputs and input combinations in the form.

**Limitations of surfacing.** By creating Web pages, surfacing does not

preserve the structure or semantics of the data gathered from the underlying DeepWeb databases. But the loss in semantics is also a lost opportunity for query answering; for example, suppose a user searched for “used ford focus 1993” and a surfaced used-car listing page included Honda Civics, with a 1993 Honda Civic for sale, but also said “has better mileage than the Ford Focus.” A traditional search engine would consider such a surfaced Web page a good result, despite not being helpful to the user. We could avoid this situation if the surfaced page had a search-engine-specific annotation that the page was for used-car listings of Honda Civics. One challenge for an automated system is to create a set of structure-aware annotations textual search engines can use effectively.

## Next Steps

These two projects represent first steps in retrieving structured data on the Web and making it directly accessible to users. Searching it is not a solved problem; in particular, search over large collections of data is still an area in need of significant research, as well as integration with other Web search. An important lesson we learned is there is significant value in analyzing collections of metadata on the Web, in addition to the data itself.

Specifically, from the collections we have worked with—forms and HTML tables—we have extracted several artifacts:

- A collection of forms (input names that appear together and values for select menus associated with input names);

- A collection of several million schemata for tables, or sets of column names appearing together; and

- A collection of columns, each with values in the same domain (such as city names, zip codes, and car makes).

**Semantic services.** Generalizing from our synonym finder and schema auto-complete, we build from the schema artifacts a set of semantic services that form a useful infrastructure for many other tasks. An example of such a service is that, given a name of an attribute, return a set of values for its column; such a service can automatically fill out forms in order to surface Deep Web content. A second



example is, given an entity, return a set of possible properties—attributes and relationships—that may be associated with it. Such a service would be useful for both information-extraction tasks and query expansion.

**Structured data from other sources.** Some of the principles of our previous projects are useful for extracting structured data from other growing sources on the Web:

*Socially created data sets.* These data sets (such as encyclopedia articles, videos, and photographs) are large and interesting and exist mainly in site-specific silos, so integrating them with information extracted from the wider Web would be useful;

*Hypertext-based data models.* These models, in which page authors use combinations of HTML elements (such as a list of hyperlinks), perform certain data-model tasks (such as indicate that all entities pointed to by the hyperlinks belong to the same set); this category can be considered a generalization of the observation that HTML tables are used to communicate relations; and

*Office-style documents.* These documents (such as spreadsheets and slide presentations) contain their own structured data, but because they are complicated, extracting information from them can be difficult, though it also means they are a tantalizing target.

**Creating and publishing structured data.** The projects we've described are reactive in the sense that they try to leverage data already on the Web. In a complementary line of work, we created Google Fusion Tables,<sup>13</sup> a service that aims to facilitate the creation, management, and publication of structured data, enabling users to upload tabular data files, including spreadsheets and CSV, of up to 100MB. The system provides ways to visualize the data—maps, charts, timelines—along with the ability to query by filtering and aggregating the data. Fusion Tables enables users to integrate data from multiple sources by performing joins across tables that may belong to different users. Users can keep the data private, share it with a select set of collaborators, or make it public. When made public, search engines are able to crawl the tables,

thereby providing additional incentive to publish data. Fusion Tables also includes a set of social features (such as collaborators conducting detailed discussions of the data at the level of individual rows, columns, and cells). For notable uses of Fusion Tables go to <https://sites.google.com/site/fusiontablestalks/stories>.

## Conclusion

Structured data on the Web involves several technical challenges: difficult to extract, typically disorganized, and often messy. The centralized control enforced by a traditional database system avoids all of them, but centralized control also misses out on the main virtues of Web data—that it can be created by anyone and covers every topic imaginable. We are only starting to see the benefits that might accrue from these virtues. In particular, as illustrated by WebTables synonym finding and schema auto-suggest, we see the results of large-scale data mining of an extracted (and otherwise unobtainable) data set.

It is often argued that only select Web-search companies are able to carry out research of the flavor we've described here. This argument holds mostly for research projects involving access to logs of search queries, but the research described here was made easier by having access to a large Web index and computational infrastructure, and much of it can be conducted at academic institutions as well, in particular when it involves such challenges as extracting the meaning of tables on the Web and finding interesting combinations of such tables. ACSDB is freely available to researchers outside of Google (<https://www.eecs.umich.edu/michjc/acsdb.html>); we also expect to make additional data sets available to foster related research. **C**

## References

- Barbosa, L. and Freire, J. Siphoning Hidden-Web data through keyword-based interfaces. In *Proceedings of the Brazilian Symposium on Databases*, 2004, 309–321.
- Bergman, M.K. The Deep Web: Surfacing hidden value. *Journal of Electronic Publishing* 7, 1 (2001).
- Cafarella, M.J., Halevy, A.Y., and Khoussainova, N. Data integration for the relational Web. *Proceedings of the VLDB Endowment* 2, 1 (2009), 1090–1101.
- Cafarella, M.J., Halevy, A.Y., Wang, D.Z., Wu, E., and Zhang, Y. WebTables: Exploring the power of tables on the Web. *Proceedings of the VLDB Endowment* 1, 1

(Aug. 2008), 538–549.

- Cafarella, M.J., Halevy, A.Y., Zhang, Y., Wang, D.Z., and Wu, E. Uncovering the relational Web. In *Proceedings of the 11th International Workshop on the Web and Databases* (Vancouver, B.C., June 13, 2008).
- Callan, J.P. and Connell, M.E. Query-based sampling of text databases. *ACM Transactions on Information Systems* 19, 2 (2001), 97–130.
- Cars.com (faq); <http://siy.cars.com/siy/qsg/faqgeneralinfo.jsp#howmanyads>
- Cazoodle apartment search; <http://apartments.cazoodle.com/>
- Chang, K.C.-C., He, B., and Zhang, Z. Toward large-scale integration: Building a metaquerier over databases on the Web. In *Proceedings of the Conference on Innovative Data Systems Research* (Asilomar, CA, Jan. 2005).
- Chen, H., Tsai, S., and Tsai, J. Mining tables from large-scale html texts. In *Proceedings of the 18th International Conference on Computational Linguistics* (Saarbrücken, Germany, July 31–Aug. 4, 2000), 166–172.
- Elmeleegy, H., Madhavan, J., and Halevy, A. Harvesting relational tables from lists on the Web. *Proceedings of the VLDB Endowment* 2, 1 (2009), 1078–1089.
- Gatterbauer, W., Bohunsky, P., Herzog, M., Krüpl, B., and Pollak, B. Towards domain-independent information extraction from Web tables. In *Proceedings of the 16th International World Wide Web Conference* (Banff, Canada, May 8–12, 2007), 71–80.
- Gonzalez, H., Halevy, A., Jensen, C., Langen, A., Madhavan, J., Shapley, R., Shen, W., and Goldberg-Kidon, J. Google Fusion Tables: Web-centered data management and collaboration. In *Proceedings of the SIGMOD ACM Special Interest Group on Management of Data* (Indianapolis, 2010). ACM Press, New York, 2010, 1061–1066.
- He, B., Patel, M., Zhang, Z., and Chang, K.C.-C. Accessing the Deep Web. *Commun. ACM* 50, 5 (May 2007), 94–101.
- Ipeirotis, P.G. and Gravano, L. Distributed search over the Hidden Web: Hierarchical database sampling and selection. In *Proceedings of the 28th International Conference on Very Large Databases* (Hong Kong, Aug. 20–23, 2002), 394–405.
- Limaye, G., Sarawagi, S., and Chakrabarti, S. Annotating and searching Web tables using entities, types, and relationships. *Proceedings of the VLDB Endowment* 3, 1 (2010), 1338–1347.
- Madhavan, J., Ko, D., Kot, L., Ganapathy, V., Rasmussen, A., and Halevy, A.Y. Google's Deep Web Crawl. *Proceedings of the VLDB Endowment* 1, 1 (2008), 1241–1252.
- Madhavan, J., Cohen, S., Dong, X.L., Halevy, A.Y., Jeffery, S.R., Ko, D., and Yu, C. Web-scale data integration: You can afford to pay as you go. In *Proceedings of the Second Conference on Innovative Data Systems Research* (Asilomar, CA, Jan. 7–10, 2007), 342–350.
- Ntoulas, A., Zerkos, P., and Cho, J. Downloading textual Hidden Web content through keyword queries. In *Proceedings of the Joint Conference on Digital Libraries* (Denver, June 7–11, 2005), 100–109.
- Raghavan, S. and Garcia-Molina, H. Crawling the Hidden Web. In *Proceedings of the 27th International Conference on Very Large Databases* (Rome, Italy, Sept. 11–14, 2001), 129–138.
- Trulia; <http://www.trulia.com/>
- Wang, Y. and Hu, J. A machine-learning-based approach for table detection on the Web. In *Proceedings of the 11th International World Wide Web Conference* (Honolulu, 2002), 242–250.
- Zanibbi, R., Blostein, D., and Cordy, J. A survey of table recognition: Models, observations, transformations, and inferences. *International Journal on Document Analysis and Recognition* 7, 1 (2004), 1–16.

**Michael J. Cafarella** ([michjc@umich.edu](mailto:michjc@umich.edu)) is an assistant professor of computer science and engineering at the University of Michigan, Ann Arbor, MI.

**Alon Halevy** ([halevy@google.com](mailto:halevy@google.com)) is Head of the Structured Data Management Research Group, Google Research, Mountain View, CA.

**Jayant Madhavan** ([jayant@google.com](mailto:jayant@google.com)) a senior software engineer at Google Research, Mountain View, CA.

© 2011 ACM 0001-0782/11/0200 \$10.00

## What would it take for a true personal knowledge base to generate the benefits envisioned by Vannevar Bush?

BY STEPHEN DAVIES

# Still Building the Memex

AS WORLD WAR II mercifully drew to a close, Vannevar Bush, President Truman's Director of Scientific Research, surveyed the post-war landscape and laid out what he viewed as the most important forthcoming challenges to humankind.<sup>9</sup> In his oft-cited article, he also described a hypothetical information storage device called the "memex,"<sup>a</sup>

a The word "memex" is thought to be an abbreviation for "memory extender," though this is never explained in the article.

intended to tackle the information overload problem that was already formidable in 1945. In Bush's own words:

"Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and to coin one at random, 'memex' will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."

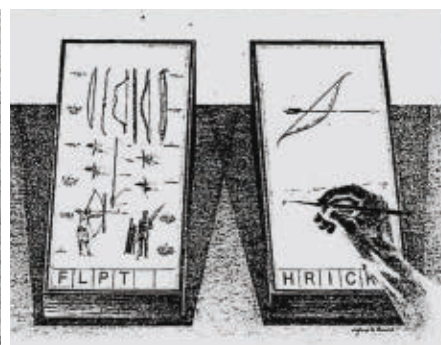
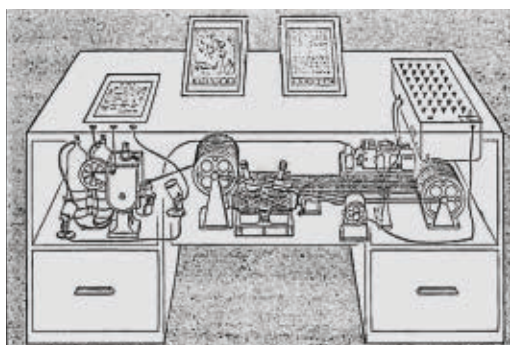
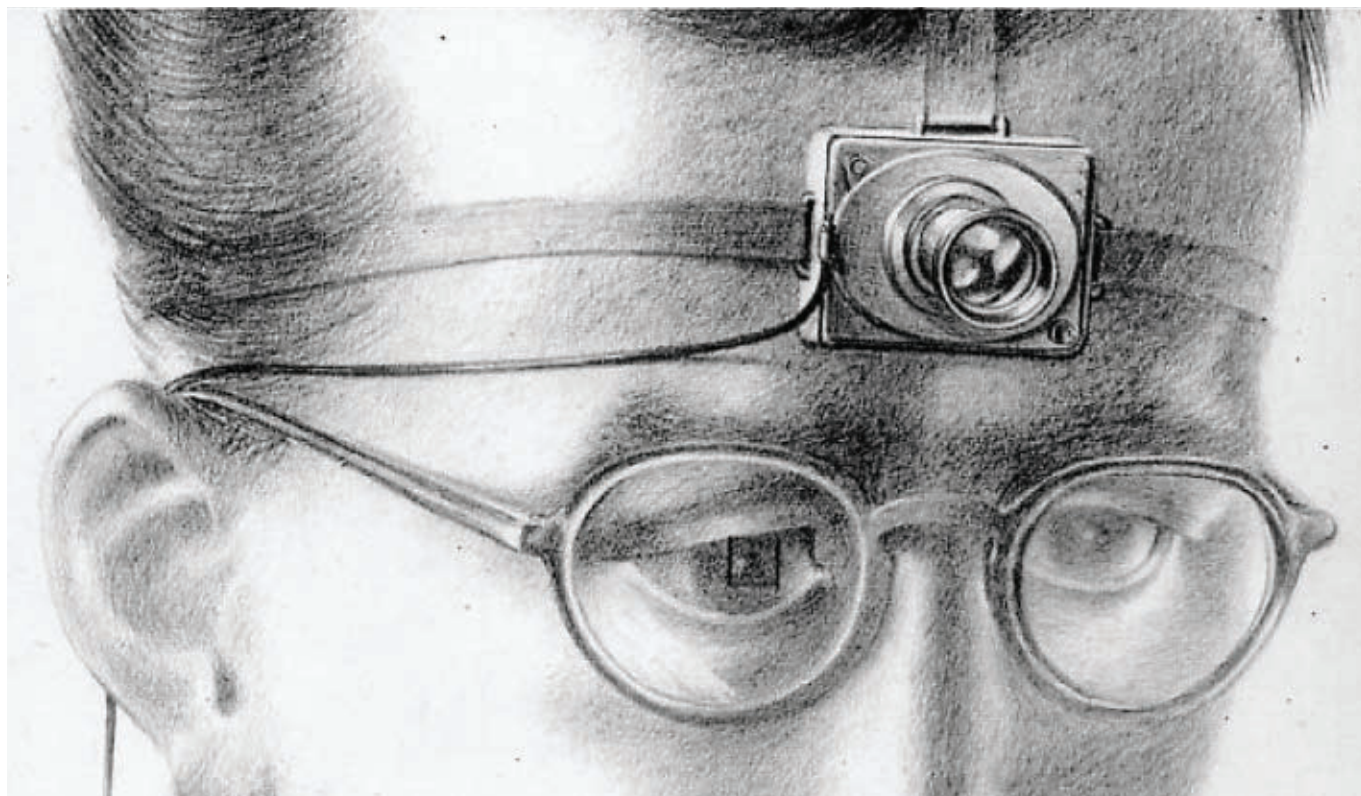
He went on to specify that the user should be able to "add marginal notes and comments," and "build a trail of his interest" through the larger information space. And Bush's emphasis throughout the article was on expanding our own powers of recollection: "Man needs to mechanize his record more fully," he says, if he is not to "become bogged down...by overtaxing his limited memory."

According to Bush, this kind of ubiquitously available digital assistant, capturing and faithfully reproducing a person's thoughts, sources, and organization of information, would be more than anything else the key to maximizing mankind's potential in the coming information age. Granted, the vision he described has inspired a variety of other research endeavors, from information retrieval to distributed hypertext systems. But

### » key insights

- Personal knowledge bases (PKBs) are becoming increasingly critical tools for information professionals struggling to cope with today's explosion of digital information.
- PKB development efforts have their roots in diverse research communities, including mind mapping, knowledge organization tools, and hypertext systems.
- Modern approaches still strive to define the perfect data model for a PKB. Different structures—such as trees, categories, and spatial layouts—provide different capabilities and limitations, yet no one model has emerged as clearly superior.





Figures from the September 10, 1945 *Life* magazine article “As We May Think” by Vannevar Bush.

a great deal of his design can be seen as an example of what this article will characterize as a *personal knowledge base*.

Personal knowledge bases have existed in some form since humankind felt compelled to manage information: card files, personal libraries, Da Vinci’s notebooks. Today there are literally dozens of software products attempting to satisfy these needs. Designers approach the problem in different ways, but have the same aim. And no wonder. If a “memex” was needed in Bush’s day, then today’s information explosion<sup>20</sup> makes it an order of magnitude more important. If a human’s only tool for retaining what they learn is their biological memory, their base of knowledge will be porous indeed.

Yet the problem is deceptively dif-

ficult to solve. How to design a system that attempts to capture human memories? To interrelate heterogeneous information, assimilated from numerous diverse sources and filtered through an individual’s subjective understanding? To give users a natural way to search and correlate and extend that information? These are mysteries that many have attempted to solve but which remain tantalizingly incomplete.

In this article, we look at a number of these systems and provide a taxonomy for classifying their approaches.

### The Personal Knowledge Base

We define a Personal Knowledge Base—or PKB—as an electronic tool through which an individual can express, capture, and later retrieve the personal knowledge he or she has ac-

quired. Our definition has three components:

*Personal:* Like Bush’s memex, a PKB is intended for private use, and its contents are custom tailored to the individual. It contains trends, relationships, categories, and personal observations that its owner sees but which no one else may agree with. Many of the issues involved in PKB design are also relevant in collaborative settings, as when a homogeneous group of people is jointly building a shared knowledge base. In this case, the knowledge base could simply reflect the consensus view of all contributors; or, perhaps better, it could simultaneously store and present alternate views of its contents, so as to honor several participants who may organize it or view it differently. This can introduce another level of complexity.



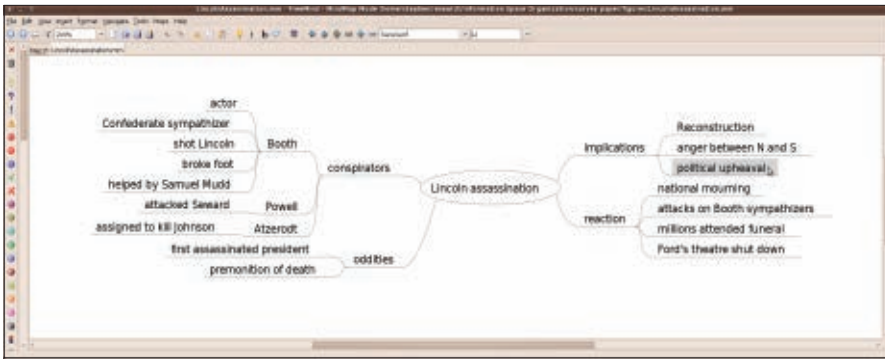


Figure 1. Freemind, a mind map creation tool.

**Knowledge:** A PKB primarily contains knowledge, not information. That is, its purpose is not simply to aggregate all the information sources one has seen, but to preserve the select knowledge that one has *learned* from those sources. Psychologically, knowledge is the formation of a mental model in one's own mind that corresponds to the information encountered.<sup>21</sup>

**Base:** A PKB preserves knowledge for the long haul. It is ideally future-proof: immune to shifts in technology and to disaster. It can be fluidly searched and browsed. It also forms a consolidated, integrated whole, without partitions that might isolate some areas of knowledge from others. This is because it is a reflection of one's own memory, which, as Bush and many others have observed, can freely associate any two thoughts together, without restriction.<sup>3,9</sup>

## Benefits

What can be gained from using a PKB? An idea of the presumed advantages can be gleaned from the way in which today's numerous solutions are "pitched:"

**Knowledge generation and formulation.** Here the emphasis is on procedure, not persistence; it is the act of

simply using the tool to express one's knowledge that helps, rather than the ability to retrieve it later. Systems boast that they can "convert random thoughts generated while you are the most creative into the linear thoughts needed most when communicating;" "help you relate and arrange random ideas;" and "stimulate your brain" (<http://mindmappersusa.com>).

**Knowledge capture.** PKBs do not merely allow one to express knowledge, but also to capture it before it elusively disappears. The point is to lower the burden of jotting down one's thoughts so that neither task nor thought process is interrupted. StayAt-Play's Idea Knot, for example, asserts that "it is very quick to open a...document and within seconds record the essence of that new idea without distractions, while your mind is focused on it and without disturbing the flow of your current work." (<http://www.stayatplay.com>).

**Knowledge organization.** A short study on note-taking habits found that "better organization" was the improvement people most desired in their own information recording practices.<sup>24</sup> PKB systems like Aquaminds Notetaker (<http://aquaminds.com>) profess to answer this need, allowing one to "organize personal information," and claiming to be "a more productive way to stay organized."

**Knowledge management and retrieval.** Perhaps the most critical aspect of a PKB is that the knowledge it stores is permanent and accessible, ready to be retrieved at any later time. PersonalKnowbase (<http://bitsmithsoft.com>) claims it will "give you a place to stash all those stray snips of knowledge where they can be quickly recalled when you need them," and MicroLog-

ic's InfoSelect lets you "find any data in an instant, no matter where or how you entered it" (<http://miclog.com>).

## PKB Systems: Past and Present

A plethora of candidate PKB systems have emerged over the past decades. Here, we give an overview of some of the more notable efforts from three distinct research communities.

**Graphical knowledge capture tools.** Much fanfare has been generated in the last 30 years around pictorial knowledge representations. Some claim that drawing informal diagrams to represent abstract knowledge is an excellent way to communicate complex ideas, enhance learning, and even to "unlock the potential of the brain."

"Mind mapping" and "concept mapping" are the two most popular paradigms in graphical knowledge capture. A mind map is essentially nothing more than a visual outline, in which a main idea or topic is written in the center of the diagram, and sub-topics radiate outward in increasing levels of specificity. The primary value is in the freeform, spatial layout, and the ability for a software application to hide or reveal select levels of detail. The open source Freemind project (see Figure 1) is just one of literally dozens of such tools.

Concept maps<sup>34</sup> are based on the premise that newly encountered knowledge must be related to one's prior knowledge in order to be properly understood. Concept maps help depict such connections graphically (see Figure 2). Like mind maps, they feature evocative words or phrases in boxes connected by lines. However, there are important differences in the underlying data model—tree vs. graph—that will discuss shortly.

**Hypertext systems.** The hypertext community proudly points to Bush's article as the cornerstone of their heritage. Hence the development of hypertext techniques, while seldom applied specifically toward PKB solutions, is historically important. Doug Engelbart, who began developing the first viable hypertext system in 1959, stated his purpose as "the augmentation of man's intellect."<sup>15</sup> In other words, Engelbart's goal was to use the hypertext model specifically to model abstract knowledge.



Figure 2. CMap, a concept map creation tool.

The TextNet<sup>37</sup> and NoteCards<sup>23</sup> (see Figure 3) systems further explored this idea. TextNet revolved around “primitive pieces of text connected with typed links to form a network similar in many ways to a semantic network.”<sup>16</sup> The subsequent NoteCards effort, one of the most influential hypertext efforts in history, was similarly designed to “formulate, structure, compare, and manage ideas.” The popular Hypercard program for the Apple Macintosh<sup>22</sup> offered similar functionality through its notion of a stack of digital cards that could be flexibly interconnected.

The sweeping vision of the Xanadu project<sup>33</sup> was for a web of documents composed of individually addressable chunks of any granularity. Any part of one document could freely refer to any part of another, and this reference would persist even when the contents of one or both documents were updated. An interface based on the parallel visualization of texts and “transpointing windows” made navigating this intricate structure possible. Though not yet fully implemented, Xanadu’s design also makes heavy use of transclusion, as described later. Other examples of knowledge-based hypertext tools include Compendium,<sup>12</sup> PersonalBrain (<http://thebrain.com>), and Tinderbox.<sup>4</sup>

**Note-taking applications.** The most explicit attempt to create a PKB as we have defined it comes from the area of note-taking applications. These software tools allow a user to create bits of text and then organize or categorize them in some way. They draw heavily on the “note-taking” metaphor since it is a familiar operation for users to carry over from their experiences with pen and paper (see Figure 4).

**Supporting bodies of research.** In addition to the kind of user interface exhibited by the systems mentioned here, several underlying technologies are vital for a PKB to function properly. Here we give a brief overview of some of the research efforts that are applicable to PKB implementation.

**Graph-based data storage and retrieval.** Since nearly all PKB systems employ some form of graph or tree as their underlying data model, research from the database community on semi-structured data storage and retrieval is relevant here. Stanford

University’s Lore project<sup>30</sup> was an early implementation of a DBMS solution specifically designed for graph-based data. The architecture provided an efficient storage mechanism, and the Lorel language permitted precise queries to be posed in the absence of a fixed schema. Lore’s notion of a “DataGuide”—a summary of the graph structure of a knowledge base, encoding all possible path traversals—could even provide a basis for assisting users in query formulation, a particular challenge when a knowledge base grows in haphazard fashion.

UnQL<sup>7</sup> was an even more expressive query language than Lorel, based on structural recursion that could be applied to tree-structured data as well as arbitrary graphs. Buneman et al. demonstrated that the language can be efficiently implemented in a way that queries are guaranteed to terminate, and can be optimized just as in the relational algebra. StruQL,<sup>16</sup> though originally designed for the specific task of Web site development, was in fact a general-purpose graph query language with similar features.

More recently, the World Wide Web Consortium’s Semantic Web initiative has prompted numerous implementations of RDF triple stores, or databases that house graph-structured data. Notable examples include Jena<sup>b</sup> and Sesame.<sup>6</sup> SPARQL,<sup>36</sup> an RDF query language for selective retrieval of graph data, was recently adopted as a W3C Recommendation. For tree (as opposed to graph) data, a multitude of XML query languages exist, including XQuery<sup>c</sup> and the recently proposed extension XQFT,<sup>2</sup> which enhances explicit tree navigation constructs with information retrieval on the nodes’ free text.

All of these efforts lend considerable optimism to PKB implementers since they provide techniques for efficiently storing and retrieving the kind of data most PKBs are likely to store.

**Graph data integration.** A large portion of a user’s knowledge consists of bits of information from the external sources they have assimilated. In some cases, these external sources may contain structured or semi-structured

data. This would be the case if one PKB wanted to subsume part of another, say, or if the information source itself was a relational database or came from a graph-structured knowledge store, expressed in RDF or another graph-based representation. In these cases, the research findings of the data integration community can be brought to bear. Stanford University’s pioneering work on the OEM model and query language<sup>35</sup> illustrated a standard mechanism for the exchange of data between diverse and dynamic sources. McBrien and Poulouvasilis<sup>29</sup> showed how XML data sources can be semantically integrated by reducing disparate schemas into a common, lower-level graph (actually, hypergraph) language. Vdovjak and Houben<sup>38</sup> provided a framework for a unified interface to query heterogeneous RDF data sources. These successes place the PKB-related task of subsuming external information on a firm theoretical footing.

**Data provenance.** When assimilating external knowledge, a PKB should also track and retain source information. Research in managing data provenance (sometimes known as data lin-

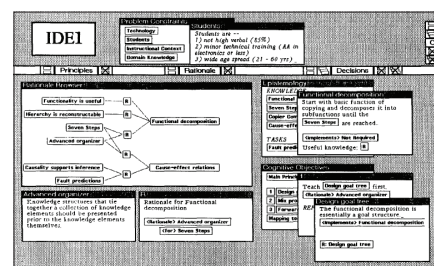


Figure 3. The NoteCards knowledge management environment.

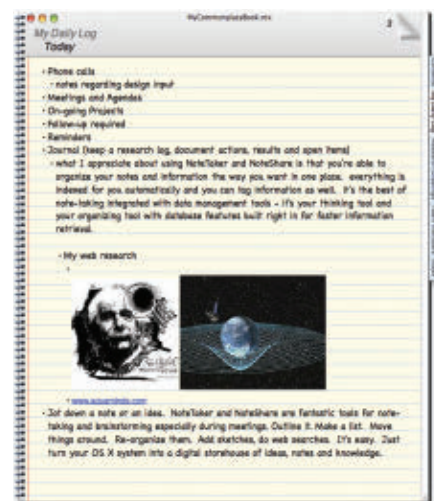


Figure 4. AquaMinds NoteTaker, a hierarchically based note-taking application.

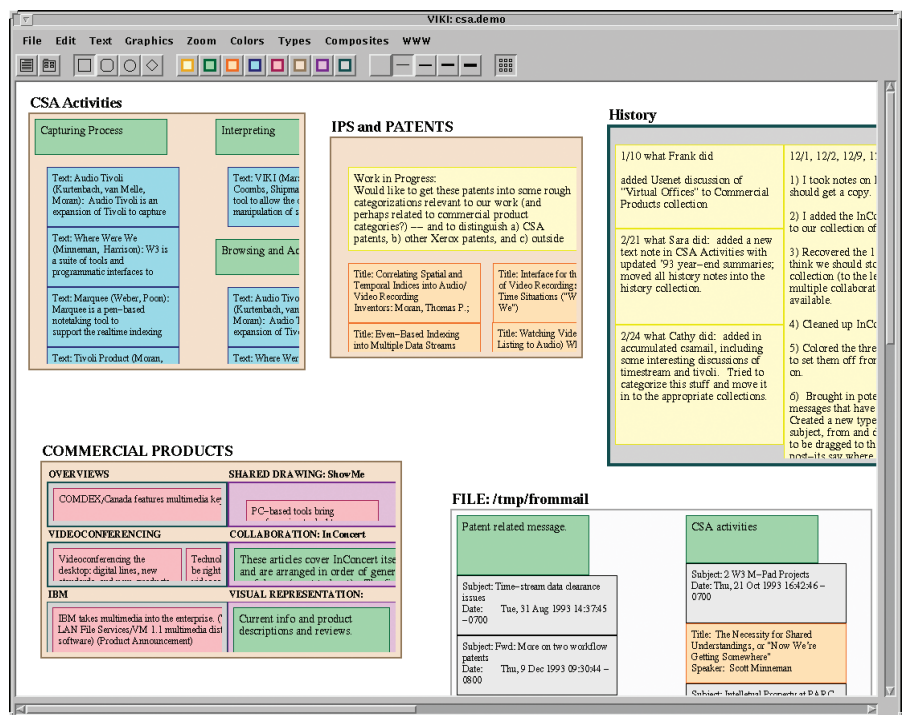
b <http://jena.sourceforge.net>.  
c <http://www.w3.org/TR/xquery>.

age) has produced numerous relevant results here. Buneman et al.<sup>8</sup> devised a system for tracking user's browsing and collection activities so they can be queryable later on. They accomplished this by supplementing the user's personal database with a separate provenance database that links items to their original sources, and to previous versions within the local database. The Trio system<sup>39</sup> also automatically tracks when and how data items came to exist, whether imported from outside sources, or computed from other known facts. This allows a history of each item to be reconstructed, and the database to be selectively filtered based on source or time information. Bhagwat et al.<sup>5</sup> specifically studied the propagation of annotations, so that as data evolves over time source data can be recovered. These techniques are applicable to PKB implementation as well, to enable users to browse and collect information and know that source information will automatically be tracked.

### Data Models

Kaplan et al. stated it well when they observed in 1990 that "dominant database management paradigms are not well suited for managing personal data," since "personal information is too ad hoc and poorly structured to warrant putting it into a record-oriented online database."<sup>25</sup> Clearly this is the case; when we want to jot down and preserve a book recommendation, directions to a restaurant, or scattered lecture notes, a rigidly structured relational database table is exactly the wrong prescription. The random information we collect defies categorization and quantization, and yet it demands some sort of structure, both to match the organized fashion in which we naturally think and to facilitate later retrieval. The question is, what sort of data model should a PKB provide?

A few definitions are in order. First, we will use the term "knowledge element" to refer to the basic building blocks of information that a user creates and works with. Most systems restrict knowledge elements to be simple words, phrases, or concepts, although some (especially note-taking systems) permit larger blocks of free text, which may even include hyper-



**Figure 5. VIKI, one of the first spatial hypertext systems. Rather than links between elements, the primary way organizational information is conveyed is through spatial clustering.**

links to external documents. Second, the term "structural framework" will cover the rules about how these knowledge elements can be structured and interrelated.

This section presents and critiques the five principal PKB structural frameworks (tree, graph, tree plus graph, spatial, and category.) The vast majority of PKB tools are based on one of these five principal frameworks, although a handful of alternates have been proposed (for example, Lifestreams' chronological approach<sup>17</sup> and Aquanet's n-ary relations.<sup>27</sup>) I will then give particular attention to Ted Nelson's ZigZag paradigm,<sup>32</sup> a more flexible model than any of these five whose expressive power can subsume all of them. Later, key characteristic of transclusion and its influence on the various frameworks will be addressed.

### The Five Primary Structural Frameworks

**Tree.** Systems that support a tree model allow knowledge elements to be organized into a containment hierarchy, in which each element has one and only one "parent." This takes advantage of the mind's natural tendency to classify objects into groups, and to further break up each classification into

subclassifications.

All of the applications for creating mind maps are based on a tree model, because a mind map *is* a tree. And most of the "notebook-based" note-taking systems use a tree model by allowing users to partition their notes into sections and subsections. Some tools extend this paradigm by permitting "crosslinks" between items (or Mind Manager's "floating topics," which are not anchored to the hierarchy.) The fact that such features are included betrays the inherent limitations of the strict tree as a modeling technique: it is simply inadequate for representing much complex information.

**Graph.** Graph-based systems—including hypertext systems and concept-mapping tools—allow users to create knowledge elements and then to interconnect them in arbitrary ways. In many systems, links between items can optionally be labeled with a word or phrase indicating the nature of the relationship, and adorned with arrowheads on one or both ends to indicate navigability.

An alluring feature of the graph data model is that it is essentially equivalent to a "semantic network,"<sup>3,40</sup> believed by many psychologists to be an excellent model for human



memory. Just as humans perceive the world in terms of concepts and the relationships between them, so a graph depicts a web of interconnected entities. The ideal PKB would supplement this model with alternative retrieval mechanisms (such as full-text indexing, or suggesting “similar items” the user has not explicitly linked) so as to compensate for the human mind’s shortcomings. But if a primary goal of a PKB is to capture the essence of a human’s thoughts, then using a graph data model as the foundation is powerfully attractive.

**Tree plus graph.** Although graphs are a strict superset of trees, trees offer some important advantages in their own right: simplicity, familiarity, ease of navigation, and the ability to conceal details at any level of abstraction. Indeed, the problem of “disorientation” in hypertext navigation<sup>11,26</sup> largely disappears with the tree model; one is never confused about “where one is” in the larger structure, because traversing the parent hierarchy gives the context of the larger surroundings. For this reason, several graph-based systems have incorporated special support for trees as well, to combine the advantages of both approaches.

One of the earliest systems to combine tree and graph primitives was TEXTNET, which featured two types of nodes: “chunks” (that contained content to be browsed and organized) and “table of contents” nodes (or “tocs.”)

Any node could freely link to any other, permitting an unrestricted graph. But a group of tocs could be combined to form a tree-like hierarchy that bot-tomed out in various chunk nodes. In this way, any number of trees could be superimposed upon an arbitrary graph, allowing it to be viewed and browsed as a tree, with all the requisite advantages.

**Spatial.** In the opposite direction, some designers have shunned links between elements altogether, favoring instead spatial positioning as the sole organizational paradigm. Capitalizing on the human’s tendency to implicitly organize through clustering, making piles, and spatially arranging, some tools offer a 2D workspace for placing and grouping items. This provides a less formal (and perhaps less intimi-dating) way for a user to gradually in-troduce structure into a set of items as it is discovered.

This approach originated from the spatial hypertext community, demon-strated in projects like VIKI/VKB<sup>28</sup> (see Figure 5). With these programs, users place information items on a canvas and can manipulate them to convey organization imprecisely. VIKI and VKB are especially notable for their ability to automatically infer the struc-ture from a user’s freeform layout: a spatial parser examines which items have been clustered together, colored or otherwise adorned similarly, and so on, and makes judgments about how

to turn these observations into ma-chine-processible assertions.

Certain note-taking tools (for exam-ple, Microsoft OneNote) also combine an overarching tree structure with spa-tial freedom on each “page.” Users can access a particular page of the note-book with basic search or tree naviga-tion facilities, and then lay out notes and images on the page as desired. Tinderbox,<sup>4</sup> in addition to supporting the graph model, also makes heavy use of the spatial paradigm, which lets users express less formal affinities be-tween items.

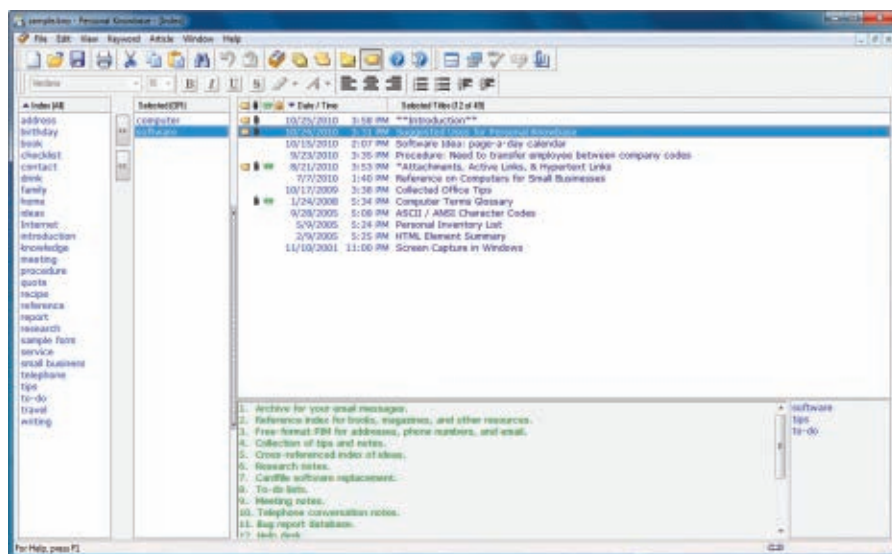
**Category.** The fifth fundamental structural framework that PKB sys-tems use is that of categories. Users may think of categories as collections, in which the category somehow en-closes or “owns” the items within it. Alternatively, they may think of label-ing items with custom-defined key-words, thereby implicitly creating a category. The important point is that a given item can be simultaneously present in multiple categories, reliev-ing the tree model’s most restrictive constraint.

The first popular application to em-brace the category approach was the original Agenda which later became a commercial product and spawned many imitations. Personal Knowbase (<http://bitsmithsoft.com>; see Figure 6), Haystack, and Chandler (<http://osafoundation.org>) are more modern examples.

## ZigZag

We treat ZigZag<sup>32</sup> separately from the five common models since it is so unique, and represents a paradigm shift in knowledge modeling. Its core idea is very simple: knowledge ele-ments can be related to one another sequentially along any number of di-mensions.

In some ways, ZigZag is an exten-sion of the structure of an ordinary spreadsheet: just as in a spreadsheet, each cell is positioned in sequence relative to other cells both horizon-tally and vertically, ZigZag allows a cell to participate in many such sequences. This seemingly simple extension actu-ally has broad impact: it turns out to be more general than any of the pre-vious five models, and each of them can be expressed by it. Its principal



**Figure 6.** Personal Knowbase, a note-taking system based on the category structural framework. Multiple customized user keywords (listed in far-left pane) can be assigned to each note (comprised of a small text document and a title). These keywords can be combined with Boolean operations in order to retrieve notes.

liability has been difficulty of understanding: users (and even researchers) steeped in the traditional paradigms sometimes struggle to break free of old assumptions. Yet if adopted on a wide scale, ZigZag's so-called "hyperthogonal" structure offers the possibility of an ultra-flexible PKB, capable of adapting to all of a user's needs.

### The Role of Transclusion

The term "transclusion," first coined by Ted Nelson,<sup>31</sup> has been used in several senses. In general it means including an excerpt from one document into another, such that the including document maintains some kind of reference to the included document. The simplest form of transclusion would be a simply copy-and-paste operation wherein a link to the original source was maintained. A stronger form is when the transcluded content is not copied, but referenced. This can allow any updates to the referred-to document to be instantly seen by the referring one, or, in an even more sophisticated scheme, it allows the referring document to maintain access to the transcluded content as it originally appeared, and also any more recent versions of it. (The Xanadu project design was based on this latter formulation.)

In the context of PKBs, transclusion means the ability to view the same knowledge element (not a copy) in multiple contexts. It is so pivotal because it is central to how the human mind processes information. We think associatively, and with high fan-out. I may consider John, for instance, as a neighbor, a fellow sports enthusiast, a father of small children, a Democrat, an invitee to a party, and a homeowner who owns certain power tools, all at once. Each of these different contexts places him in relationship to a different set of elements in my mind. Without delving into psychological research to examine exactly how the mind encodes such associations, it seems clear that if we are to build a comprehensive personal knowledge base containing potentially everything a person knows, it must have the ability to transclude knowledge elements.

Bush's original design of the memex explicitly prescribed the transclusion concept, for instance in his notion of a "skip trail." "The historian,"

he writes, "with a vast chronological account of a people, parallels it with a skip trail that stops only at the salient items." In this way, the full account of the subject can be summarized in a sort of digest that refers to select items from the original, larger trail. A modern example of transclusion is Mediawiki,<sup>d</sup> the software used to host, among other sites, Wikipedia. Its use of template tags permits a source page's current contents to be dynamically included and embedded within another page.

Adding transclusion to the tree model effectively turns it into a directed acyclic graph (DAG), in which an item can have multiple parents. This is what Trigg and Halasz achieved with their extensions to the tree model.<sup>23,37</sup>

A similar mechanism can be applied to graph models, as with Tinderbox's "alias" feature. In Tinderbox, information is broken up into "notes," which can appear on the screen as spatially laid out rectangles with links between them. By creating an "alias" for a note, one can summon its appearance on a different graph layout than the note originally appeared. Compendium also allows its nodes to be present on multiple views, and the Popcorn data model<sup>13</sup> was based entirely on transclusion. This seems closer to how the mind operates: we associate ideas with contexts, but we do not embed ideas irreversibly into the first context we happened to place them in. Tightly binding an element to its original context, therefore, seems like the wrong approach.

### Architecture

The idea of a PKB gives rise to some important architectural considerations. While not constraining the nature of what knowledge can be expressed, the architecture nevertheless affects matters such as availability and workflow.

**File based.** The vast majority of solutions mentioned in this article use a simple storage mechanism based on flat files in a file system. This is true of virtually all the mind-mapping tools, concept-mapping tools, and note-taking tools, and even a number of hypertext tools (for example, NoteCards, Tinderbox). Typically, the main "unit"

of a user's knowledge design—whether that be a mind map, a concept map, an outline, or a "notebook"—is stored in its own file somewhere in the file system. The application can find and load such files via the familiar "File | Open..." paradigm, at which point it typically maintains the entire knowledge structure in memory.

This approach takes advantage of the average user's familiarity with file "open" and "save" operations, but does have ramifications on its utility as a PKB. Users must choose one of two basic strategies: either store all of their knowledge in a single file; or else break up their knowledge and store it across a number of different files, presumably according to subject matter and/or time period. The first choice can result in insurmountable scalability problems if the system is heavily used, while the second may force an unnatural partitioning of topics and an inability to link disparate items together.

**Database based.** Architectures involving a database to store user knowledge address these concerns. Knowledge elements reside in a global space, which allows any idea to relate to any other: now a user can relate a book he read on productivity not only to other books on productivity, but also to "that hotel in Orlando that our family stayed in last spring," because that is where he remembers having read the book. Though such a relationship may seem "out of bounds" in traditional knowledge organization, it is exactly the kind of retrieval path that humans often employ in retrieving memories.<sup>3</sup>

Agenda<sup>25</sup> and gIBIS<sup>12</sup> were two early tools that incorporated a relational database backend in their architecture. More generally, the issues surrounding storage of graph-based data in relational databases<sup>6,18</sup> or in special-purpose databases<sup>30</sup> have received much attention, giving PKB designers ample guidance for how to architect their persistence mechanism.

**Client server.** Decoupling the actual knowledge store from the PKB user interface can achieve architectural flexibility. As with all client-server architectures, the benefits include load distribution, platform interoperability, data sharing, and ubiquitous availability. Increased complexity and la-

<sup>d</sup> <http://www.mediawiki.org>

tency are among the liabilities, which can indeed be considerable factors in PKB design. Examples of client-server PKBs include MyBase Networking Edition (<http://wjsoft.com>), and Haystack's three-tiered architecture.<sup>1</sup>


A variation of the client-server approach is of course Web-based systems, in which the client system consists of nothing but a (possibly enhanced) browser. This gives the same ubiquitous availability that client-server approaches do, while minimizing (or eliminating) the setup and installation required on each client machine.

**Handheld devices.** Lastly, we mention mobile devices as a possible PKB architecture. Storing all of one's personal knowledge on a handheld computer would solve the availability problem, of course, and even more completely than would a client-server or Web-based architecture. The safety of the information is an issue, since if the handheld device were to be lost or destroyed, the user could face irrevocable data loss; this is easily remedied, however, by periodically synchronizing the handheld device's contents with a host computer. More problematic is simply the limitations of the hardware. Screen real estate, processing power, and storage capacity are of course much more limited, and this hampers their overall effectiveness.


### The PKB of the Future

Personal knowledge management is a real and pressing problem, as the sheer number of products included in this article attests. Yet it does not appear that Vannevar Bush's dream has yet been fully realized on a wide scale. Nearly every system mentioned here has its circle of loyal adherents ("I find Tinderbox indispensable for my work and every update makes it that much more mind-blowing."<sup>e</sup> "The Greatest Invention in Human History? I vote for Microsoft OneNote."<sup>f</sup>) But certainly when compared with word processors, spreadsheets, or Web browsers, PKB usage lags far behind.

What would it take for a true PKB solution to appeal to a wide audience



**The idea of a PKB gives rise to some important architectural considerations. While not constraining the nature of what knowledge can be expressed, the architecture nevertheless affects matters such as availability and workflow.**



and generate the kinds of benefits Bush envisioned? Synthesizing lessons from the analysis here, the following recommendations for future research seem apparent:

- A PKB data model should support transclusion in some form. Allowing elements to appear in multiple contexts is simply the only way to faithfully capture human knowledge.

- Graph, tree, category, and spatial paradigms should probably be combined, so as to leverage the advantages of each. Human beings think in each of these ways, and the claustrophobic effect of some knowledge tools stems from trying to force its users to think in only one of them. An ideal PKB should allow users to express their knowledge as an arbitrary graph (semantic network), while also building containment hierarchies and specifying categories as desired to express those naturally occurring relationships.

- The most suitable architecture is probably a knowledge server with a database backend that supports both desktop and handheld clients. This is because people form knowledge, and need to retrieve it, in many different settings. Locating a user's knowledge base on, say, a single desktop computer would mean their knowledge is available to them only in one particular setting: say at work, or at home, and only while at their desk. This limits the efficacy of the PKB by compartmentalizing it.

- Users must be able to associate any two knowledge elements anywhere in their PKB. This calls for either a single unified database, or else some kind of metastructure overlaid on individual files that allows their inner contents to be referred to. The latter approach is taken by the iDM model<sup>14</sup> that represents a user's entire personal dataspace.<sup>19</sup> iDM maps structural information both inside and outside files into a single graph, which can be coherently queried, eliminating the problem of artificial partitioning between disparate files.

- PKB interfaces should make it easy to assimilate "snippets" from the objective information sources a user encounters. As the research of Anderson and others has shown, our knowledge consists largely of bits and pieces of information that we have gathered

<sup>e</sup> <http://www.eastgate.com/Tinderbox/news.html>

<sup>f</sup> <http://www.c3associates.com/2007/04/the-greatest-invention-in-human-history/>



from diverse sources, synthesized into a mental framework that allows us to make sense of it. As the user is materializing their framework explicitly in a PKB tool, it should be easy to grab the short excerpts that they find relevant and import them into the knowledge base in a painless way. This is equivalent to the simpler form of transclusion, as defined previously.

► Standardization between PKB vendors needs to take place so that a user's knowledge is not inextricably bound to a particular tool. A PKB should conform to these standards through some kind of import/export facility.

► Finally, researchers should take a good look at the tools that *have* been adopted on a wide scale—blogs or wikis, for instance. Though heavily based on free text rather than richly interconnected knowledge elements, one of their functions can be seen as makeshift knowledge management. Perhaps the best route to a successful PKB would be to take advantage of the broad adoption of such tools and enhance them with the capability of expressing knowledge in a more structured form.

To pull all these ideas together, imagine a distributed system that securely stores your personal knowledge and is available to you anywhere, anytime: from any computer, or from a handheld device that you always carry with you. Furthermore, the knowledge it contains is in a flexible form that can readily accommodate your very thoughts. It contains all the concepts you have perceived in the past and want to recall—historical events, business plans, phone numbers, scientific formulas—and does not encourage you to isolate them from one another, or to prematurely commit to a structure that you might find restrictive later. The concepts can be linked together as in a graph, clustered visually on canvases, classified in multiple categories, and/or arranged hierarchically, all for different purposes. As further information is encountered—from reading documents, brainstorming project plans, or just experiencing life—it is easy to assimilate into the tool, either by capturing snippets of text and relating it to what is already known, or by creating new concepts and combining them with the easily

retrievable old. External documents can be linked into the knowledge structure in key places, so that they can be classified and easily retrieved. It is effortless to augment the content with annotations, and to rearrange it to reflect new understandings. And this gold mine of knowledge is always exportable in a form that is compatible with other, similar systems that have different features and price points.

Such a tool would surely be a boon to anyone who finds their own mind to be insufficient for retaining and leveraging the knowledge they acquire. As Vannevar Bush stirringly wrote: “Presumably man’s spirit should be elevated if he can better review his shady past and analyze more completely and objectively his present problems.” ■

## References

- Adar, E., Karger, D. and Stein, L.A. Haystack: Per-user information environments. In *Proceedings of the 8th International Conference on Information Knowledge Management*, (Kansas City, MO, 1999), 413–422.
- Amer-Yahia, S., Botev, C., Dörre, J., and Shanmugasundaram, J. XQuery full-text extensions explained. *IBM Systems Journal* 45 (2006), 335–351.
- Anderson, J.R. *Cognitive Psychology and Its Implications*, 3rd Ed. W.H. Freeman, NY, 1990.
- Bernstein, M., Collages, Composites, Construction. In *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia*, (Nottingham, U.K., 2003).
- Bhagwat, D., Chiticariu, L., Tan, W.C. and Vijayvargiya, G. An annotation management system for relational databases. *The Intern. J. Very Large Data Bases* 14 (2005), 373–396.
- Broekstra, J., Kampman, A. and Van Harmelen, F. Sesame: A generic architecture for storing and querying rdf and rdf schema. *Lecture Notes in Computer Science*, (2002), 54–68.
- Buneman, P., Fernandez, M., and Suciu, D. UnQL: A query language and algebra for semistructured data based on structural recursion. *The VLDB Journal* 9 (2000), 76–110.
- Buneman, P., Chapman, A., and Cheney, J. Provenance management in curated databases. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, ACM, NY, 539–550.
- Bush, V. As we may think. *The Atlantic Monthly* 176, 1 (July 1945), 101–108.
- Canas, A.J., Hill, G., Carff, R., Suri, N., Lott, J., Gomez, G., Eskridge, T.C., Arroyo, M. and Carvajal, R. CmapTools: A knowledge modeling and sharing environment. In *Proceedings of the 1st International Conference on Concept Mapping*, (Pamplona, Spain, 2005), 125–133.
- Conklin, J. Hypertext: An introduction and survey. *Computer* 20, 9 (Sept. 1987), 17–41.
- Conklin, J. and Begeman, M.L. gIBIS: A hypertext tool for exploratory policy discussion. In *Proceedings of the 1988 ACM Conference on Computer-supported Cooperative Work*, (Portland, OR), 140–152.
- Davies, S., Allen, S., Raphaelson, J., Meng, E., Engleman, J., King, R., and Lewis, C. Popcorn: The Personal Knowledge Base. In *Proceedings of the 6th ACM Conference on Designing Interactive Systems* (2006), 150–159.
- Dittrich, J.P. and Salles, M.A.V. iDM: A unified and versatile data model for personal dataspace management. In *Proceedings of the 32nd International Conference on Very Large Data Bases* (2006), 367–378.
- Engelbart, D.C. A conceptual framework for the augmentation of man's intellect. P.W. Howerton, ed. *Vistas in Information Handling*. Spartan Books, Washington, D.C., 1963, 1–29.
- Fernandez, M., Florescu, D., Levy, A. and Suciu, D. A query language for a web-site management system. *ACM SIGMOD Record* 26 (1997), 4–11.
- Fertig, S., Freeman, E. and Gelernter, D. Lifestreams: An alternative to the desktop metaphor. In *Proceedings of the Conference on Human Factors in Computing Systems* (Vancouver, B.C., 1996), 410–411.
- Florescu, D., Inria, R. and Kossmann, D. Storing and querying XML data using an RDBMS. *IEEE Data Engineering Bulletin* 22, 3 (1999).
- Franklin, M., Halevy, A. and Maier, D. From databases to dataspace: A new abstraction for information management. *ACM SIGMOD Record* 34 (2005), 27–33.
- Gantz, J. The Diverse and Exploding Digital Universe. White paper. International Data Corporation, Framingham, MA, (Mar. 2008); <http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf>.
- Gentner, D. and Stevens, A.L., eds. *Mental Models*. Lawrence Erlbaum Assoc., NJ, 1983.
- Goodman, D. *The Complete HyperCard Handbook*. Bantam Books, NY, 2003.
- Halasz, F.G., Moran, T.P. and Trigg, R.H. NoteCards in a Nutshell. *ACM SIGCHI Bulletin* 17 (1987) 45–52.
- Hayes, G., Pierce, J.S. and Abowd, G.D. Practices for capturing short important thoughts. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, (Ft. Lauderdale, FL, 2003), 904–905.
- Kaplan, S.J., Kapor, M.D., Belove, E.J., Landsman, R.A. and Drake, T.R. Agenda: A personal information manager. *Commun. ACM* 33, 7 (July 1990), 105–116.
- Mantei, M.M. Disorientation behavior in person-computer interaction. *Communications Department*, University of Southern California, 1982.
- Marshall, C., Halasz, F.G., Rogers, R.A. and Janssen, W.C. Aquanet: A hypertext tool to hold your knowledge in place. In *Proceedings of the 3rd Annual ACM Conference on Hypertext*, (San Antonio, TX, 1991), 261–275.
- Marshall, C. and Shipman, F. Spatial hypertext: designing for change. *Commun. ACM* 38, 8 (Aug.1995), 88–97.
- McBrien, P. and Poulouvasilis, A. A semantic approach to integrating XML and structured data sources. *Proceedings of the 13th International Conference on Advanced Information Systems Engineering*. Springer-Verlag, London, U.K, 2001, 330–345.
- McHugh, J., Abiteboul, S., Goldman, R., Quass, D., and Widom, J. Lore: A database management system for semistructured data. *ACM SIGMOD Record* 26 (1997), 54–66.
- Nelson, T.H. The heart of connection: Hypermedia unified by transclusion. *Commun. ACM* 38, 8 (Aug. 1985), 31–33.
- Nelson, T.H. A Cosmology for a different computer universe: Data model, mechanisms, virtual machine and visualization infrastructure. *J. Digital Information* 5, 1 (2006).
- Nelson, T.H. Xanalogical structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use. *ACM Computing Surveys* 31, 4 (1999).
- Novak, J.D. The theory underlying concept maps and how to construct them. *Institute for Human and Machine Cognition*, University of West Florida, 2003.
- Papakonstantinou, Y., Garcia-Molina, H., and Widom, J. Object exchange across heterogeneous information sources. In *Proceedings of the 11th International Conference on Data Engineering* (1995), 251–260.
- Prud'Hommeaux, E. and Seaborne, A. SPARQL query language for RDF. *W3C working draft*, 4, (2006).
- Trigg, R.H. and Weiser, M. TEXTNET: A network-based approach to text handling. *ACM Trans. Info. Systems* 4, 1 (1986), 1–23.
- Vdovjak, R. and Houben, G.J. RDF based architecture for semantic integration of heterogeneous information sources. In *Proceedings of the Workshop on Information Integration on the Web* (2001), 51–57.
- Widom, J. Trio: A system for integrated management of data, accuracy, and lineage. In *Proceedings of the Conference on Innovative Data Systems Research* (2005).
- Woods, W.A. What's in a link: Foundations for semantic networks. In R.J. Brachman and J. Levesque, eds. *Readings in Knowledge Representation*, Morgan Kaufmann, 1985.

**Stephen Davies** (stephen@umw.edu) is an assistant professor of computer science at the University of Mary Washington, Fredericksburg, VA.

# research highlights

---

P. 90

## **Technical Perspective Markov Meets Bayes**

By Fernando Pereira

P. 91

## **The Sequence Memoizer**

By Frank Wood, Jan Gasthaus, Cédric Archambeau, Lancelot James, and Yee Whye Teh

---

P. 99

## **Technical Perspective DRAM Errors in the Wild**

By Norman P. Jouppi

P. 100

## **DRAM Errors in the Wild: A Large-Scale Field Study**

By Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber

# Technical Perspective

## Markov Meets Bayes

By Fernando Pereira

PROBABILISTIC SEQUENCE MODELS have become so pervasive in science and engineering that we often act as if the sequence modeling problem is basically solved. On the Internet, such models power the large-scale systems for statistical machine translation, such as Google Translate, that receive millions of queries each day. These systems use a probabilistic sequence model—a language model—to select among candidate translations in the target language. For example, consider translating the English phrases “to show interest” and “to charge interest” into Portuguese. In the first phrase, the word “interest” should be translated as “interesse” (curiosity) while in the second it should be translated as “juro” (return on lending). In each case, a Portuguese language model would assign higher probability to the correct translation based on the context established by the preceding verb. Today, such language models are derived from the statistics of large corpora with as many as  $10^{10}$  sentences of text. Moreover, in practical systems for machine translation or speech recognition, they may assign probabilities to sequences that contain as many as  $10^6$  distinct words.


The history of probabilistic sequence models, too long and complex to review here, dates back to Markov at the turn of the last century. To a first approximation, today’s practice is to memorize a subset of  $cw$  patterns occurring in the training data, where  $w$  is a token (for example, Unicode point, DNA base, word) and  $c$  is a context of preceding tokens. The statistics of these patterns are then used to estimate the probability of a token appearing after a given sequence of tokens. For modelers, the critical modeling questions are how to select which patterns to store, and from these patterns, how to estimate the probability that a token  $w$  follows a context  $c$  when  $w$  has never or rarely been seen in that context. Though informed by decades of research, cur-

rent practices are still something of a black art. They often reflect the particular data structures and algorithms used to create sequence models and, for large-scale systems, the need for distribution implementations. We practitioners seem for the most part resigned to these complications, although naturally we wonder if there are more elegant ways to manage the balance between memorizing old patterns and generalizing to new ones.

The Sequence Memoizer (SM) detailed in the following paper addresses this balance in a novel way by extending several previous ideas: suffix trees for storing prediction contexts of unbounded length;<sup>2</sup> factorable priors for integrating contexts of different lengths into a final probability estimate;<sup>5</sup> and nonparametric Bayesian methods for incorporating the uncertainty over which contexts best predict the next token.<sup>4</sup> The authors had the critical insight to combine a particular form of hierarchical Pitman-Yor process with standard techniques for linear-time, context-tree creation. This combination yields the first linear-time, unbounded-context sequence model based on principled Bayesian techniques.

How well does it work? On standard test collections using standard metrics, the SM matches or outperforms all previous methods that store the same context statistics. Why does it work so well? One likely reason is that the hierarchical Pitman-Yor process is better at modeling the *power law* statistics of natural sequence data: in many types of sequences, patterns that are individually rare still account for a substantial fraction of all observed patterns. By modeling these statistics, the SM is better able to integrate the predictions from contexts of different lengths.

The SM is an important advance in probabilistic sequence modeling that finally makes nonparametric Bayesian methods practical for large data sets. Is sequence modeling therefore

solved? Hardly. First, there remains a bit of black art in the SM, whose probability estimates cannot be computed in closed form but must instead be approximated with stochastic (Monte Carlo) methods for inference. Not only do these methods require some empirical tuning, but they also create a trade-off between predictive power and computational complexity. We lack any theoretical understanding of this trade-off in SMs when stochastic inference is taken into account. Finally, for many applications, long contiguous patterns of tokens seem a very inefficient representation of context. For instance, natural language text often has a hierarchical structure, where relatively distant words (say, the main verb and the head noun of the direct object) may be strongly correlated while the words in between are less correlated to either;<sup>1</sup> similarly, DNA sequences can exhibit strong correlations between bases in two consecutive exons but much weaker correlations with the bases in the intervening intron. Furthermore, in certain contexts it seems a reasonable approximation to interchange sets of tokens or token sequences that belong to particular syntactic or semantic classes.<sup>3</sup> An interesting question is whether more refined nonparametric Bayesian models can capture these statistical properties. 

### References

1. Chelba, C. A structured language model. In *Proceedings of the 35th Annual Meeting of the ACL* (Morristown, NJ, 1997), Assoc. Computational Linguistics, 498–500.
2. Cleary, J.G., and Teahan, W.J. Unbounded length contexts for PPM. *The Computer J.* 40 (1997), 67–75.
3. Lin, D. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning* (1998), 296–304; <http://dblp.uni-trier.de/>.
4. Mochihashi, D., and Sumita, E. The infinite Markov model. *Advances in Neural Information Processing Systems* 20 (2008), 1017–1024.
5. Willems, F., Shtarkov, Y., and Tjalkens, T. The context-tree weighting method: Basic properties. *IEEE Trans. Info. Theory* 41, 3 (May 1995), 653–664.

**Fernando Pereira** is research director at Google, Menlo Park, CA.

© 2011 ACM 0001-0782/11/0200 \$10.00



# The Sequence Memoizer

By Frank Wood, Jan Gasthaus, Cédric Archambeau, Lancelot James, and Yee Whye Teh

## Abstract

**Probabilistic models of sequences play a central role in most machine translation, automated speech recognition, lossless compression, spell-checking, and gene identification applications to name but a few. Unfortunately, real-world sequence data often exhibit long range dependencies which can only be captured by computationally challenging, complex models. Sequence data arising from natural processes also often exhibits power-law properties, yet common sequence models do not capture such properties. The sequence memoizer is a new hierarchical Bayesian model for discrete sequence data that captures long range dependencies and power-law characteristics, while remaining computationally attractive. Its utility as a language model and general purpose lossless compressor is demonstrated.**

## 1. INTRODUCTION

It is an age-old quest to predict what comes next in sequences. Fortunes have been made and lost on the success and failure of such predictions. Heads or tails? Will the stock market go up by 5% tomorrow? Is the next card drawn from the deck going to be an ace? Does a particular sequence of nucleotides appear more often than usual in a DNA sequence? In a sentence, is the word that follows the United going to be States, Arab, Parcel, Kingdom, or something else? Using a probabilistic model of sequences fit to a particular set of data is usually an effective way of answering these kinds of questions.

Consider the general task of sequence prediction. For some sequences, the true probability distribution of the next symbol does not depend on the previous symbols in the sequence. For instance, consider flipping a coin that comes up heads with probability  $p$  or tails with probability  $1 - p$  every time it is flipped. Subsequent flips of such a coin are completely independent and have the same distribution (in statistics, such coin flips are called *independent and identically distributed* [iid]). In particular, heads will occur with probability  $p$  irrespective of whether previous flips have come up heads or tails. Assuming we observe a sequence of such coin flips, all we need to do is to estimate  $p$  in order to fully characterize the process that generated the data.

For more interesting processes, the distribution of the next symbol often depends in some complex way on previous outcomes. One example of such a process is natural language (sequences of words). In English, the distribution of words that follow the single-word context United is quite different from the distribution of words that follow Cricket and rugby are amongst the most popular sports in the United. In the first case, the distribution is relatively broad (though not nearly as broad as the distribution given no context at all), giving significant probability to words such as States, Kingdom, Airlines, and so forth, whereas in the second case, the distribution is

almost certainly highly peaked around Kingdom. Information from distant context (Cricket and rugby) impacts the distribution of the next word profoundly. Production of natural language is but one example of such a process; the real world is replete with other examples.

Employing models that capture long range contextual dependencies will often improve one's ability to predict what comes next, as illustrated in the example above. Of course, modeling the distribution of the next symbol emitted by a process will only be improved by consideration of longer contexts if the generating mechanism actually does exhibit long range dependencies. Unfortunately, building models that capture the information contained in longer contexts can be difficult, both statistically and computationally. The sequence memoizer (SM) captures such long range dependencies in a way that is both statistically effective and scales well computationally.

While the SM and related models are useful for predicting the continuation of sequences, prediction is not the only application for these models. Automated speech recognition and machine translation require assessing the typicality of sequences of words (i.e., Is this sentence a probable English sentence?). Speaker or writer identification tasks require being able to distinguish typicality of phrases under word sequence models of different writers' styles. Classifying a sequence of machine instructions as malicious or not requires establishing the typicality of the sequence under each class. Models of sequences can be used for predicting the continuation of sequences, clustering or classifying sequences, detecting change points in sequences, filling in gaps, compressing data, and more.

In this article we describe the SM in terms of general sequences over a discrete alphabet of symbols, though often we will refer to sequences of words when giving intuitive explanations.

## 2. PREDICTING SEQUENCES

To start, let  $\Sigma$  be the set of symbols that can occur in some sequence. This set can consist of dictionary entries, ASCII values, or  $\{A, C, G, T\}$  in case of DNA sequences. Suppose that we are given a sequence<sup>a</sup>  $\mathbf{x} = x_1, x_2, \dots, x_T$  of symbols

The work reported in this paper originally appeared in "A Hierarchical Bayesian Language Model based on Pitman-Yor Processes," published in the *Proceedings of International Conference on Computational Linguistics and the Association for Computational Linguistics*, 2006; "A Stochastic Memoizer for Sequence Data," published in the *Proceedings of the International Conference on Machine Learning*, 2009 and "Lossless compression based on the Sequence Memoizer" published in the *Proceedings of the IEEE Data Compression Conference*, 2010.

from  $\Sigma$  and would like to estimate the probability that the next symbol takes on a particular value.

One way to estimate the probability that the next symbol takes some value  $s \in \Sigma$  is to use the relative frequency of its occurrence in  $\mathbf{x}$ , i.e., if  $s$  occurs frequently in  $\mathbf{x}$  we expect its probability of appearing next to be high as well. Assuming that  $\mathbf{x}$  is long enough, doing this will be better than giving equal probability to all symbols in  $\Sigma$ . Let us denote by  $N(s)$  the number of occurrences of  $s$  in  $\mathbf{x}$ . Our estimate of the probability of  $s$  being the next symbol is then  $G(s) = N(s)/T = N(s)/\sum_{s' \in \Sigma} N(s')$ . The function  $G$  is a *discrete distribution* over the elements of  $\Sigma$ : it assigns a non-negative number  $G(s)$  to each symbol  $s$  signifying the probability of observing  $s$ , with the numbers summing to one over  $\Sigma$ .

Of course, this approach is only reasonable if the process generating  $\mathbf{x}$  has no history dependence (e.g., if  $\mathbf{x}$  is produced by a sequence of tosses of a biased coin). It is highly unsatisfying if there are contextual dependencies which we can exploit. If we start accounting for context, we can quickly improve the quality of the predictions we make. For instance, why not take into account the preceding symbol? Let  $\mathbf{u}$  be another symbol. If the last symbol in  $\mathbf{x}$  is  $\mathbf{u}$ , then we can estimate the probability of the next symbol being  $s$  by counting the number of times  $s$  occurs after  $\mathbf{u}$  in  $\mathbf{x}$ . As before, we can be more precise and define

$$G_{\mathbf{u}}(s) = \frac{N(\mathbf{us})}{\sum_{s' \in \Sigma} N(\mathbf{us}')} \quad (1)$$

to be the estimated probability of  $s$  occurring after  $\mathbf{u}$ , where  $N(\mathbf{us})$  is the number of occurrences of the subsequence  $\mathbf{us}$  in  $\mathbf{x}$ . The function  $G_{\mathbf{u}}$  is again a discrete distribution over the symbols in  $\Sigma$ , but it is now a *conditional distribution* as the probability assigned to each symbol  $s$  depends on the context  $\mathbf{u}$ .

In the hope of improving our predictions, it is natural to extend this counting procedure to contexts of length greater than one. The extension of this procedure to longer contexts is notationally straightforward, requiring us only to reinterpret  $\mathbf{u}$  as a sequence of length  $n \geq 1$  (in fact, for the remainder of this article boldface type variables will indicate sequences, and we will use  $\Sigma^*$  to denote the set of all finite sequences). Unfortunately, using this exact procedure for estimation with long contexts leads to difficulties, which we will consider next.

### 3. MAXIMUM LIKELIHOOD

Some readers may realize that the counting procedure described above corresponds to a ubiquitous statistical estimation technique called *maximum likelihood* (ML) estimation. The general ML estimation setup is as follows: we observe some data  $\mathbf{x}$  which is assumed to have been generated by some underlying stochastic process and wish to estimate parameters  $\Theta$  for a probabilistic model of this process. A probabilistic model defines a distribution  $P(\mathbf{x}|\Theta)$  over  $\mathbf{x}$  parameterized by  $\Theta$ , and the ML estimator is the value of  $\Theta$  maximizing  $P(\mathbf{x}|\Theta)$ . In our

<sup>a</sup> It is straightforward to consider multiple sequences in our setting, we consider being given only one sequence in this paper for simplicity.

case, the data consists of the observed sequence, and the parameters are the conditional distributions  $G_{\mathbf{u}}$  for some set of  $\mathbf{u}$ 's.

In a sense, ML is an *optimistic* procedure, in that it assumes that  $\mathbf{x}$  is an accurate reflection of the true underlying process that generated it, so that the ML parameters will be an accurate estimate of the true parameters. It is this very optimism that is its Achilles heel, since it becomes overly confident about its estimates. This situation is often referred to as *overfitting*. To elaborate on this point, consider the situation in which we have long contexts. The denominator of (1) counts the number of times that the context  $\mathbf{u}$  occurs in  $\mathbf{x}$ . Since  $\mathbf{x}$  is of finite length, when  $\mathbf{u}$  is reasonably long, the chance that  $\mathbf{u}$  never occurs at all in  $\mathbf{x}$  can be quite high, so (1) becomes undefined with a zero denominator. More pernicious still is if we are “lucky” and  $\mathbf{u}$  did occur once or a few times in  $\mathbf{x}$ . In this case (1) will assign high probability to the few symbols that just by chance did follow  $\mathbf{u}$  in  $\mathbf{x}$ , and zero probability to other symbols. Does it mean that these are the only symbols we expect to see in the future following  $\mathbf{u}$ , or does it mean that the amount of data we have in  $\mathbf{x}$  is insufficient to characterize the conditional distribution  $G_{\mathbf{u}}$ ? Given a complex process with many parameters the latter is often the case, leading to ML estimates that sharpen far too much around the exact observations and don't reflect our true uncertainty.

Obviously, if one uses models that consider only short context lengths, this problem can largely be avoided if one has enough data to estimate some (relatively) smaller number of conditional distributions. This is precisely what is typically done: one makes a *fixed-order Markov assumption* and restricts oneself to estimating collections of distributions conditioned on short contexts (for instance, an  $n$ th-order Markov model, or an  $m$ -gram language model). The consequence of doing this is that ML estimation becomes feasible, but longer-range dependencies are discarded. By assumption and design, they cannot be accounted for by such restrictive models.

Even having imposed such a restriction, overfitting often remains an issue. This has led to the development of creative approaches to its avoidance. The language modeling and text compression communities have generally called these *smoothing* or *back-off* methodologies (see Chen and Goodman<sup>3</sup> and references therein). In the following, we will propose a *Bayesian* approach that retains uncertainty in parameter estimation and thus avoids overconfident estimates.

### 4. BAYESIAN MODELING

As opposed to ML, the Bayesian approach is inherently *conservative*. Rather than trusting the data fully, Bayesian parameter estimation incorporates both evidence from the data as well as from prior knowledge of the underlying process. Furthermore, uncertainty in estimation is taken into account by treating the parameters  $\Theta$  as random, endowed with a *prior distribution*  $P(\Theta)$  reflecting the prior knowledge we have about the true data generating process. The prior distribution is then combined with the likelihood  $P(\mathbf{x}|\Theta)$  to yield, via *Bayes' Theorem* (the namesake of the approach),

the *posterior distribution*  $P(\Theta|\mathbf{x}) = P(\Theta)P(\mathbf{x}|\Theta)/P(\mathbf{x})$ , which specifies the belief about the parameter  $\Theta$  after combining both sources of information. Computations such as prediction are then done taking into account the *a posteriori* uncertainty about the underlying parameters.

What kinds of prior knowledge about natural sequence data might we wish to employ? We make use of two: that natural sequence data often exhibits power-law properties, and that conditional distributions of similar contexts tend to be similar themselves, particularly in the sense that recency matters. We will consider each of these in turn in the rest of this section.

#### 4.1. Power-law scaling

As with many other natural phenomena like social networks and earthquakes, occurrences of words in a language follow a *power-law scaling*.<sup>23</sup> This means that there are a small number of words that occur disproportionately frequently (e.g., the, to, of), and a very large number of rare words that, although each occurs rarely, when taken together make up a large proportion of the language. The power-law scaling in written English is illustrated in Figure 1. In this subsection we will describe how to incorporate prior knowledge about power-law scaling in the true generative process into our Bayesian approach. To keep the exposition simple, we will start by ignoring contextual dependencies and instead focus only on one way of estimating probability distributions that exhibit power-law scaling.

To see why it is important to incorporate knowledge about power-law scaling, consider again the ML estimate given by the relative frequency of symbol occurrences  $G(s) = N(s) / \sum_{s' \in \Sigma} N(s')$ . For the frequently occurring symbols, their corresponding probabilities will be well estimated since they are based on many observations of the symbols. On the other hand, our estimates of the rare symbol probabilities will

not be good at all. In particular, if a rare symbol did not occur in our sequence (which is likely), our estimate of its probability will be zero, while the probability of a rare symbol that did occur just by chance in our sequence will be overestimated. Since most symbols in  $\Sigma$  will occur quite rarely under a power-law, our estimates of  $G(s)$  will often be inaccurate.

To encode our prior knowledge about power-law scaling, we use a prior distribution called the Pitman–Yor process (PYP),<sup>15</sup> which is a distribution over the discrete probability distribution  $G = \{G(s)\}_{s \in \Sigma}$ . It has three parameters: a *base distribution*  $G_0 = \{G_0(s)\}_{s \in \Sigma}$ , which is the mean of the PYP and reflects our prior belief about the frequencies of each symbol, a *discount* parameter  $\alpha$  between 0 and 1 which governs the exponent of the power-law, and a *concentration* parameter  $c$  which governs the variability around the mean  $G_0$ . When  $\alpha = 0$ , the PYP loses its power-law properties and reduces to the more well-known Dirichlet process. In this paper, we assume  $c = 0$  instead for simplicity; see Gasthaus and Teh<sup>6</sup> for the more general case when  $c$  is allowed to be positive. When we write  $G \sim \text{PY}(\alpha, G_0)$  it means that  $G$  has a prior given by a PYP with the given parameters. Figure 1 illustrates the power-law scaling produced by PY processes.

To convey more intuition about the PYP we can consider how using it affects our estimate of symbol frequencies. Note that in our Bayesian framework  $G$  is random, and one of the standard steps in a procedure called *inference* is to estimate a posterior distribution  $P(G|\mathbf{x})$  from data. The probability that symbol  $s \in \Sigma$  occurs next is then:

$$P(x_{T+1} = s | \mathbf{x}) = \int P(x_{T+1} = s | G) P(G | \mathbf{x}) dG = \mathbb{E}[G(s)], \quad (2)$$

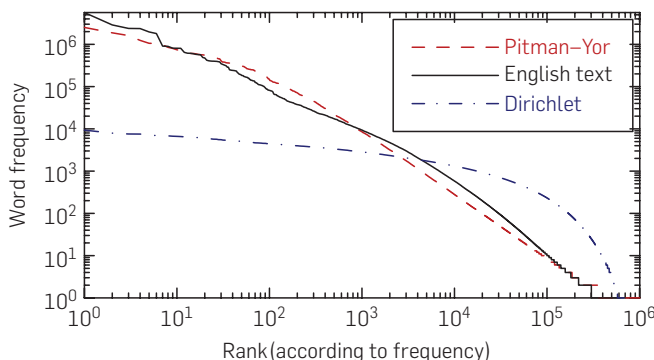
where  $\mathbb{E}$  in this case stands for expectation with respect to the posterior distribution  $P(G|\mathbf{x})$ . This integral is a standard Bayesian computation that sometimes has an analytic solution but often does not. When it does not, like in this situation, it is often necessary to turn to numerical integration approaches, including sampling and Monte Carlo integration.<sup>16</sup>

In the case of the PYP,  $\mathbb{E}[G(s)]$  can be computed as described at a high level in the following way. In addition to the counts  $\{N(s')\}_{s' \in \Sigma}$ , assume there is another set of random “counts”  $\{M(s')\}_{s' \in \Sigma}$  satisfying  $1 \leq M(s') \leq N(s')$  if  $N(s') > 0$  and  $M(s') = 0$  otherwise. The probability of symbol  $s \in \Sigma$  occurring next is then given by:

$$\mathbb{E}[G(s)] = \mathbb{E} \left[ \frac{N(s) - \alpha M(s) + \sum_{s' \in \Sigma} \alpha M(s') G_0(s)}{\sum_{s' \in \Sigma} N(s')} \right]. \quad (3)$$

Given this, it is natural to ask what purpose these  $M(s)$ ’s serve. By studying Equation 3, it can be seen that each  $M(s)$  reduces the count  $N(s)$  by  $\alpha M(s)$  and that the total amount subtracted is then redistributed across all symbols in  $\Sigma$  proportionally according to the symbols’ probability under the base distribution  $G_0$ . Thus non-zero counts are usually reduced, with larger counts typically reduced by a larger amount. Doing this mitigates the overestimation of probabilities of rare symbols that happen to appear by chance. On the other hand, for symbols that did not appear at all, the estimates of their probabilities are pulled upward

**Figure 1. Illustration of the power-law scaling of word frequencies in English text.** The relative word frequency (estimated from a large corpus of written text) is plotted against each word’s rank when ordered according to frequency. One can see that there are a few very common words and a large number of relatively rare words; in fact, the 200 most common words account for over 50% of the observed text. The rank/frequency relationship is very close to a pure power law relationship which would be a perfectly straight line on this log–log plot. Also plotted are samples drawn from a PYP (in blue) and a Dirichlet distribution (in red) fitted to the data. The Pitman–Yor captures the power-law statistics of the English text much better than the Dirichlet.





from zero, mitigating underestimation of their probability. We describe this effect as “stealing from the rich and giving to the poor.” This is precisely how the PYP manifests a power-law characteristic. If one thinks of the  $M(s)$ ’s and  $\alpha$  as parameters then one could imagine ways to set them to best describe the data. Intuitively this is not at all far from what is done, except that the  $M(s)$ ’s and  $\alpha$  are themselves treated in a Bayesian way, i.e., we average over them under the posterior distribution in Equation 3.

#### 4.2. Context trees

We now return to making use of the contextual dependencies in  $\mathbf{x}$  and to estimating all of the conditional distributions  $G_{\mathbf{u}}$  relevant to predicting symbols following a general context  $\mathbf{u}$ . The assumption we make is that if two contexts are similar, then the corresponding conditional distributions over the symbols that follow those contexts will tend to be similar as well. A simple and natural way of defining similarity between contexts is that of overlapping contextual suffixes. This is easy to see in a concrete example from language modeling. Consider the distribution over words that would follow  $\mathbf{u} = \text{in the United States of}$ . The assumption we make is that this distribution will be similar to the distribution following the shorter context, the *United States of*, which we in turn expect to be similar to the distribution following *United States of*. These contexts all share the same length three suffix.

In this section and the following one, we will discuss how this assumption can be codified using a *hierarchical Bayesian model*.<sup>8, 11</sup> To start, we will only consider fixed, finite length contexts. When we do this we say that we are making an  $n$ th order Markov assumption. This means that each symbol only depends on the last  $n$  observed symbols. Note that this assumption dictates that distributions are not only similar but equal among contexts whose suffixes overlap in their last  $n$  symbols. This equality constraint is a strong assumption that we will relax in Section 5.

We can visualize the similarity assumption we make by constructing a *context tree*: Arrange the contexts  $\mathbf{u}$  (and the associated distributions  $G_{\mathbf{u}}$ ) in a tree where the parent of a node  $\mathbf{u}$ , denoted  $\sigma(\mathbf{u})$ , is given by its *longest proper suffix* (i.e.,  $\mathbf{u}$  with its first symbol from the left removed). Figure 2 gives an example of a context tree with  $n = 3$  and  $\Sigma = \{0, 1\}$ . Since for now we

are making an  $n$ th order Markov assumption, it is sufficient to consider only the contexts  $\mathbf{u} \in \Sigma_n^* = \{\mathbf{u}' \in \Sigma^* : |\mathbf{u}'| \leq n\}$  of length at most  $n$ . The resulting context tree has height  $n$  and the total number of nodes in the tree grows exponentially in  $n$ . The memory complexity of models built on such context trees usually grows too large and too quickly for reasonable values of  $n$  and  $|\Sigma|$ . This makes it nearly impossible to estimate *all* of the distributions  $G_{\mathbf{u}}$  in the naïve way described in Section 2. This estimation problem led us to hierarchical Bayesian modeling using Pitman–Yor processes.

#### 4.3. Hierarchical Pitman–Yor processes

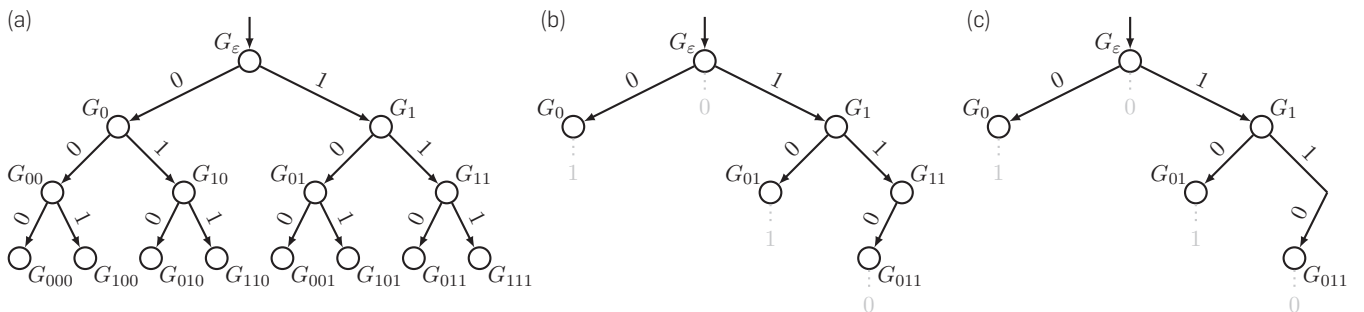
Having defined a context tree and shown that the Pitman–Yor prior over distributions exhibits power-law characteristics, it remains to integrate the two.

Recall that  $G \sim \mathcal{PY}(\alpha, G_0)$  means that  $G$  is a random distribution with a PYP prior parameterized by a discount parameter  $\alpha$  and a base distribution  $G_0$ . The expected value of  $G$  under repeated draws from the PYP is the base distribution  $G_0$ . Because of this fact we can use this process to encode any assumption that states that on average  $G$  should be similar to  $G_0$ . To be clear, this is just a prior assumption. As always, observing data may lead to a change in our belief. We can use this mechanism to formalize the context tree notion of similarity. In particular, to encode the belief that  $G_{\mathbf{u}}$  should be similar to  $G_{\sigma(\mathbf{u})}$ , we can use a PYP prior for  $G_{\mathbf{u}}$  with base distribution  $G_{\sigma(\mathbf{u})}$ . We can apply the same mechanism at each node of the context tree, leading to the following model specification:

$$\begin{aligned} G_{\varepsilon} &\sim \mathcal{PY}(\alpha_0, G_0) \\ G_{\mathbf{u}} | G_{\sigma(\mathbf{u})} &\sim \mathcal{PY}(\alpha_{|\mathbf{u}|}, G_{\sigma(\mathbf{u})}) \quad \text{for all } \mathbf{u} \in \Sigma_n^* \setminus \varepsilon \\ x_i | \mathbf{x}_{i-n:i-1} = \mathbf{u}, G_{\mathbf{u}} &\sim G_{\mathbf{u}} \quad \text{for } i = 1, \dots, T \end{aligned} \quad (4)$$

The second line says that a priori the conditional distribution  $G_{\mathbf{u}}$  should be similar to  $G_{\sigma(\mathbf{u})}$ , its parent in the context tree. The variation of  $G_{\mathbf{u}}$  around its mean  $G_{\sigma(\mathbf{u})}$  is described by a PYP with a context length-dependent discount parameter  $\alpha_{|\mathbf{u}|}$ . At the top of the tree the distribution  $G_{\varepsilon}$  for the empty context  $\varepsilon$  is similar to an overall base distribution  $G_0$ , which specifies our prior belief that each symbol  $s$  will appear with probability  $G_0(s)$ . The third line describes the

**Figure 2. (a) Full context tree containing all contexts up to length 3 over symbol set  $\Sigma = \{0, 1\}$ . (b) Context tree actually needed for the string 0110. Observations in the context in which they were observed are denoted in gray below the corresponding context. (c) Compact context tree for the same string, with non-branching chains marginalized out.**



$n$ th order Markov model for  $\mathbf{x}$ : It says that the distribution over each symbol  $x_i$  in  $\mathbf{x}$ , given that its context consisting of the previous  $n$  symbols  $x_{i-n:i-1}$  is  $\mathbf{u}$ , is simply  $G_{\mathbf{u}}$ .

The hierarchical Bayesian model in Equation 4 is called the *hierarchical Pitman-Yor process*.<sup>18</sup> It formally encodes our context tree similarity assumption about the conditional distributions using dependence among them induced by the hierarchy, with more similar distributions being more dependent. It is this dependence which allows the model to share information across the different contexts, and subsequently improve the estimation of all conditional distributions. It is worth noting that there is a well-known connection between the hierarchical PYP and a type of smoothing for  $m$ -gram language models called interpolated Kneser–Ney.<sup>10, 18</sup>

## 5. SEQUENCE MEMOIZER

The name *sequence memoizer* (SM) refers to both an extension of the hierarchical PYP model presented in the previous section, as well as to the set of techniques required to make practical use of the extended model. We first describe how the SM model extends the hierarchical PYP model and then discuss how to reduce the complexity of the model to make it computationally tractable. Finally, we sketch how inference is done in the SM.

### 5.1. The model

The SM model is a notationally subtle but important extension to the hierarchical PYP model (4) is described in the previous section. Instead of limiting the context lengths to  $n$ , the model is extended to include the set of distributions in all contexts of any (finite) length. This means that the distribution over each symbol is now conditioned on all previous symbols, not just the previous  $n$ .

Formally, the SM model is defined exactly as the hierarchical PYP model in Equation 4, but with two differences. First, the contexts range over all finite nonempty strings,  $\mathbf{u} \in \Sigma^+ \setminus \epsilon$ . Second, in the third line of Equation 4, instead of conditioning only on the previous  $n$  symbols, we condition on all previous symbols, so that  $x_i | \mathbf{x}_{1:i-1} = \mathbf{u}$ ,  $G_{\mathbf{u}} \sim G_{\mathbf{u}}$ . The assumptions embodied in the resulting model remain the same as that for the hierarchical PYP model: power-law scaling and similarity between related contexts.

The SM model can be interpreted as the limit of a hierarchical PYP model as the Markov order  $n$  tends to infinity. One's first impression of such a model might be that it would be impossible to handle, both statistically because of overfitting and other problems, and computationally because the model as described so far cannot even be represented in a computer with finite memory! Fortunately the Bayesian approach, where we compute the posterior distribution and marginalize over the parameters as in Equation 2 to obtain estimators of interest, prevents overfitting. Additionally, the techniques we develop in the next subsection make computation in this model practical.

### 5.2. Compacting the context tree

While lifting the finite restriction on context lengths

seems very desirable from a modeling perspective, the resulting SM model is a prior over an infinite number of parameters (conditional distributions)  $\{G_{\mathbf{u}}\}_{\mathbf{u} \in \Sigma^+}$ . In order to compute in this model, the number of conditional distributions that is accessed must be reduced to a finite number. The key to realizing that this is possible is that given a *finite length* sequence of symbols  $\mathbf{x}$ , we only need access to a finite number of conditional distributions. In particular, we only need  $G_{x_{1:i}}$  where  $i = 0, \dots, T$  and all the ancestors of each  $G_{x_{1:i}}$  in the context tree. The ancestors are needed because each  $G_{\mathbf{u}}$  has a prior that depends on its parent  $G_{\sigma(\mathbf{u})}$ . The resulting set of conditional distributions that the sequence  $\mathbf{x}$  actually depends on consists of  $G_{\mathbf{u}}$  where  $\mathbf{u}$  ranges over all contiguous substrings of  $\mathbf{x}$ , a finite set of  $\mathcal{O}(T^2)$  contexts. All other contexts in the tree can effectively be ignored. We denote this subtree of the context tree that  $\mathbf{x}$  actually depends on by  $\mathcal{T}(\mathbf{x})$ ; Figure 2b shows an example with  $\mathbf{x} = 0110$ .

Computation with such a quadratically sized context tree is possible but inefficient, especially if the sequence length is large. A second step reduces the tree down to a linear number of nodes. The key observation underlying this reduction is that many of the contexts that appear in  $\mathcal{T}(\mathbf{x})$  only appear in non-branching chains, i.e., each node on the chain only has one child in  $\mathcal{T}(\mathbf{x})$ . For example, in Figure 2b, the context 11 only occurs as a suffix of the longer context 011, and is part of the non-branching chain  $G_1 \xrightarrow{1} G_{11} \xrightarrow{0} G_{011}$ . In such a situation,  $G_{11}$  serves no purpose except to relate  $G_{011}$  with  $G_1$ . If we can directly express the prior of  $G_{011}$  in terms of  $G_1$ , then we can effectively ignore  $G_{11}$  and *marginalize* it out from the model.

Fortunately, a remarkable property related to an operation on Pitman–Yor processes called *coagulation* allows us to perform this marginalization exactly.<sup>14</sup> Specifically in the case of  $G_{11} | G_1 \sim \mathcal{PY}(\alpha_2, G_1)$  and  $G_{011} | G_{11} \sim \mathcal{PY}(\alpha_3, G_{11})$ , the property states simply that  $G_{011} | G_1 \sim \mathcal{PY}(\alpha_2 \alpha_3, G_1)$  where  $G_{11}$  has been marginalized out. In other words, the prior for  $G_{011}$  is another PYP whose discount parameter is simply the product of the discount parameters along the chain leading into it on the tree  $\mathcal{T}(\mathbf{x})$ , while the base distribution is simply the head of the chain  $G_1$ .

In general, applying the same marginalization procedure to all the non-branching chains of  $\mathcal{T}(\mathbf{x})$ , we obtain a *compact context tree*  $\hat{\mathcal{T}}(\mathbf{x})$  where all internal nodes have at least two children (all others have been integrated out). This is illustrated in Figure 2c, where each chain is replaced by an edge labeled by the sequence of symbols on the original edges of the chain (in the example only  $G_1 \xrightarrow{1} G_{11} \xrightarrow{0} G_{011}$  is replaced by  $G_1 \xrightarrow{01} G_{011}$ ). One can easily show that the number nodes in the compact context tree  $\hat{\mathcal{T}}(\mathbf{x})$  is at most twice the length of the sequence  $\mathbf{x}$  (independent of  $|\Sigma|$ ).

At this point some readers may notice that the compact context tree has a structure reminiscent of a data structure for efficient string operations called a suffix tree.<sup>9</sup> In fact, the structure of the compact context tree is given by the suffix tree for the *reverse* sequence  $x_T, x_{T-1}, \dots, x_1$ . Similar extensions from fixed-length to unbounded-length contexts, followed by reductions in the context trees, have also been developed

in the compression literature.<sup>4,19</sup>

### 5.3. Inference and prediction

As a consequence of the two marginalization steps described in the previous subsection, inference in the full SM model with an infinite number of parameters is equivalent to inference in the compact context tree  $\hat{T}(\mathbf{x})$  with a linear number of parameters. Further, the prior over the conditional distributions on  $\hat{T}(\mathbf{x})$  still retains the form of a hierarchical PYP: each node still has a PYP prior with its parent as the base distribution. This means that inference algorithms developed for the finite-order hierarchical PYP model can be easily adapted to the SM. We will briefly describe the inference algorithms we employ.

In the SM model we are mainly interested in the predictive distribution of the next symbol being some  $s \in \Sigma$  given some context  $\mathbf{u}$ , conditioned on an observed sequence  $\mathbf{x}$ . As in Equation 2, this predictive distribution is expressed as an expectation  $\mathbb{E}[G_{\mathbf{u}}(s)]$  over the posterior distribution of  $\{G_{\mathbf{u}'}\}_{\mathbf{u}' \in \hat{T}(\mathbf{x})}$ . Just as in Equation 3 as well, it is possible to express  $\mathbb{E}[G_{\mathbf{u}}(s)]$  as an expectation over a set of random counts  $\{N(\mathbf{u}'s'), M(\mathbf{u}'s')\}_{\mathbf{u}' \in \hat{T}(\mathbf{x}), s' \in \Sigma}$ :

$$\mathbb{E}[G_{\mathbf{u}}(s)] = \mathbb{E}\left[\frac{N(\mathbf{u}s) - \alpha_{\mathbf{u}}M(\mathbf{u}s) + \sum_{s' \in \Sigma} \alpha_{\mathbf{u}}M(\mathbf{u}s')G_{\sigma(\mathbf{u})}(s)}{\sum_{s' \in \Sigma} N(\mathbf{u}s')}\right] \quad (5)$$

Again, the first term in the numerator can be interpreted as a count of the number of times  $s$  occurs in the context  $\mathbf{u}$ , the second term is the reduction applied to the count, while the third term spreads the total reduction across  $\Sigma$  according to the base distribution  $G_{\sigma(\mathbf{u})}(s)$ . Each context  $\mathbf{u}$  now has its own discount parameter  $\alpha_{\mathbf{u}}$ , which is the product of discounts on the non-branching chain leading to  $\mathbf{u}$  on  $\mathcal{T}(\mathbf{x})$ , while the parent  $\sigma(\mathbf{u})$  is the head of the chain. Notice that Equation 5 is defined recursively, with the predictive distribution  $G_{\mathbf{u}}$  in context  $\mathbf{u}$  being a function of the same in the parent  $\sigma(\mathbf{u})$  and so on up the tree.

The astute reader might notice that the above does not quite work if the context  $\mathbf{u}$  does not occur in the compact context tree  $\hat{T}(\mathbf{x})$ . Fortunately the properties of the hierarchical PYP work out in our favor, and the predictive distribution is simply the one given by the longest suffix of  $\mathbf{u}$  that is in  $\mathcal{T}(\mathbf{x})$ . If this is still not in  $\hat{T}(\mathbf{x})$ , then a converse of the coagulation property (called *fragmentation*) allows us to reintroduce the node back into the tree.

To evaluate the expectation (5), we use stochastic (Monte Carlo) approximations where the expectation is approximated using *samples* from the posterior distribution. The samples are obtained using Gibbs sampling<sup>16</sup> as in Teh<sup>18</sup> and Wood,<sup>21</sup> which repeatedly makes local changes to the counts, and using sequential Monte Carlo<sup>5</sup> as in Gasthaus et al.,<sup>7</sup> which iterates through the sequence  $x_1, x_2, \dots, x_T$ , keeping track of a set of samples at each step, and updating the samples as each symbol  $x_i$  is incorporated into the model.

## 6. DEMONSTRATION

We now consider two target applications: language modeling and data compression. It is demonstrated that the SM model is able to achieve better performance than

most state-of-the-art techniques by capturing long-range dependencies.

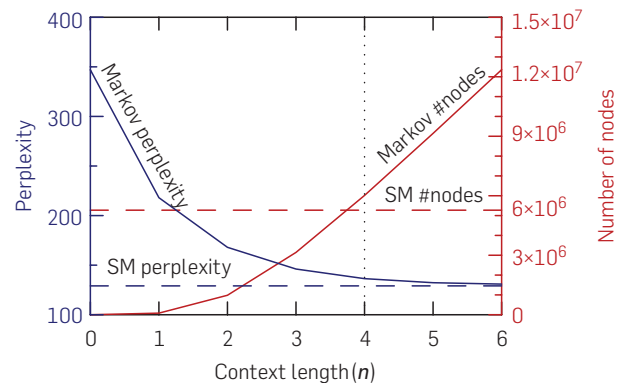
### 6.1. Language modeling

Language modeling is the task of fitting probabilistic models to sentences (sequences of words), which can then be used to judge the plausibility of new sequences being sentences in the language. For instance, “God save the Queen” should be given a higher score than “Queen the God save” and certainly more than “glad slave the spleen” under any model of English. Language models are mainly used as building blocks in natural language processing applications such as statistical machine translation and automatic speech recognition. In the former, for example, a translation algorithm might propose multiple sequences of English words, at least one of which is hoped to correspond to a good translation of a foreign language source. Usually only one or a few of these suggested sequences are plausible English language constructions. The role of the language model is to judge which English construction is best. Better language models generally lead to better translators.

Language model performance is reported in terms of a standard measure called *perplexity*. This is defined as  $2^{\ell(\mathbf{x})}$  where  $\ell(\mathbf{x}) = -\frac{1}{|\mathbf{x}|} \sum_{i=1}^{|\mathbf{x}|} \log_2 P(x_i | \mathbf{x}_{1:i-1})$  is the average *log-loss* on a sequence  $\mathbf{x}$  and the average number of bits per word required to encode the sequence using an optimal code. Another interpretation of perplexity is that it is the average number of guesses the model would have to make before it guessed each word correctly (if it makes these guesses by drawing samples from its estimate of the conditional distribution). For the SM model  $P(x_i | \mathbf{x}_{1:i-1})$  is computed as in Equation 5. Both lower log-loss and lower perplexity are better.

Figure 3 compares the SM model against  $n$ th order Markov

**Figure 3.** In blue is the performance of the SM model (dashed line) versus  $n$ th order Markov models with hierarchical PYP priors (solid line) as  $n$  varies (test data perplexity, lower is better). In red is the computational complexity of the SM model (dashed line) versus the Markov models (solid line) in terms of the number of nodes in the context tree/trie. For this four million word New York Times corpus, as  $n$  passes 4, the memory complexity of the Markov models grows larger than that of the SM, yet, the SM model yields modeling performance that is better than all Markov models regardless of their order. This suggests that for  $n \geq 4$  the SM model is to be preferred: it requires less space to store yet results in a comparable if not better model.



<sup>b</sup> Note that an  $n$ th order Markov model is an  $m$ -gram model where  $m = n + 1$ .



models with hierarchical PYP priors, for various values of  $n$ , on a four million word New York Times corpus<sup>b</sup>. Table 1 compares the hierarchical PYP Markov model and the SM model against other state-of-the-art models, on a 14 million word Associated Press news article corpus. The AP corpus is a benchmark corpus for which the performance of many models is available. It is important to note that it was processed by Bengio et al.<sup>2</sup> to remove low frequency words. Note that this is to the detriment of the SM, which is explicitly designed to improve modeling of low word frequencies due to power-law scaling. It is also a relatively small corpus, limiting the benefits of the SM model at capturing longer range dependencies.

Our results show that the SM model is a competitive language model. However, perplexity results alone do not tell the whole story. As more data is used to estimate a language model, typically its performance improves. This means that computational considerations such as memory and runtime must enter into the discussion about what constitutes a good language model. In many applications, fast prediction is imperative, in others, particularly in online settings, incorporation of new data into the model must be fast. In comparison to more complex language models, prediction in the SM has real-world time complexity that is essentially the same as that of a smoothed finite-order Markov model, while its memory complexity is linear in the amount of data. The computational complexity of Markov models theoretically does not depend on the amount of data but is exponential in the Markov order, rendering straightforward extensions to higher orders impractical. The SM model directly fixes this problem while remaining computationally tractable. Constant space, constant time extensions to the SM model<sup>1</sup> have been developed, which show great promise for language modeling and other applications.

## 6.2. Compression

Shannon's celebrated results in information theory<sup>17</sup> have led to lossless compression technology that, given a coding distribution, nearly optimally achieves the theoretical lower limit (given by the log-loss) on the number of bits needed to encode a sequence. Lossless compression is closely related to sequence modeling: an incrementally constructed probabilistic sequence model such as the SM can be used to adaptively construct coding distributions which can then be

**Table 1. Language modeling performance for a number of models on an Associated Press news corpus (lower perplexity is better). Interpolated and modified Kneser–Ney are state-of-the-art language models. Along with hierarchical PYP and the sequence memoizer, these models do not model relationships among words in the vocabulary. Provided for comparison are the results for the models of Bengio et al. and Mnih et al. which belong to a different class of models that learn word representations from data**

Source	Perplexity
Bengio et al. <sup>2</sup>	109.0
Mnih et al. <sup>13</sup>	83.9
4-gram Interpolated Kneser–Ney <sup>3,18</sup>	106.1
4-gram Modified Kneser–Ney <sup>3,18</sup>	102.4
4-gram Hierarchical PYP <sup>18</sup>	101.9
Sequence Memoizer <sup>21</sup>	96.9

**Table 2. Compression performance in terms of weighted average log-loss (average bits per byte under optimal entropy encoding, lower is better) for the Calgary corpus, a standard benchmark collection of diverse filetypes. The results for unboundedlength context PPM is from Cleary and Teahan.<sup>4</sup> The results for CTW is from Willems.<sup>20</sup> The bzip2 and gzip results come from running the corresponding standard unix command line tools with no extra arguments**

Model	SM	PPM	CTW	bzip2	gzip
Average bits/byte	<b>1.89</b>	1.93	1.99	2.11	2.61

directly used for compression based on entropy coding.

We demonstrate the theoretical performance of a lossless compressor based on the SM model on a number of standard compression corpora. Table 2 summarizes a comparison of our lossless compressor against other state-of-the-art compressors on the Calgary corpus, a well-known compression benchmark consisting of 14 files of different types and varying lengths.

In addition to the experiments on the Calgary corpus, SM compression performance was also evaluated on a 100MB excerpt of the English version of Wikipedia (XML text dump).<sup>12</sup> On this excerpt, the SM model achieved a log-loss of 1.66 bits/symbol amounting to a compressed file size of 20.80MB. While this is worse than 16.23MB achieved by the best demonstrated Wikipedia compressor, it demonstrates that the SM model can scale to sequences of this length. We have also explored the performance of the SM model when using a larger symbol set ( $\Sigma$ ). In particular, we used the SM model to compress UTF-16 encoded Chinese text using a 16-bit alphabet. On a representative text file, the Chinese Union version of the bible, we achieved a log-loss of 4.91 bits per Chinese character, which is significantly better than the best results in the literature (5.44 bits).<sup>22</sup>

## 7. CONCLUSION

The SM achieves improved compression and language modeling performance. These application-specific performance improvements are arguably worthwhile scientific achievements by themselves. Both have the potential to be tremendously useful, and may yield practical consequences of societal and commercial value.

We encourage the reader, however, not to mentally categorize the SM as a compressor or language model. Nature is replete with discrete sequence data that exhibits long-range dependencies and power-law characteristics. The need to model the processes that generate such data is likely to grow in prevalence. The SM is a general purpose model for discrete sequence data that remains computationally tractable despite its power and despite the fact that it makes only very general assumptions about the data generating process.

Our aim in communicating the SM is also to encourage readers to explore the fields of probabilistic and Bayesian modeling in greater detail. Expanding computational capacity along with significant increases in the amount and variety of data to be analyzed across many scientific and engineering disciplines is rapidly enlarging the class of probabilistic models that one can imagine and employ. Over the coming decades hierarchical Bayesian models are

likely to become increasingly prevalent in data analysis oriented fields like applied statistics, machine learning, and computer science. It is our belief that the SM and its ilk will come to be seen as relatively simple building blocks for the enormous and powerful hierarchical models of tomorrow.

Source code and example usages of the SM are available at <http://www.sequencemoizer.com/>. A lossless compressor built using the SM can be explored at <http://www.deplump.com/>.

## Acknowledgments

We wish to thank the Gatsby Charitable Foundation and Columbia University for funding. 

## References

- Bartlett, N., Pfau, D., Wood, F. Forgetting counts: Constant memory inference for a dependent hierarchical Pitman–Yor process. In *27th International Conference on Machine Learning*, to appear (2010).
- Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* 3 (2003), 1137–1155.
- Chen, S.F., Goodman, J. An empirical study of smoothing techniques for language modeling. *Comput. Speech Lang.* 13, 4 (1999), 359–394.
- Cleary, J.G., Teahan, W.J. Unbounded length contexts for PPM. *Comput. J.* 40 (1997), 67–75.
- Doucet, A., de Freitas, N., Gordon, N.J. *Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science.* Springer-Verlag, New York, May 2001.
- Gasthaus, J., Teh, Y.W. Improvements to the sequence memoizer. In *Advances in Neural Information Processing Systems 23*, to appear (2010).
- Gasthaus, J., Wood, F., Teh, Y.W. Lossless compression based on the sequence memoizer. *Data Compression Conference 2010*. J.A. Storer, M.W. Marcellin, eds. Los Alamitos, CA, USA, 2010, 337–345. IEEE Computer Society.
- Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B. *Bayesian data analysis*. Chapman & Hall, CRC, 2nd edn, 2004.
- Giegerich, R., Kurtz, S. From Ukkonen to McCreight and Weiner: A unifying view of linear-time suffix tree construction. *Algorithmica* 19, 3 (1997), 331–353.
- Goldwater, S., Griffiths, T.L., Johnson, M. Interpolating between types and tokens by estimating power law generators. In *Advances in Neural Information Processing Systems 18* (2006), MIT Press, 459–466.
- MacKay, D.J.C., Peto, L.B. A hierarchical Dirichlet language model. *Nat. Lang. Eng.* 1, 2 (1995), 289–307.
- Mahoney, M. Large text compression benchmark. URL: <http://www.matmahoney.net/text/text.html> (2009).
- Mnih, A., Yuecheng, Z., Hinton, G. Improving a statistical language model through non-linear prediction. *Neurocomputing* 72, 7–9 (2009), 1414–1418.
- Pitman, J. Coalescents with multiple collisions. *Ann. Probab.* 27 (1999), 1870–1902.
- Pitman, J., Yor, M. The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *Ann. Probab.* 25 (1997), 855–900.
- Robert, C.P., Casella, G. *Monte Carlo Statistical Methods*. Springer Verlag, 2004.
- Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* (reprinted in *ACM SIGMOBILE Mobile Computing and Communications Review* 2001) (1948).
- Teh, Y.W. A hierarchical Bayesian language model based on Pitman–Yor processes. In *Proceedings of the Association for Computational Linguistics* (2006), 985–992.
- Willems, F.M.J. The context-tree weighting method: Extensions. *IEEE Trans. Inform. Theory* 44, 2 (1998), 792–798.
- Willems, F.M.J. CTW website. URL: <http://www.ele.tue.nl/ctw/> (2009).
- Wood, F., Archambeau, C., Gasthaus, J., James, L., Teh, Y.W. A stochastic memoizer for sequence data. In *26th International Conference on Machine Learning* (2009), 1129–1136.
- Wu, P., Teahan, W.J. A new PPM variant for Chinese text compression. *Nat. Lang. Eng.* 14, 3 (2007), 417–430.
- Zipf, G. *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press, Cambridge, MA, 1932.

**Frank Wood** (fwood@stat.columbia.edu), Department of Statistics, Columbia University, New York.

**Jan Gasthaus** (j.gasthaus@gatsby.ucl.ac.uk), Gatsby Computational Neuroscience Unit, University College London, England.

**Cédric Archambeau** (cedric.archambeau@xerox.com), Xerox Research Centre Europe, Grenoble, France.

**Lancelot James** (lancelot@ust.hk), Department of Information, Systems, Business, Statistics and Operations Management, Hong Kong University of Science and Technology, Kowloon, Hong Kong.

**Yee Whye Teh** (ywteh@gatsby.ucl.ac.uk), Gatsby Computational Neuroscience Unit, University College London, England.

© 2011 ACM 0001-0782/11/0200 \$10.00



Association for  
Computing Machinery

Advancing Computing as a Science & Profession



MentorNet

You've come a long way.  
Share what you've learned.



ACM has partnered with MentorNet, the award-winning nonprofit e-mentoring network in engineering, science and mathematics. MentorNet's award-winning **One-on-One Mentoring Programs** pair ACM student members with mentors from industry, government, higher education, and other sectors.

- Communicate by email about career goals, course work, and many other topics.
- Spend just **20 minutes a week** - and make a huge difference in a student's life.
- Take part in a lively online community of professionals and students all over the world.



Make a difference to a student in your field.  
Sign up today at: [www.mentornet.net](http://www.mentornet.net)  
Find out more at: [www.acm.org/mentornet](http://www.acm.org/mentornet)

MentorNet's sponsors include 3M Foundation, ACM, Alcoa Foundation, Agilent Technologies, Amylin Pharmaceuticals, Bechtel Group Foundation, Cisco Systems, Hewlett-Packard Company, IBM Corporation, Intel Foundation, Lockheed Martin Space Systems, National Science Foundation, Naval Research Laboratory, NVIDIA, Sandia National Laboratories, Schlumberger, S.D. Bechtel, Jr. Foundation, Texas Instruments, and The Henry Luce Foundation.

# Technical Perspective

## DRAM Errors in the Wild

By Norman P. Jouppi

IN AN ERA of mobile devices used as windows into services provided by computing in the cloud, the cost and reliability of services provided by large warehouse-scale computers<sup>1</sup> is paramount. These warehouse-scale computers are implemented with racks of servers, each one typically consisting of one or two processor chips but many memory chips. Even with a crash-tolerant application layer, understanding the sources and types of errors in server memory systems is still very important.

Similarly, as we look forward to exascale performance in more traditional supercomputing applications, even memory errors correctable through traditional error-correcting codes can have an outsized impact on the total system performance.<sup>3</sup> This is because in many systems, execution on a node with hardware-corrected errors that are logged in software runs significantly slower than on nodes without errors. Since execution of bulk synchronous parallel applications is only as fast as the slowest local computation, in a million-node computation the slowdown of one node from memory errors can end up delaying the entire million-node system.

At the system level, low-end PCs have historically not provided any error detection or correction capability, while servers have used error-correcting codes (ECC) that have enabled correction of a single error per codeword. This worked especially well when a different memory chip was used for each bit read or written by a memory bus (such as when using “x1” memory chips). However, in the last 15 years as memory busses have become wider, more bits on the bus need to be read or written from each memory chip, leading to the use of memory chips that can provide four (“x4”) or more bits at a time to a memory bus. Unfortunately, this increases the probability of errors correlated across multiple bits, such as when part of a chip address circuit fails. In

**I hope the following paper will motivate the collection and publication of even more large-scale system memory reliability data.**


order to handle cases where an entire chip’s contribution to a memory bus is corrupted, chip-kill correct error correcting codes have been developed.<sup>2</sup>

Since the introduction of DRAMs in the mid-1970s, there has been much work on improving the reliability of individual DRAM devices. Some of the classic problems addressed were tolerance of radiation, from either impurities in the package or cosmic sources. In contrast, very little information has been published on reliability of memory at the system level. There are several reasons for this. First, much of the industrial data is specific to particular memory or CPU vendors. This industrial data typically focuses on configurations that are particularly problematic. Therefore neither DRAM, CPU, nor system vendors find it in their best interest to publish this data.

Nevertheless, in order to advance the field, knowledge of the types of memory errors, their frequencies, and conditions that exacerbate or are unrelated to higher error rates are of critical importance. In order to fill this gap, Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber analyzed measurements of memory errors in a large fleet of commodity

servers over a period of 2.5 years. They collected data on multiple DRAM capacities, technologies, and vendors (suitably anonymized), totaling millions of DIMM days.

They found that DRAM error behavior at the system level differs markedly from widely held assumptions. For example, they found DRAM error rates that are orders of magnitude more common than previously reported. Additionally, they found that temperature has a surprisingly low effect on memory errors. However, even though errors are rare, they are highly correlated in time by DIMM. Under-scoring the importance of chip-kill error correction in servers, they found that the use of chip-kill error correction can reduce the rate of uncorrectable errors by a factor of 3–8.

I hope the following paper will motivate the collection and publication of even more large-scale system memory reliability data. This work and future studies will be instrumental in aiding architects and system designers to address and solve the real problems in memory system reliability, enabling both cost-effective and reliable cloud services as well as efficiently extending supercomputing to the exascale. 

### References

1. Barroso, L. and Hölzle, U. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis Lectures on Computer Science*. Morgan Claypool, 2009.
2. Dell, T.J. A white paper on the benefits of chipkill-correct ECC for PC server main memory. *IBM Microelectronics*, 1997.
3. Yelick, K. Ten Ways to Waste a Parallel Computer; <http://isca09.cs.columbia.edu/ISCA09-WasteParallelComputer.pdf>

**Norman P. Jouppi** (Norm.Jouppi@hp.com) is a Senior Fellow and Director of Hewlett-Packard’s Intelligent Infrastructure Lab in Palo Alto, CA.



# DRAM Errors in the Wild: A Large-Scale Field Study

By Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber

## Abstract

Errors in dynamic random access memory (DRAM) are a common form of hardware failure in modern compute clusters. Failures are costly both in terms of hardware replacement costs and service disruption. While a large body of work exists on DRAM in laboratory conditions, little has been reported on real DRAM failures in large production clusters. In this paper, we analyze measurements of memory errors in a large fleet of commodity servers over a period of 2.5 years. The collected data covers multiple vendors, DRAM capacities and technologies, and comprises many millions of dual in-line memory module (DIMM) days.

The goal of this paper is to answer questions such as the following: How common are memory errors in practice? What are their statistical properties? How are they affected by external factors, such as temperature and utilization, and by chip-specific factors, such as chip density, memory technology, and DIMM age?

We find that DRAM error behavior in the field differs in many key aspects from commonly held assumptions. For example, we observe DRAM error rates that are orders of magnitude higher than previously reported, with 25,000–70,000 errors per billion device hours per Mb and more than 8% of DIMMs affected by errors per year. We provide strong evidence that memory errors are dominated by hard errors, rather than soft errors, which previous work suspects to be the dominant error mode. We find that temperature, known to strongly impact DIMM error rates in lab conditions, has a surprisingly small effect on error behavior in the field, when taking all other factors into account. Finally, unlike commonly feared, we do not observe any indication that newer generations of DIMMs have worse error behavior.

## 1. INTRODUCTION

Errors in dynamic random access memory (DRAM) devices have been a concern for a long time.<sup>3,11,15–17,22</sup> A memory error is an event that leads to the logical state of one or multiple bits being read differently from how they were last written. Memory errors can be caused by electrical or magnetic interference (e.g., due to cosmic rays), can be due to problems with the hardware (e.g., a bit being permanently damaged), or can be the result of corruption along the data path between the memories and the processing elements. Memory errors can be classified into soft errors, which randomly corrupt bits but do not leave physical damage; and hard errors, which corrupt bits in a repeatable manner because of a physical defect.

The consequence of a memory error is system-dependent. In systems using memory without support for error correction and detection, a memory error can lead to a machine crash or

applications using corrupted data. Most memory systems in server machines employ error correcting codes (ECC),<sup>6</sup> which allow the detection and correction of one or multiple bit errors. If an error is uncorrectable, i.e., the number of affected bits exceed the limit of what the ECC can correct, typically a machine shutdown is forced. In many production environments, including ours, a single uncorrectable error (UE) is considered serious enough to replace the dual in-line memory module (DIMM) that caused it.

Memory errors are costly in terms of the system failures they cause and the repair costs associated with them. In production sites running large-scale systems, memory component replacements rank near the top of component replacements<sup>19</sup> and memory errors are one of the most common hardware problems to lead to machine crashes.<sup>18</sup> There is also a fear that advancing densities in DRAM technology might lead to increased memory errors, exacerbating this problem in the future.<sup>3,12,13</sup>

Despite the practical relevance of DRAM errors, very little is known about their prevalence in real production systems. Existing studies; for example, see Baumann, Borucki et al., Johnston, May and Woods, Normand, and Ziegler and Lanford,<sup>3,4,9,11,16,22</sup> are mostly based on lab experiments using accelerated testing, where DRAM is exposed to extreme conditions (such as high temperature) to artificially induce errors. It is not clear how such results carry over to real production systems. The few prior studies that are based on measurements in real systems are small in scale, such as recent work by Li et al.,<sup>10</sup> who report on DRAM errors in 300 machines over a period of 3–7 months. Moreover, existing work is not always conclusive in their results. Li et al. cite error rates in the 200–5000 FIT per Mb range from previous lab studies, and themselves found error rates of <1 FIT per Mb.

This paper provides the first large-scale study of DRAM memory errors in the field. It is based on data collected from Google's server fleet over a period of more than 2 years making up many millions of DIMM days. The DRAM in our study covers multiple vendors, DRAM densities and technologies (DDR1, DDR2, and FBDIMM).

The goal of this paper is to answer the following questions: How common are memory errors in practice? How are they affected by external factors, such as temperature, and system utilization? How do they vary with chip-specific factors, such as chip density, memory technology, and DIMM age? What are their statistical properties?

The original version of this paper was published in *Proceedings of ACM SIGMETRICS*, June 2009.

## 2. BACKGROUND AND DATA

Our data covers the majority of machines in Google's fleet and spans nearly 2.5 years, from January 2006 to June 2008. Each machine comprises a motherboard with some processors and memory DIMMs. We study six different hardware *platforms*, where a platform is defined by the motherboard and memory generation. We refer to these platforms as platforms A to F throughout the paper.

The memory in these systems covers a wide variety of the most commonly used types of DRAM. The DIMMs come from multiple manufacturers and models, with three different capacities (1GB, 2GB, 4GB), and cover the three most common DRAM technologies: Double Data Rate 1 (DDR1), Double Data Rate 2 (DDR2), and Fully-Buffered (FBDIMM).

Most memory systems in use in servers today are protected by error detection and correction codes. Typical error codes today fall in the single error correct double error detect (SECCDED) category. That means they can reliably detect and correct any single-bit error, but they can only detect and not correct multiple bit errors. More powerful codes can correct and detect more error bits in a single memory word. For example, a code family known as chip-kill<sup>7</sup> can correct up to four adjacent bits at once, thus being able to work around a completely broken 4-bit wide DRAM chip. In our systems, Platforms C, D, and F use SECCDED, while Platforms A, B, and E rely on error protection based on chipkill. We use the terms correctable error (CE) and uncorrectable error (UE) in this paper to generalize away the details of the actual error codes used. Our study relies on data collected by low-level daemons running on all our machines that directly access hardware counters on the machine to obtain counts of correctable and uncorrectable DRAM errors.

If done well, the handling of correctable memory errors is largely invisible to application software. In contrast, UEs typically lead to a catastrophic failure. Either there is an explicit response action (such as a machine reboot), or there is risk of a data-corruption-induced failure, such as a kernel panic. In the systems we study, all UEs are considered serious enough to shut down the machine and replace the DIMM at fault.

Memory errors can be soft errors, which randomly corrupt bits, but do not leave any physical damage; or hard errors, which corrupt bits in a repeatable manner because of a physical defect (e.g., stuck bits). Our measurement

infrastructure captures both hard and soft errors, but does not allow us to reliably distinguish these types of errors. All our numbers include both hard and soft errors.

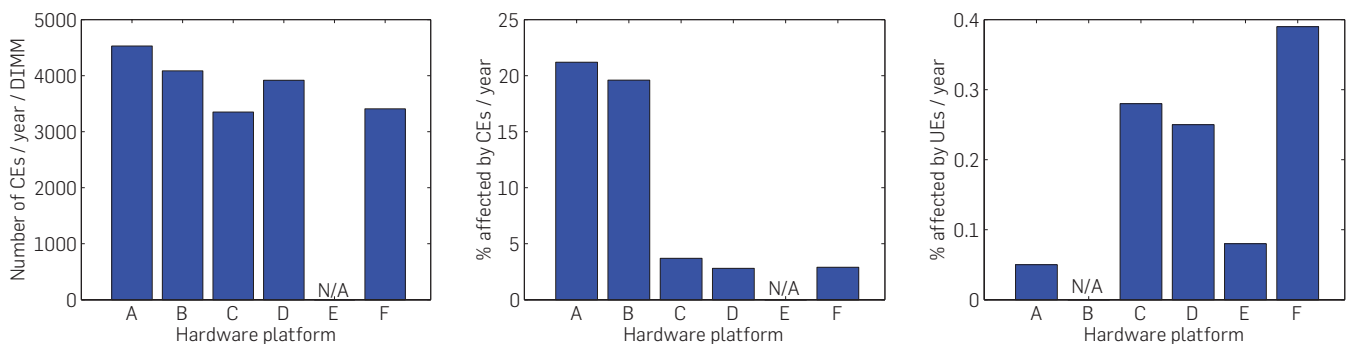
In order to avoid the accumulation of single-bit errors in a memory array over time, memory systems can employ a hardware scrubber<sup>14</sup> that scans through the memory, while the memory is otherwise idle. Any memory words with single-bit errors are written back after correction, thus eliminating the single-bit error if it was soft. Three of our hardware platforms (Platforms C, D, and F) make use of memory scrubbers. The typical scrubbing rate in those systems is 1GB every 45 min. In the other platforms (Platforms A, B, and E) errors are only detected on access.

## 3. HOW COMMON ARE ERRORS?

The analysis of our data shows that CEs are not rare events: We find that about a third of all machines in Google's fleet, and over 8% of individual DIMMs saw at least one CE per year. Figure 1 (left) shows the average number of CEs across all DIMMs in our study per year of operation broken down by hardware platform. Figure 1 (middle) shows the fraction of DIMMs per year that experience at least one CE. Consistently across all platforms, errors occur at a significant rate, with a fleet-wide average of nearly 4,000 errors per DIMM per year. The fraction of DIMMs that experience CEs varies from around 3% (for Platforms C, D and F) to around 20% (for Platforms A and B). Our per-DIMM rates of CEs translate to an average of 25,000–75,000 FIT (failures in time per billion hours of operation) per Mb and a median FIT range of 778–25,000 per Mb (median for DIMMs with errors). We note that this rate is significantly higher than the 200–5,000 FIT per Mb reported in previous studies and will discuss later in the paper reasons for the differences in results.

We also analyzed the rate of UEs and found that across the entire fleet 1.3% of machines are affected by UEs per year, with some platforms seeing as many as 2%–4% of machines affected. Figure 1 (right) shows the fractions of DIMMs that see the UEs in a given year, broken down by hardware platform. We note that, while the rate of CEs was comparable across platforms (recall Figure 1 (left)), the incidence of UEs is much more variable, ranging from 0.05% to 0.4%. In particular, Platforms C and D have a 3–6 times higher probability of seeing a UE than Platforms A and E.

**Figure 1. Frequency of errors: The average number of correctable errors (CEs) per year per DIMM (left), the fraction of DIMMs that see at least one CE in a given year (middle) and the fraction of DIMMs that see at least one uncorrectable error (UE) in a given year (right). Platforms C, D, and F use SECCDED, while platforms A, B, and E rely on error protection based on chipkill.**



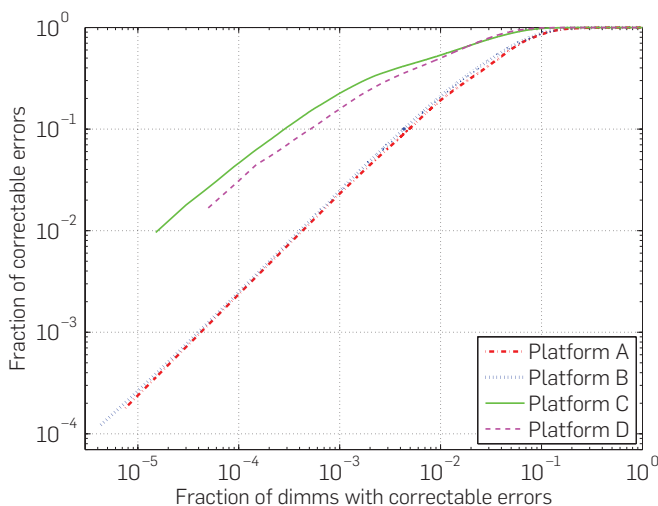
The differences in the rates of UEs between different platforms bring up the question of what factors impact the frequency of UEs. We investigated a number of factors that might explain the difference in memory rates across platforms, including temperature, utilization, DIMM age, capacity, DIMM manufacturer or memory technology (detailed tables included in the full paper<sup>20</sup>). While some of these affect the frequency of errors, they are not sufficient to explain the differences we observe between platforms.

While we cannot be certain about the cause of the differences between platforms, we hypothesize that the differences in UEs are due to differences in the error correction codes in use. In particular, Platforms C, D, and F are the only platforms that do not use a form of chip-kill.<sup>7</sup> Chip-kill is a more powerful code that can correct certain types of multiple bit errors, while the codes in Platforms C, D, and F can only correct single-bit errors.

While the above discussion focused on descriptive statistics, we also studied the statistical distribution of errors in detail. We observe that for all platforms the distribution of the number of CEs per DIMM per year is highly variable. For example, when looking only at those DIMMs that had at least one CE, there is a large difference between the mean and the median number of errors: the mean ranges from 20,000 to 140,000, while the median numbers are between 42 and 167.

When plotting the distribution of CEs over DIMMs (see Figure 2), we find that for all platforms the top 20% of DIMMs with errors make up over 94% of all observed errors. The shape of the distribution curve provides evidence that it follows a power-law distribution. Intuitively, the skew in the distribution means that a DIMM that has seen a large number of errors is likely to see more errors in the future. This is an interesting observation as this is not a property one would expect for soft errors (which should follow a random pattern) and might point to hard (or intermittent) errors as a major source of errors. This observation motivates us to take a closer look at correlations in Section 5.

**Figure 2. The distribution of correctable errors over DIMMs: The graph plots the fraction  $Y$  of all errors that is made up by the fraction  $X$  of DIMMs with the largest number of errors.**



## 4. IMPACT OF EXTERNAL FACTORS

In this section, we study the effect of various factors, including DIMM capacity, temperature, utilization, and age. We consider all platforms, except for Platform F, for which we do not have enough data to allow for a fine-grained analysis, and Platform E, for which we do not have data on CEs.

### 4.1. Temperature

Temperature is considered to (negatively) affect the reliability of many hardware components due to the strong physical changes on materials that it causes. In the case of memory chips, high temperature is expected to increase leakage current,<sup>2,8</sup> which in turn leads to a higher likelihood of flipped bits in the memory array. In the context of large-scale production systems, understanding the exact impact of temperature on system reliability is important, since cooling is a major cost factor. There is a trade-off to be made between increased cooling costs and increased downtime and maintenance costs due to higher failure rates.

To investigate the effect of temperature on memory errors, we plot in Figure 3 (left) the monthly rate of CEs as a function of temperature, as measured by a temperature sensor on the motherboard of each machine. Since temperature information is considered confidential, we report *relative* temperature values, where a temperature of  $x$  on the X-axis means the temperature was  $x^\circ\text{C}$  higher than the lowest temperature observed for a given platform. For better readability of the graphs, we *normalize* CE error rates for each platform by the platform's average CE rate, i.e., a value of  $y$  on the Y-axis refers to a CE rate that was  $y$  times higher than the average CE rate.

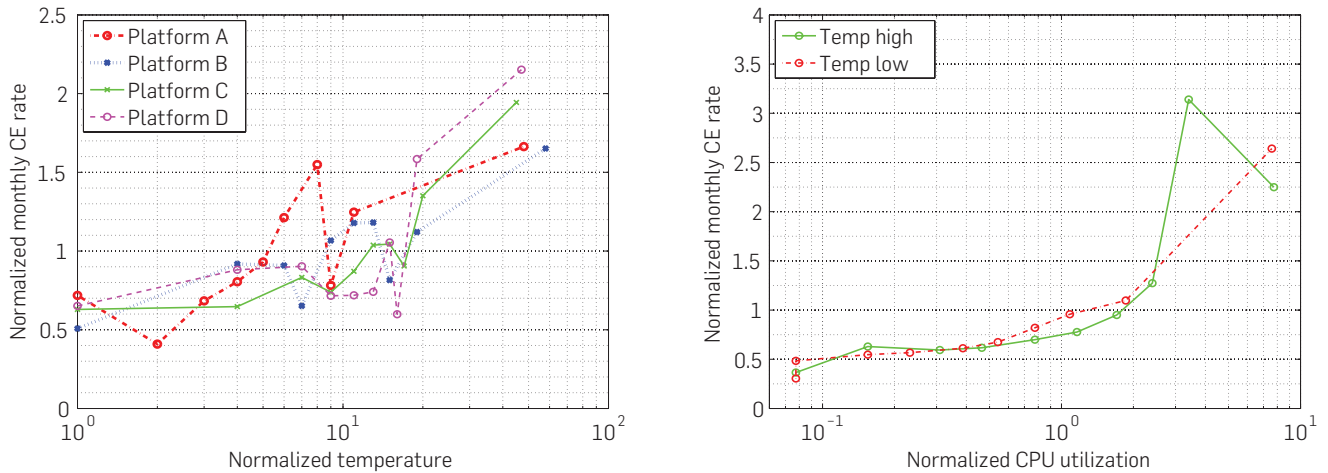
Figure 3 (left) shows that for all platforms higher temperatures are correlated with higher CE rates. For all platforms, the CE rate increases by at least a factor of 2 for an increase of temperature by  $20^\circ\text{C}$ ; for some it nearly triples.

It is not clear whether this correlation indicates a causal relationship, i.e., higher temperatures inducing higher error rates. Higher temperatures might just be a proxy for higher system utilization, i.e., the utilization increases leading independently to higher error rates and higher temperatures. In Figure 3 (right), we therefore isolate the effects of temperature from the effects of utilization. We divide the utilization measurements (CPU utilization) into deciles and report for each decile the observed error rate when temperature was “high” (above median temperature) or “low” (below median temperature). We observe that when controlling for utilization, the effects of temperature vanish. We also repeated these experiments with higher differences in temperature, e.g., by comparing the effect of temperatures above the 9th decile to temperatures below the 1st decile. In all cases, for the same utilization levels the error rates for high versus low temperature are very similar.

The results presented above were achieved by correlating the number of errors observed in a given month with the average temperature in that month. In our analysis, we also experimented with different measures of temperature, including temperatures averaged over different time scales (ranging from 1 h, to 1 day, to 1 month, to a dimm's lifetime), variability in temperature, and number of temperature excursions (i.e., number of times the temperature went above some



**Figure 3. The effect of temperature:** The left graph shows the normalized monthly rate of experiencing a correctable error (CE) as a function of the monthly average temperature, in deciles. The right graph shows the monthly rate of experiencing a CE as a function of CPU utilization, depending on whether the temperature was high (above median temperature) or low (below median temperature). We observe that when isolating the effects of temperature by controlling for utilization, it has much less of an effect.



threshold). We could not find significant levels of correlations between errors and any of the above measures for temperature when controlling for utilization.

#### 4.2. Utilization

The observations in Section 4.1 point to system utilization as a major contributing factor in the observed memory error rates. Ideally, we would like to study specifically the impact of memory utilization (i.e., number of memory accesses). Unfortunately, obtaining data on memory utilization requires the use of hardware counters, which our measurement infrastructure does not collect. Instead, we study two signals that we believe provide indirect indication of memory activity: CPU utilization and memory allocated. CPU utilization is the load activity on the CPU(s) measured instantaneously as a percentage of total CPU cycles used out of the total CPU cycles available and are averaged per machine for each month. For lack of space, we include here only results for CPU utilization. Results for memory allocated are similar and provided in the full paper.<sup>20</sup>

Figure 4 (left) shows the normalized monthly rate of CEs as a function of CPU utilization. We observe clear trends of increasing CE rates with increasing CPU utilization. Averaging across all platforms, the CE rates grow roughly logarithmically as a function of utilization levels (based on the roughly linear increase of error rates in the graphs, which have log scales on the X-axis).

One might ask whether utilization is just a proxy for temperature, where higher utilization leads to higher system temperatures, which then cause higher error rates. In Figure 4 (right), we therefore isolate the effects of utilization from those of temperature. We divide the observed temperature values into deciles and report for each range the observed error rates when utilization was “high” or “low.” High utilization means the utilization (CPU utilization and allocated memory, respectively) is above median, and low means the utilization was below median. We observe that even when keeping temperature fixed and focusing on one particular

temperature decile, there is still a huge difference in the error rates, depending on the utilization. For all temperature levels, the CE rates are by a factor of 2–3 higher for high utilization compared to low utilization.

One might argue that the higher error rate for higher utilization levels might simply be due to a higher detection rate of errors: In systems, where errors are only detected on application access, higher utilization increases the chance that an error will be detected and recorded (when an application accesses the affected cell). However, we also observe a correlation between utilization and error rates for Platforms C and D, which employ a memory scrubber. For these systems, any error will eventually be detected and recorded, if not by an application access then by the scrubber (unless it is overwritten before it is being read).

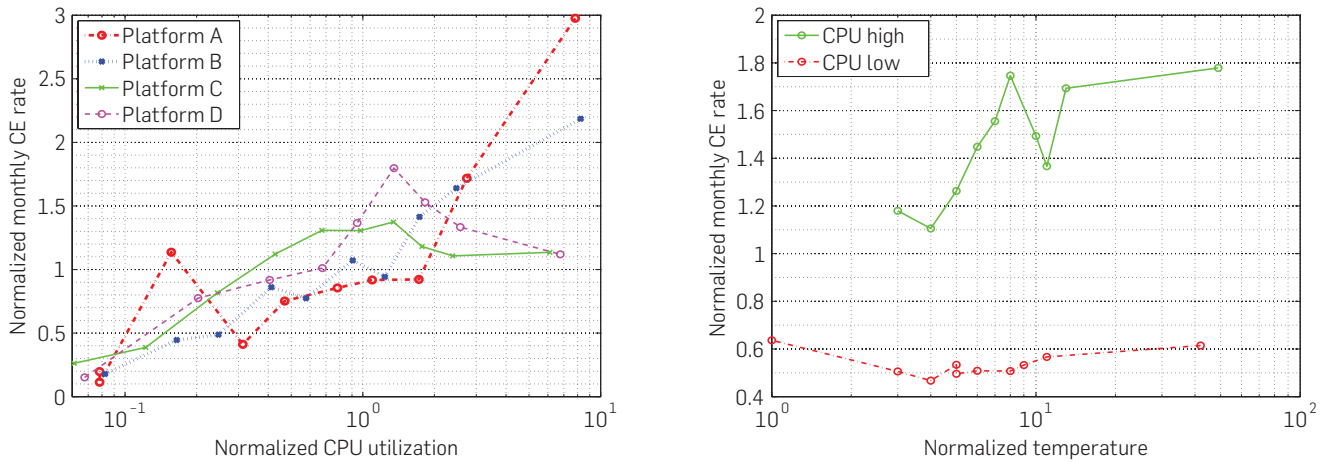
Our hypothesis is that the correlation between error rates and utilization is due to *hard* errors, such as a defective memory cell. In these cases, even if an error is detected and the system tries to correct it by writing back the correct value, the next time the memory cell is accessed it might again trigger a memory error. In systems with high utilization, chances are higher that the part of the hardware that is defective will be exercised frequently, leading to increased error rates. We will provide more evidence for our hard error hypothesis in Section 5.

#### 4.3. Aging

Age is one of the most important factors in analyzing the reliability of hardware components, since increased error rates due to early aging/wear-out limit the lifetime of a device. As such, we look at changes in error behavior over time for our DRAM population, breaking it down by age, platform, technology, correctable, and UEs.

Figure 5 shows normalized CE rates as a function of age for all platforms that have been in production for long enough to study age-related affects. We find that age clearly affects the CE rates for all platforms, and we observe similar trends also if we break the data further down by platform, manufacturer,

**Figure 4. The effect of utilization: The normalized monthly CE rate as a function of CPU utilization (left), and while controlling for temperature (right).**



and capacity (graphs included in full paper<sup>20</sup>).

For a more fine-grained view of the effects of aging and to identify trends, we study the mean cumulative function (MCF) of errors. While our full paper<sup>20</sup> includes several MCF plots, for lack of space we only summarize the results here. In short, we find that age severely affects CE rates: We observe an increasing incidence of errors as DIMMs get older, but only up to a certain point, when the incidence becomes almost constant (few DIMMs start to have CEs at very old ages). The age when errors first start to increase and the steepness of the increase vary per platform, manufacturer, and DRAM technology, but is generally in the 10–18 month range. We also note the lack of infant mortality for almost all populations. We attribute this to the weeding out of bad DIMMs that happens during the burn-in of DIMMs prior to putting them into production.

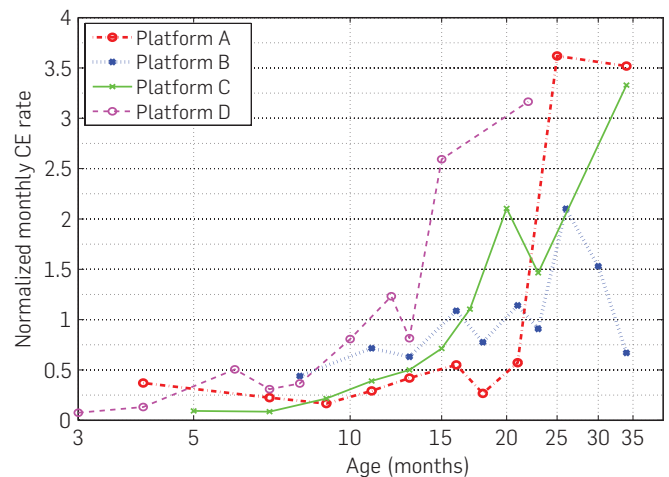
#### 4.4. DIMM capacity and chip size

Since the amount of memory used in typical server systems keeps growing from generation to generation, a commonly asked question, when projecting for future systems, is how an increase in memory affects the frequency of memory errors. In this section, we focus on one aspect of this question. We ask how error rates change, when increasing the capacity of individual DIMMs.

To answer this question we consider all DIMM types (type being defined by the combination of platform and manufacturer) that exist in our systems in two different capacities. Typically, the capacities of these DIMM pairs are either 1GB and 2GB, or 2GB and 4GB. Figure 6 shows for each of these pairs the factor by which the monthly probability of CEs, the CE rate, and the probability of UEs changes, when doubling capacity.

Figure 6 indicates a trend toward worse error behavior for increased capacities, although this trend is not consistent. While in some cases the doubling of capacity has a clear negative effect (factors larger than 1 in the graph), in others it has hardly any effect (factor close to 1 in the graph). For example, for Platform A, Mfg1 doubling the capacity increases UEs, but not CEs. Conversely, for Platform D, Mfg-6 doubling the

**Figure 5. The effect of age: The normalized monthly rate of experiencing a CE as a function of age by platform.**



capacity affects CEs, but not UEs.

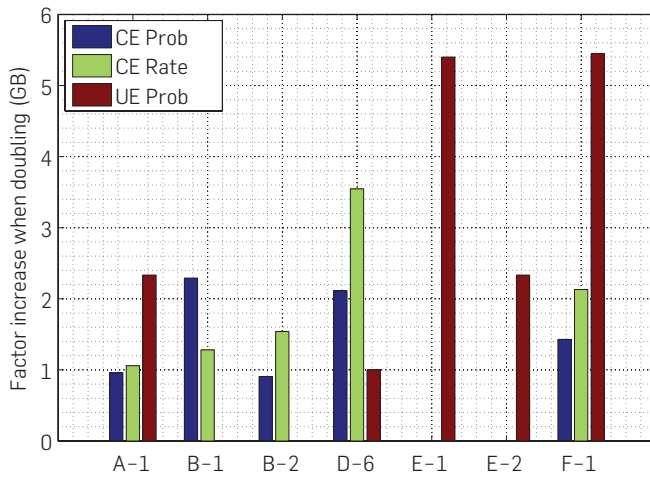
The difference in how scaling capacity affects errors might be due to differences in how larger DIMM capacities are built, since a given DIMM capacity can be achieved in multiple ways. For example, a 1Gb DIMM with ECC can be manufactured with 36 256-Mb chips, or 18 512-Mb chips or with 9 1-Gb chips.

We studied the effect of chip sizes on correctable and UEs, controlling for capacity, platform (dimm technology), and age. The results are mixed. When two chip configurations were available within the same platform, capacity and manufacturer, we sometimes observed an increase in average CE rates and sometimes a decrease. This either indicates that chip size does not play a dominant role in influencing CEs or there are other, stronger confounders in our data that we did not control for.

In addition to a correlation of chip size with error rates, we

Some bars are omitted, as we do not have data on UEs for Platform B and data on CEs for Platform E.

**Figure 6. Memory errors and DIMM capacity:** The graph shows for different Platform-Manufacturer pairs the factor increase in CE rates, CE probabilities and UE probabilities, when doubling the capacity of a DIMM.



also looked for correlations of chip size with incidence of correctable and UEs. Again we observe no clear trends. We also repeated the study of chip size effect without taking information on the manufacturer and/or age into account, again without any clear trends emerging.

The best we can conclude therefore is that any chip size effect is unlikely to dominate error rates given that the trends are not consistent across various other confounders, such as age and manufacturer.

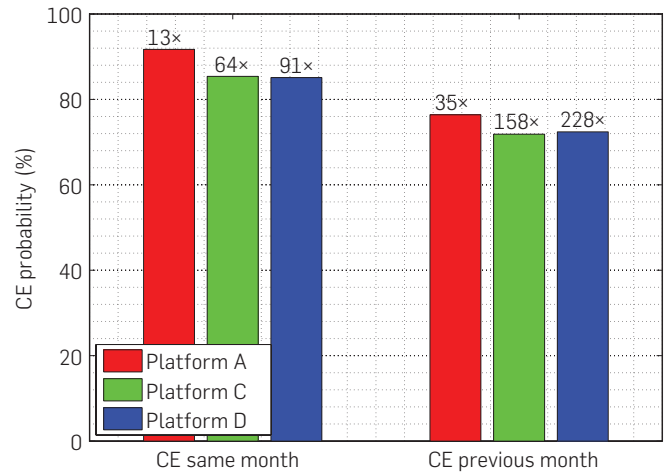
## 5. A CLOSER LOOK AT CORRELATIONS

The goal of this section is to study correlations between errors. Understanding correlations might help identify when a DIMM is likely to produce a large number of errors in the future and replace it before it starts to cause serious problems.

We begin by looking at correlations between CEs within the same DIMM. Figure 7 shows the probability of seeing a CE in a given month, depending on whether there were CEs in the same month (group of bars on the left) or the previous month (group of bars on the right). As the graph shows, for each platform the monthly CE probability increases dramatically in the presence of prior errors. In more than 85% of the cases a CE is followed by at least one more CE in the same month. Depending on the platform, this corresponds to an increase in probability between 13× to more than 90×, compared to an average month. Also seeing CEs in the previous month significantly increases the probability of seeing a CE: The probability increases by factors of 35× to more than 200×, compared to the case when the previous month had no CEs.

We also study correlations over time periods longer than a month and correlations between the number of errors in 1 month and the next, rather than just the probability of occurrence. Our study of the autocorrelation function for the number of errors observed per DIMM per month shows that even at lags of up to 7 months the level of correlation is still significant. When looking at the number of errors observed per month, we find that the larger the number of errors experienced in a month, the larger the expected number of errors in

**Figure 7. Correlations between correctable and uncorrectable errors:** The graph shows the probability of seeing a CE in a given month, depending on whether there were previously CEs observed in the same month (three left-most bars) or in the previous month (three right-most bars). The numbers on top of each bar show the factor increase in probability compared to the CE probability in a random month (three left-most bars) and compared to the CE probability when there was no CE in the previous month (three right-most bars).



the following month. For example, in the case of Platform C, if the number of CEs in a month exceeds 100, the expected number of CEs in the following month is more than 1,000. This is a 100× increase compared to the CE rate for a random month. Graphs illustrating the above findings and more details are included in the full paper.<sup>20</sup>

While the above observations let us conclude that CEs are predictive of future CEs, maybe the more interesting question is how CEs affect the probability of future *uncorrectable* errors. Since UEs are simply multiple bit corruptions (too many for the ECC to correct), one might suspect that the presence of CEs increases the probability of seeing a UE in the future. This is the question we focus on next.

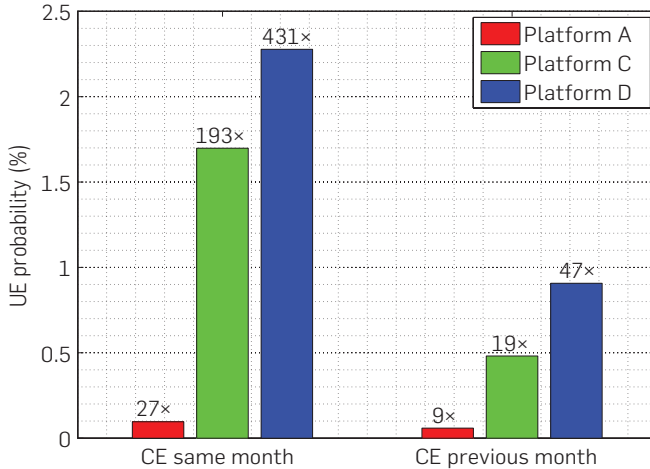
The three left-most bars in Figure 8 show how the probability of experiencing a UE in a given month increases if there are CEs in the same month. For all platforms, the probability of a UE is significantly larger in a month with CEs compared to a month without CEs. The increase in the probability of a UE ranges from a factor of 27× (for Platform A) to more than 400× (for Platform D). While not quite as strong, the presence of CEs in the preceding month also affects the probability of UEs. The three right-most bars in Figure 8 show that the probability of seeing a UE in a month following a month with at least one CEs is larger by a factor of 9× to 47× than if the previous month had no CEs.

We find that not only the presence, but also the rate of observed CEs in the same month affects the probability of a later UE. Higher rates of CEs translate to a higher probability of UEs. We see similar, albeit somewhat weaker trends when plotting the probability of UEs as a function of the number of CEs in the previous month. The UE probabilities are about

Throughout this section, when we say “in the same month” we mean within a 30-day period, rather than calendar month.



**Figure 8. Correlations between correctable and uncorrectable errors:** The graph shows the UE probability in a month depending on whether there were CEs earlier in the same month (three left-most bars) or in the previous month (three right-most bars). The numbers on top of the bars give the increase in UE probability compared to a month without CEs (three left-most bars) and the case where there were no CEs in the previous month (three right-most bars).



8× lower than if the same number of CEs had happened in the same month, but still significantly higher than in a random month.

Given the above observations, one might want to use CEs as an early warning sign for impending UEs. Another interesting view is therefore what fraction of UEs are actually preceded by a CE, either in the same month or the previous month. We find that 65%–80% of UEs are preceded by a CE in the same month. Nearly 20%–40% of UEs are preceded by a CE in the previous month. These probabilities are significantly higher than those in an average month.

The above observations lead to the idea of early replacement policies, where a DIMM is replaced once it experiences a significant number of CEs, rather than waiting for the first UE. However, while UE probabilities are greatly increased after observing CEs, the absolute probabilities of a UE are still relatively low (e.g., 1.7%–2.3% in the case of Platform C and Platform D, see Figure 8).

We also experimented with more sophisticated methods for predicting UEs, including CART (classification and regression trees) models based on parameters such as the number of CEs in the same and previous month, CEs and UEs in other DIMMs in the machine, DIMM capacity and model, but were not able to achieve significantly better prediction accuracy. Hence, replacing DIMMs solely based on CEs might be worth the price only in environments where the cost of downtime is high enough to outweigh the cost of the relatively high rate of false positives.

Our study of correlations and the presented evidence of correlations between errors, both in short and in longer time scales, might also shed some light on the common nature of errors. In simple terms, our results indicate that once a DIMM starts to experience errors it is likely to continue to have errors. This observation makes it more likely that most of the observed errors are due to hard errors, rather than soft errors. The occurrence of hard errors would also explain the

correlation between utilization and errors that we observed in Section 4.1.

## 6. SUMMARY AND DISCUSSION

This paper studied the incidence and characteristics of DRAM errors in a large fleet of commodity servers. Our study is based on data collected over more than 2 years and covers DIMMs of multiple vendors, generations, technologies, and capacities. Below, we briefly summarize our results and discuss their implications.

**Conclusion 1:** We found the incidence of memory errors and the range of error rates across different DIMMs to be much higher than previously reported.

A third of machines and over 8% of DIMMs in our fleet saw at least one CE per year. Our per-DIMM rates of CEs translate to an average of 25,000–75,000 FIT (failures in time per billion hours of operation) per Mb, while previous studies report 200–5,000 FIT per Mb. The number of CEs per DIMM is highly variable, with some DIMMs experiencing a huge number of errors, compared to others. The annual incidence of UEs was 1.3% per machine and 0.22% per DIMM.

**Conclusion 2:** More powerful error codes (chip-kill versus SECDED) can reduce the rate of UEs by a factor of 3–8.

We observe that platforms with more powerful error codes (chip-kill versus SECDED) were able to significantly reduce the rate of UEs (from 0.25%–0.4% per DIMM per year for SECDED-based platforms, to 0.05%–0.08% for chipkill based platforms). Nonetheless, the remaining incidence of UEs makes a crash-tolerant application layer indispensable for large-scale server farms.

**Conclusion 3:** There is no evidence that newer generation DIMMs have worse error behavior (even when controlling for DIMM age). There is also no evidence that one technology (DDR1, DDR2, FB-DIMM) or one manufacturer consistently outperforms the others.

There has been much concern that advancing densities in DRAM technology will lead to higher rates of memory errors in future generations of DIMMs. We study DIMMs in six different platforms, which were introduced over a period of several years, and observe no evidence that CE rates increase with newer generations. In fact, the DIMMs used in the three most recent platforms exhibit lower CE rates, than the two older platforms, despite generally higher DIMM capacities. This indicates that improvements in technology are able to keep up with adversarial trends in DIMM scaling.

**Conclusion 4:** Within the range of temperatures our production systems experience in the field, temperature has a surprisingly low effect on memory errors.

Temperature is well known to increase error rates. In fact, artificially increasing the temperature is a commonly used tool for accelerating error rates in lab studies. Interestingly,

we find that differences in temperature in the range they arise naturally in our fleet's operation (a difference of around 20°C between the 1st and 9th temperature decile) seem to have a marginal impact on the incidence of memory errors, when controlling for other factors, such as utilization.

**Conclusion 5:** Error rates are strongly correlated with utilization.

We find that DIMMs in machines with high levels of utilization, as measured by CPU utilization and the amount of memory allocated, see on average a 4–10 times higher rates of CEs, even when controlling for other factors, such as temperature.

**Conclusion 6:** DIMM capacity tends to be correlated with CE and UE incidence.

When considering DIMMs of the same type (manufacturer and hardware platform), that only differ in their capacity, we see a trend of increased CE and UE rates for higher capacity DIMMs. Based on our data we do not have conclusive results on the effect of chip size and chip density, but we are in the process of conducting a more detailed study that includes these factors.

**Conclusion 7:** The incidence of CEs increases with age.

Given that DRAM DIMMs are devices without any mechanical components, unlike for example hard drives, we see a surprisingly strong and early effect of age on error rates. For all DIMM types we studied, aging in the form of increased CE rates sets in after only 10–18 months in the field.

**Conclusion 8:** Memory errors are strongly correlated.

We observe strong correlations among CEs within the same DIMM. A DIMM that sees a CE is 13–228 times more likely to see another CE in the same month, compared to a DIMM that has not seen errors. Correlations exist at short time scales (days) and long time scales (up to 7 months).

We also observe strong correlations between CEs and UEs. Most UEs are preceded by one or more CEs, and the presence of prior CEs greatly increases the probability of later UEs. Still, the absolute probabilities of observing a UE following a CE are relatively small, between 0.1% and 2.3% per month, so replacing a DIMM solely based on the presence of CEs would be attractive only in environments where the cost of downtime is high enough to outweigh the cost of the expected high rate of false positives.

**Conclusion 9:** Error rates are unlikely to be dominated by soft errors.

The strong correlation between errors in a DIMM at both short and long time scales, together with the correlation between utilization and errors, leads us to believe that a large fraction of the errors are due to hard errors.

Conclusion 9 is an interesting observation, since much previous work has assumed that soft errors are the dominating

error mode in DRAM. Some earlier work estimates hard errors to be orders of magnitude less common than soft errors<sup>21</sup> and to make up about 2% of all errors.<sup>1</sup> Conclusion 9 might also explain the significantly higher rates of memory errors we observe compared to previous studies.

## Acknowledgments

We would like to thank Luiz Barroso, Urs Hoelzle, Chris Johnson, Nick Sanders, and Kai Shen for their feedback on drafts of this paper. We would also like to thank those who contributed directly or indirectly to this work: Kevin Bartz, Bill Heavlin, Nick Sanders, Rob Sprinkle, and John Zapisek. Special thanks to the System Health Infrastructure team for providing the data collection and aggregation mechanisms. Finally, the first author would like to thank the System Health Group at Google for hosting her during the summer of 2008. ☐

## References

1. Mosys adds soft-error protection, correction. *Semiconductor Business News* (28 Jan. 2002).
2. Al-Ars, Z., van de Goor, A.J., Braun, J., Richter, D. Simulation based analysis of temperature effect on the faulty behavior of embedded DRAMs. In *ITC'01: Proceedings of the 2001 IEEE International Test Conference* (2001).
3. Baumann, R. Soft errors in advanced computer systems. *IEEE Design Test Comput.* (2005), 258–266.
4. Borucki, L., Schindlbeck, G., Slayman, C. Comparison of accelerated DRAM soft error rates measured at component and system level. In *Proceedings of 46th Annual International Reliability Physics Symposium* (2008).
5. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E. Bigtable: A distributed storage system for structured data. In *Proceedings of OSDI'06* (2006).
6. Chen, C., Hsiao, M. Error-correcting codes for semiconductor memory applications: A state-of-the-art review. *IBM J. Res. Dev.* 28, 2 (1984), 124–134.
7. Dell, T.J. A white paper on the benefits of chipkill-correct ECC for PC server main memory. *IBM Microelectronics* (1997).
8. Hamamoto, T., Sugiura, S., Sawada, S. On the retention time distribution of dynamic random access memory (DRAM). *IEEE Trans. Electron Dev.* 45, 6 (1998), 1300–1309.
9. Johnston, A.H. Scaling and technology issues for soft error rates. In *Proceedings of the 4th Annual Conference on Reliability* (2000).
10. Li, X., Shen, K., Huang, M., Chu, L. A memory soft error measurement on production systems. In *Proceedings of USENIX Annual Technical Conference* (2007).
11. May, T.C., Woods, M.H. Alpha-particle-induced soft errors in dynamic memories. *IEEE Trans. Electron Dev.* 26, 1 (1979).
12. Messer, A., Bernadat, P., Fu, G., Chen, D., Dimitrijevic, Lie, D., Mannaru, D.D., Riska, R., Milojicic, D. Susceptibility of commodity systems and software to memory soft errors. *IEEE Trans. Comput.* 53, 12 (2004).
13. Milojicic, D., Messer, A., Shau, J., Fu, G., Munoz, A. Increasing relevance of memory hardware errors: A case for recoverable programming models. In *Proceedings of the 9th ACM SIGOPS European workshop* (2000).
14. Mukherjee, S.S., Emer, J., Fossum, T., Reinhardt, S.K. Cache scrubbing in microprocessors: Myth or necessity? In *PRDC '04: Proceedings of the 10th IEEE Pacific Rim International Symposium on Dependable Computing* (2004).
15. Mukherjee, S.S., Emer, J., Reinhardt, S.K. The soft error problem: An architectural perspective. In *HPCA '05: Proceedings of the 11th International Symposium on High-Performance Computer Architecture* (2005).
16. Normand, E. Single event upset at ground level. *IEEE Trans. Nucl. Sci.* 6, 43 (1996), 2742–2750.
17. O'Gorman, T.J., Ross, J.M., Taber, A.H., Ziegler, J.F., Muhlfeld, H.P., Montrose, C.J., Curtis, H.W., Walsh, J.L. Field testing for cosmic ray soft errors in semiconductor memories. *IBM J. Res. Dev.* 40, 1 (1996).
18. Schroeder, B., Gibson, G.A. A large scale study of failures in high-performance computing systems. In *DSN 2006: Proceedings of the International Conference on Dependable Systems and Networks* (2006).
19. Schroeder, B., Gibson, G.A. Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? In *5th USENIX FAST Conference* (2007).
20. Schroeder, B., Pinheiro, E., Weber, W.-D. DRAM errors in the wild: A large-scale field study. In *Proceedings of ACM SIGMETRICS* (2009).
21. Takeuchi, K., Shimohigashi, K., Kozuka, H., Toyabe, T., Itoh, K., Kurosawa, H. Origin and characteristics of alpha-particle-induced permanent junction leakage. *IEEE Trans. Electron Dev.* (Mar. 1999).
22. Ziegler, J.F., Lanford, W.A. Effect of cosmic rays on computer memories. *Science* 206 (1979), 776–788.

**Bianca Schroeder** (bianca@cs.toronto.edu), Computer Science Department, University of Toronto, Toronto, Canada.

**Wolf-Dietrich Weber**, Google Inc., Mountain View, CA.

**Eduardo Pinheiro**, Google Inc., Mountain View, CA.

© 2011 ACM 0001-0782/11/0200 \$10.00

## 3M

### Lead Interactivity Researcher

The Display and Graphics Business Laboratory (DGBL) is looking for a Lead Interactivity Researcher.

#### Key responsibilities for this position include:

- ▶ Conducting applied research in novel forms of human-computer, human-device, and device-device interaction
- ▶ Designing and implementing novel user interfaces for applications and devices
- ▶ Recommending and exploring novel lines of research
- ▶ Evaluating research ideas
- ▶ Preparing presentations and publications
- ▶ Building and evaluating prototype systems and devices
- ▶ Advising/mentoring junior researchers

#### Basic Qualifications:

- ▶ Bachelor's degree from an accredited institution in Computer Science or Computer Engineering is required
- ▶ Minimum of five (5) years experience leading a corporate and/or academic research program is required

Apply URL: <http://jobs.3m.com/job/Austin-Lead-Interactivity-Researcher-Job-TX-73301/1037669/>

### Austin Peay State University Assistant Professor/Computer Science

The Dept of Computer Science & Information Tech at Austin Peay State University invites applications for a tenure-track assistant professor position in computer science beginning August 2011. For more information, see <http://bit.ly/g2DTPO>

### Company Name: Confidential Analyst, London, UK

A Business/ Technical Analyst role for an international media group based in Covent Garden, LONDON, UK.

Responsibilities will include analysis of requirements, proposed solutions, functional and technical specifications. Team technologies include: distributed systems, relational databases, data mining, codecs, pattern recognition. This is a great opportunity to join an exciting technology company working with the largest media companies in the world.

Please send CVs attached to developerapplications2010@gmail.com.

### DePaul University Assistant/Associate Professor

The School of Computing at DePaul University invites applications for a tenure-track position in distributed systems. We seek candidates with a research interest in data-intensive distributed systems, cloud computing, distributed databases, or closely related areas. For more information, see <https://facultyopportunities.depaul.edu/applicants/Central?quickFind=50738>.

### DePaul University College of Computing and Digital Media Assistant/Associate Professor in Game Design

The College of Computing and Digital Media at DePaul University invites applications for a tenure-track position in game design. Areas of interest incl. mechanic design; experimental games; level design; prototyping; serious games. Industry exp. a plus. For more information, see <https://facultyopportunities.depaul.edu/applicants/Central?quickFind=50739>

### Hawai'i Pacific University Assistant/Associate/Professor of Computer Science

Hawai'i Pacific University's Department of Mathematics and Computer Science invites applications for a career-track Assistant or Associate Professor, or Professor of Computer Science position to start in Fall, 2011. Applicants should have a Ph.D., Ed.D., or D.B.A. in Computer Science, Information Systems, or a related field; and teaching experience at the university level. Preferred qualifications include three or more years experience teaching in a computer science or related degree program, and experience as a CS/IS professional. The successful candidate will teach various courses in undergraduate and possibly graduate computer science; contribute to department goals such as curriculum development; maintain scholarly and professional development; and contribute to the governance of the university.

HPU is the largest private university in Hawai'i, with about 9000 students from over 100 countries. The University has three campuses linked by shuttle: a large and vibrant downtown campus in Honolulu, a scenic 135-acre campus in the green foothills of the windward side of Oahu, and the oceanfront Oceanic Institute for marine sciences. HPU also has an extensive military program. The Department of Mathematics and Computer Science is part of the College of Natural and Computational Sciences at HPU. Information about the HPU computer science program can be found at [www.hpu.edu/cs](http://www.hpu.edu/cs).

To apply, please visit our employment website: [www.hpu.edu/employment](http://www.hpu.edu/employment) and mail a letter of application, official transcripts, curriculum vitae, statement of teaching philosophy, list of publications, and three letters of recommendation that include an assessment of teaching abilities by mail to: Human Resources, Faculty Position in Computer Science, 1132 Bishop Street, Suite 310, Honolulu, HI 96813. HPU is an equal opportunity employer.

### Max Planck Institute for Informatics Junior Research Groups Leaders in the Max Planck Center for Visual Computing and Communication

The Max Planck Institute for Informatics (MPII), as the coordinator of the Max Planck Center for Visual Computing and Communication (MPC-VCC), invites applications for

### Junior Research Groups Leaders in the Max Planck Center for Visual Computing and Communication

The Max Planck Center for Visual Computing and Communications offers young scientists in information technology the opportunity to develop their own research program addressing important problems in areas such as image communication, computer graphics, geometric computing, imaging systems, computer vision, human machine interface, distributed multimedia architectures, multimedia networking, and visual media security. The center includes an outstanding group of faculty members at Stanford's Computer Science and Electrical Engineering Departments, the Max Planck Institute for Informatics, and Saarland University.

The program begins with a preparatory 1-2 year postdoc phase (**Phase P**) at the Max Planck Institute for Informatics, followed by a two-year appointment at Stanford University (**Phase I**) as a visiting assistant professor, and then a position at the Max Planck Institute for Informatics as a junior research group leader (**Phase II**). However, the program can be entered flexibly at each phase, commensurate with the experience of the applicant.

Applicants to the program must have completed an outstanding PhD. Exact duration of the preparatory postdoc phase is flexible, but we typically expect this to be about 1-2 years. Applicants who completed their PhD in Germany may enter Phase I of the program directly. Applicants for Phase II are expected to have completed a postdoc stay abroad and must have demonstrated their outstanding research potential and ability to successfully lead a research group.

The Max Planck Center is an equal opportunity employer and women are encouraged to apply.



Additional information is available on the website <http://www.mpc-vc.de>

Reviewing of applications will commence on **January 31, 2011**. The final deadline is **March 31, 2011**. Applicants should submit their CV, copies of their school and university reports, list of publications, reprints of five selected publications, names of references, a brief description of their previous research and a detailed description of the proposed research project (including possible opportunities for collaboration with existing research groups at Saarbrücken and Stanford) to:

Prof. Dr. Hans-Peter Seidel, Max Planck Institute for Informatics, Campus E 1 4, 66123 Saarbrücken, Germany, Email: [hpseidel@mpi-inf.mpg.de](mailto:hpseidel@mpi-inf.mpg.de)

### Max Planck Institute for Software Systems (MPI-SWS)

#### Tenure-track openings

Applications are invited for tenure-track and tenured faculty positions in all areas related to the study, design, and engineering of software systems. These areas include, but are not limited to, data and information management, programming systems, software verification, parallel, distributed and networked systems, and embedded systems, as well as cross-cutting areas like security, machine learning, usability, and social aspects of software systems. A doctoral degree in computer science or related areas and an outstanding research record are required. Successful candidates are expected to build a team and pursue a highly visible research agenda, both independently and in collaboration with other groups. Senior candidates must have demonstrated leadership abilities and recognized international stature.

MPI-SWS, founded in 2005, is part of a network of eighty Max Planck Institutes, Germany's premier basic research facilities. MPIs have an established record of world-class, foundational research in the fields of medicine, biology, chemistry, physics, technology and humanities. Since 1948, MPI researchers have won 17 Nobel prizes. MPI-SWS aspires to meet the highest standards of excellence and international recognition with its research in software systems.

To this end, the institute offers a unique environment that combines the best aspects of a university department and a research laboratory:

- a)** Faculty receive generous base funding to build and lead a team of graduate students and post-docs. They have full academic freedom and publish their research results freely.
- b)** Faculty supervise doctoral theses, and have the opportunity to teach graduate and undergraduate courses.
- c)** Faculty are provided with outstanding technical and administrative support facilities as well as internationally competitive compensation packages.

MPI-SWS currently has 8 tenured and tenure-track faculty, and is funded to support 17 faculty and about 100 doctoral and post-doctoral positions. Additional growth through outside funding is possible. We maintain an open, international and diverse work environment and seek applications from outstanding researchers regardless of national origin or citizenship. The working language is English; knowledge of the German language is not required for a successful career at the institute.



UNIVERSITÄT  
DES  
SAARLANDES



# Saarland University is seeking to establish several Junior Research Groups (W1/W2)

within the Cluster of Excellence "Multimodal Computing and Interaction" which was established by the German Research Foundation (DFG) within the framework of the German Excellence Initiative.

The term "multimodal" describes the different types of digital information such as text, speech, images, video, graphics, and high-dimensional data, and the way it is perceived and communicated, particularly through vision, hearing, and human expression. The challenge is now to organize, understand, and search this multimodal information in a robust, efficient and intelligent way, and to create dependable systems that allow natural and intuitive multimodal interaction. We are looking for highly motivated young researchers with a background in the research areas of the cluster, including algorithmic foundations, secure and autonomous networked systems, open science web, information processing in the life sciences, visual computing, large-scale virtual environments, synthetic virtual characters, text and speech processing and multimodal dialog systems. Additional information on the Cluster of Excellence is available on <http://www.mmci.uni-saarland.de>. Group leaders will receive junior faculty status at Saarland University, including the right to supervise Bachelor, Master and PhD students. Positions are limited to five years.

Applicants for W1 positions (phase I of the program) must have completed an outstanding PhD. Upon successful evaluation after two years, W1 group leaders are eligible for promotion to W2. Direct applicants for W2 positions (phase II of the program) must have completed a postdoc stay and must have demonstrated outstanding research potential and the ability to successfully lead their own research group. Junior research groups are equipped with a budget of 80k to 100k Euros per year to cover research personnel and other costs.

Saarland University has leading departments in computer science and computational linguistics, with more than 200 PhD students working on topics related to the cluster (see <http://www.informatik-saarland.de> for additional information). The German Excellence Initiative recently awarded multi-million grants to the Cluster of Excellence "Multimodal Computing and Interaction" as well as to the "Saarbrücken Graduate School of Computer Science". An important factor to this success were the close ties to the Max Planck Institute for Informatics, the German Research Center for Artificial Intelligence (DFKI), and the Max Planck Institute for Software Systems which are co-located on the same campus.

Candidates should submit their application (curriculum vitae, photograph, list of publications, short research plan, copies of degree certificates, copies of the five most important publications, list of five references) to Conny Liegl, Cluster of Excellence, Campus E1 7, 66123 Saarbrücken, Germany. Please, also send your application as a single PDF file to [applications@mmci.uni-saarland.de](mailto:applications@mmci.uni-saarland.de).

The review of applications will begin on February 18, 2011: all applicants are strongly encouraged to submit applications by that date. Final decisions will be made following a candidate symposium that will be held in the week of March 21 – 25, 2011.

Saarland University is an equal opportunity employer. In accordance with its policy of increasing the proportion of women in this type of employment, the University actively encourages applications from women. For candidates with equal qualification, preference will be given to people with physical disabilities.

The institute is located in Kaiserslautern and Saarbrücken, in the tri-border area of Germany, France and Luxembourg. The area offers a high standard of living, beautiful surroundings and easy access to major metropolitan areas in the center of Europe, as well as a stimulating, competitive and collaborative work environment. In immediate proximity are the MPI for Informatics, Saarland University, the Technical University of Kaiserslautern, the German Center for Artificial Intelligence (DFKI), and the Fraunhofer Institutes for Experimental Software Engineering and for Industrial Mathematics.

Qualified candidates should apply online at <http://www.mpi-sws.org/application>. The review of applications will begin on January 3, 2011, and applicants are strongly encouraged to apply by that date; however, applications will continue to be accepted through January 2011.

The institute is committed to increasing the representation of minorities, women and individuals with physical disabilities in Computer Science. We particularly encourage such individuals to apply.

---

### **Polytechnic of Namibia** **Professors/Associate Professors/ Senior Lecturers**

To lecture in areas of specialization, develop curricula and study materials, participate in administrative responsibilities at departmental/faculty or institutional level; conduct research and consult with industry. Identify, propose and manage relevant departmental projects and sources of funding.

We are looking applicants with broad & deep knowledge and experience in one of the following fields: Software Engineering; Basic Computer Studies; Business Computing; Computer Systems & Network; Mathematic foundations of information Technology and any other related fields.

The successful candidate is expected to foster new and existing areas of research and take the curriculum lead. See our website for further information: [www.sit.polytechnic.edu.na](http://www.sit.polytechnic.edu.na)

---

### **Princeton University** **Computer Science** **Assistant Professor** **Tenure-Track Positions**

The Department of Computer Science at Princeton University invites applications for faculty positions at the Assistant Professor level. We are accepting applications in all areas of Computer Science.

Applicants must demonstrate superior research and scholarship potential as well as teaching ability. A PhD in Computer Science or a related area is required.

Successful candidates are expected to pursue an active research program and to contribute significantly to the teaching programs of the department. Applicants should include a resume contact information for at least three people who can comment on the applicant's professional qualifications.

There is no deadline, but review of applications will start in December 2010; the review of applicants in the field of theoretical computer science will begin as early as October 2010.

Princeton University is an equal opportunity employer and complies with applicable EEO and

affirmative action regulations. You may apply online at:

<http://www.cs.princeton.edu/jobs> Requisition Number: 1000520

---

### **Princeton University** **Computer Science Department** **Postdoc Research Associate**

The Department of Computer Science at Princeton University is seeking applications for post-doctoral or more senior research positions in theoretical computer science. Candidates will be affiliated with the Center for Computational Intractability (CCI) or the Princeton Center for Theoretical Computer Science. Candidates should have a PhD in computer science, a related field, or on track to finish by August 2011. Candidates affiliated with the CCI will have visiting privileges at partner institutions NYU, Rutgers University, and The Institute for Advanced Study. Review of candidates will begin Jan 1, 2011, and will continue until positions are filled. Applicants should submit a CV, research statement, and contact information for three references.

Princeton University is an equal opportunity employer and complies with applicable EEO and affirmative action regulations. Apply to: <http://jobs.princeton.edu/requisition#1000829>

---

### **Profinity** **Senior Web/SQL developer**

We are looking for experienced web developers with strong SQL server 2005+ experience. The job is integrating new customers into an existing order/payment processing system using classic ASP, T-SQL, Python, and Powershell. Work will be from home and communication will be over Skype, phone, & email.

Transact SQL, Microsoft technologies. Problem solving skills are paramount. This job will entail converting business stakeholder needs into software, so excellent communication skills and patience is a must. Apply:

[http://www.profinity.com/contact\\_us.asp](http://www.profinity.com/contact_us.asp)

---

### **Texas A&M University** **Department of Visualization** **Assistant Professor**

Tenure-track faculty in the area of interactive media. Responsibilities include research/creative work, advising graduate/undergraduate levels, service to dept, university & field, teaching inc. intro courses in game design & development.

Candidates must demonstrate collaborative efforts across disciplinary lines. Graduate degree related to game design & development, mobile media, interactive graphics, interactive art, multimedia or simulation is required. Apply URL: <http://www.viz.tamu.edu>

---

### **Toyota Technological Institute** **at Chicago** **Computer Science Faculty Positions at All Levels**

Toyota Technological Institute at Chicago (TTIC) is a philanthropically endowed degree-granting

institute for computer science located on the University of Chicago campus. The Institute is expected to reach a steady-state of 12 traditional faculty (tenure and tenure track), and 12 limited term faculty. Applications are being accepted in all areas, but we are particularly interested in

- Theoretical computer science
- Speech processing
- Machine learning
- Computational linguistics
- Computer vision
- Computational biology
- Scientific computing

Positions are available at all ranks, and we have a large number of limited term positions currently available.

For all positions we require a Ph.D. Degree or Ph.D. candidacy, with the degree conferred prior to date of hire. Submit your application electronically at: <http://ttic.uchicago.edu/facapp/>

*Toyota Technological Institute at Chicago is an  
Equal Opportunity Employer*

---

### **University of Oregon** **Visiting Asst. Professor**

The Department of Computer and Information Science (CIS) at the University of Oregon invites applications for two post-doctoral positions. The positions will involve both research (in support of one of the projects listed below) and teaching (two courses a year). Both aspects of the position will be supported by tenure-track faculty mentors. The expected outcomes include career development of the position holder as well as valuable research and education contributions to the department. These positions are anticipated as two-year appointments based on satisfactory performance. There will be opportunities for supplemental research summer support as well as the possibility of one-year extension. Employment begins September 16, 2011.

Applicants must have a Ph.D. in computer science or a closely related field, a demonstrated record of excellence in research, a strong commitment to teaching and engagement with students. The positions are limited to recent PhDs (within 3 years of graduation) who have not held tenure-track faculty positions.

The University of Oregon is an AAU research university located in Eugene and within one-hour drive of both the Pacific Ocean and the snow-capped Cascade Mountains. The CIS department offers a stimulating and friendly environment for collaborative teaching and research both within the department and with other departments on campus. The department's primary research emphases are in the areas of networking, programming languages, parallel and distributed computing, automated reasoning, human-computer interaction, and computer and network security. More information about the department, its programs, and current faculty can be found at <http://www.cs.uoregon.edu>

Applications will be accepted electronically through the department's web site (only). Application information can be found at <http://www.cs.uoregon.edu/Employment/>. Applicants should submit their curriculum vitae, names of three references, a statement of research and teaching interests, a proposal for contributions to the identi-

fied project, and selected publications through the website. Review of applications will begin Feb. 15, 2011 and continue until the position is filled.

The University of Oregon is an Equal Opportunity/Affirmative Action institution committed to cultural diversity and compliance with the Americans with Disabilities Act.

The University of Oregon is committed to create a more inclusive and diverse institution and seeks candidates with demonstrated potential to contribute positively to its diverse community.

**The research projects for the two positions are as follows:**

**Project A:**

The candidate will be performing novel research on ensuring the security and privacy of home-care rehabilitation delivery systems and devices. A strong background in operating systems, networks, and security is required, with expertise in embedded systems (specifically kernel programming in embedded environments) and consumer device environments (e.g., Android, Google TV, Wii) being highly desirable. Experience with traffic analysis and protection mechanisms in home networking environments is also desirable, as is a familiarity with cloud computing environments.

The successful candidate will play a key role in a research team, and be involved in writing grant proposals, and teaching. This candidate will work with Drs. Jun Li, Stephen Fickas, and Kevin Butler, as well as talented and motivated graduate and undergraduate students within the department. The successful candidate will be

mentored to facilitate transition to an independent research career through opportunities to publish in top-tier conferences and journals, as well as opportunities to gain leadership experience through managing a research team working in an exciting and innovative area.

**Project B:**

The candidate in the area of machine learning and databases will be mentored by Professors Dejing Dou and Daniel Lowd. We are especially interested in a researcher who can work on projects at the interface between statistical machine learning and data management. Projects would involve developing novel machine learning algorithms and representations and designing scalable, reliable databases and web-based computing systems. Application areas include biomedical data mining, health care informatics, social network analysis, and information extraction from the web. Collaborations with other professors specializing in high performance computing, cloud computing, and social networks are also possible.

Apply URL:

<http://www.cs.uoregon.edu/Employment/>

**Utah State University  
Assistant Professor**

Applications are invited for a faculty position at the Assistant Professor level, for employment beginning Fall 2011. Applicants must have completed a PhD in computer science by the time of appointment. The position requires demonstrat-

ed research success, a significant potential for attracting external research funding, excellence in teaching both undergraduate and graduate courses, the ability to supervise student research, and excellent communication skills.

USU offers competitive salaries and outstanding medical, retirement, and professional benefits (see <http://www.usu.edu/hr/> for details). The department currently has approximately 280 undergraduate majors, 80 MS students and 27 PhD students. There are 17 full time faculty. The BS degree is ABET accredited. Utah State University is a Carnegie Research Doctoral extensive University of over 23,000 students, nestled in a mountain valley 80 miles north of Salt Lake City, Utah. Opportunities for a wide range of outdoor activities are plentiful. Housing costs are at or below national averages, and the area provides a supportive environment for families and a balanced personal and professional life. Women, minority, veteran and candidates with disabilities are encouraged to apply. USU is sensitive to the needs of dual-career couples. Utah State University is an affirmative action/equal opportunity employer, with a National Science Foundation ADVANCE Gender Equity program, committed to increasing diversity among students, faculty, and all participants in university life. Applications must be submitted using USU's online job-opportunity system. To access this job opportunity directly and begin the application process, visit <https://jobs.usu.edu/applicants/Central?quickFind=55484>.

The review of the applications will begin on January 15, 2011 and continue until the position is filled. The salary will be competitive and depend on qualifications.

# Take Advantage of ACM's Lifetime Membership Plan!

- ◆ **ACM Professional Members** can enjoy the convenience of making a single payment for their entire tenure as an ACM Member, and also be protected from future price increases by taking advantage of **ACM's Lifetime Membership** option.
- ◆ **ACM Lifetime Membership** dues may be tax deductible under certain circumstances, so becoming a Lifetime Member can have additional advantages if you act before the end of 2011. (Please consult with your tax advisor.)
- ◆ Lifetime Members receive a certificate of recognition suitable for framing, and enjoy all of the benefits of **ACM Professional Membership**.

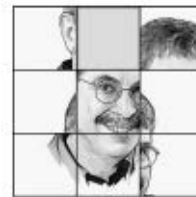
Learn more and apply at:  
**<http://www.acm.org/life>**



Association for  
Computing Machinery

*Advancing Computing as a Science & Profession*





DOI:10.1145/1897816.1897845

Peter Winkler

# Puzzled

## Parsing Partitions

*Welcome to three new puzzles. Solutions to the first two will be published next month; the third is (as yet) unsolved. In each, the issue is how your intuition matches up with the mathematics.*

The theme is partitions. Recall from freshman year that a set  $A$  is a subset of a set  $B$  if every element of  $A$  is also in  $B$ . A partition of a set  $S$  is a collection of subsets of  $S$  such that every element of  $S$  is in exactly one of the subsets in the collection. Pretty basic, right? But “basic” is not the same as “easy.” Try proving the following reasonable-looking statements about partitions of a fifth-grade class.

**1.** On Monday, Ms. Feldman partitioned her fifth-grade class into  $k$  subsets (of various sizes) to work on different projects. On Tuesday, she repartitioned the same students into  $k+1$  subsets. Show that at least two students were in smaller subsets on Tuesday than they were on Monday.

**2.** On Wednesday, Ms. Feldman divided her class into just two parts, but a little too much socialization emerged in each of them, distracting the students from the work at hand. The next day (Thursday) she is again determined to partition the class into two subsets but this time in such a way that no student has more than half of his/her own friends in his/her own subset. Show that such a partition always exists.

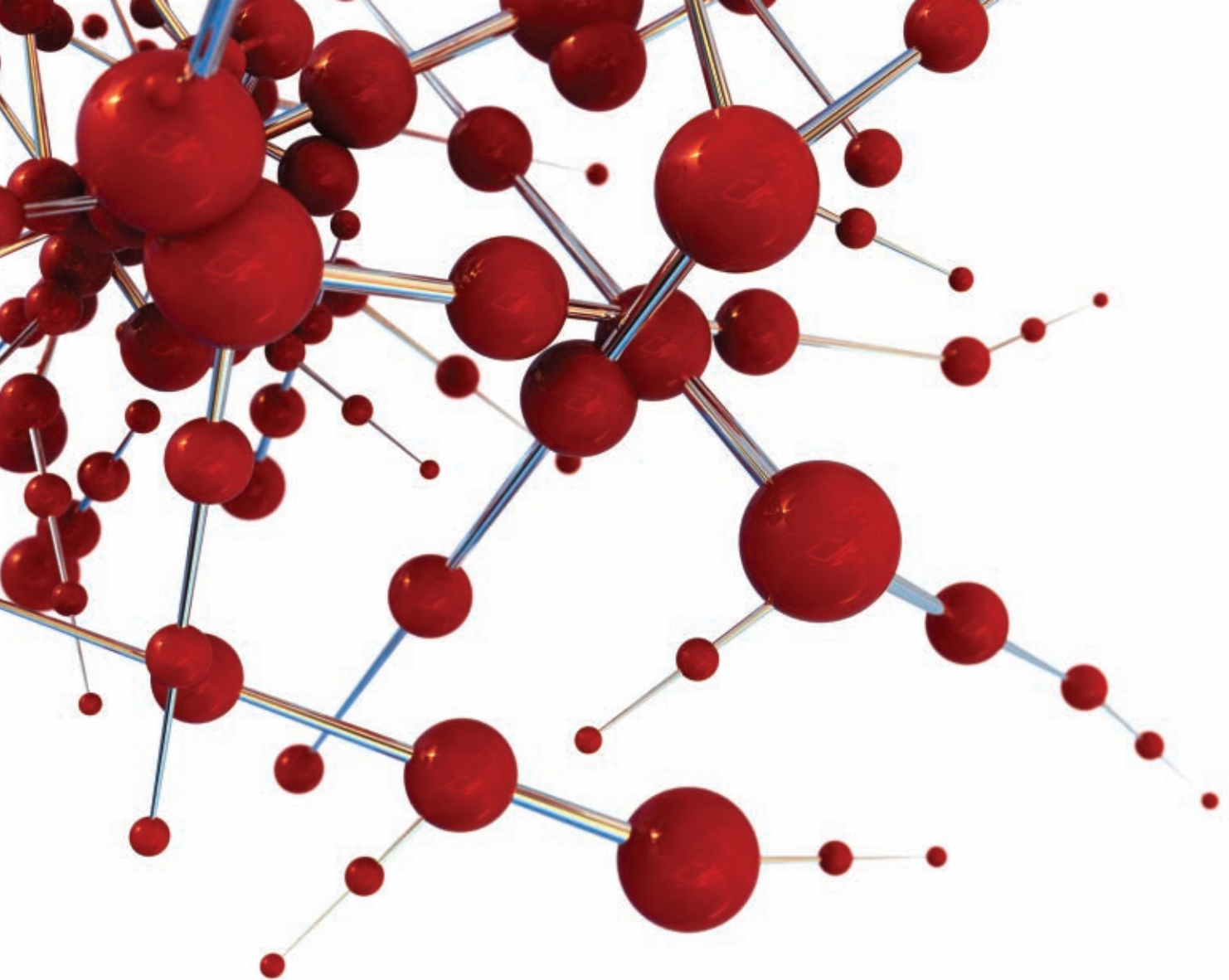
**3.** Now it's Friday, and all other fifth-grade teachers are out sick. This means Ms. Feldman is in charge of the entire fifth grade, which, to her consternation, has (countably) infinitely many students. Persevering, she is again determined to partition the students into two subsets in such a way that no student has more friends in his/her own subset than in the other subset.

Is it guaranteed that no matter how the friendships are structured there is always a way to do this?

In Puzzles 2 and 3, we assumed that friendship is a symmetric relation; that is, if student  $X$  is a friend of student  $Y$ , then the reverse is true as well. In Puzzle 3, some students may have infinitely many friends; it is OK if such a student has infinitely many friends in his/her own subset, provided infinitely many friends are also in the other subset. So it shouldn't be difficult to find a partition with the desired property. Right? So why can't anyone prove it?

All readers are encouraged to submit prospective puzzles for future columns to [puzzled@cacm.acm.org](mailto:puzzled@cacm.acm.org).

**Peter Winkler** ([puzzled@cacm.acm.org](mailto:puzzled@cacm.acm.org)) is Professor of Mathematics and of Computer Science and Albert Bradley Third Century Professor in the Sciences at Dartmouth College, Hanover, NH.



**CONNECT WITH OUR  
COMMUNITY OF EXPERTS.**

**[www.reviews.com](http://www.reviews.com)**



Association for  
Computing Machinery

**Reviews.com**

They'll help you find the best new books  
and articles in computing.

**Computing Reviews is a collaboration between the ACM and Reviews.com.**

**October 22–27, 2011**

**Co-located with SPLASH/OOPSLA**

**Hilton Portland & Executive Tower**

**Portland, Oregon USA**

# **ONWARD! 2011**

**ACM Conference on New Ideas in  
Programming and Reflections on Software**

**Submissions for papers, workshops, essays, and films >> April 8, 2011**

**Chair**

Robert Hirschfeld

Hasso-Plattner-Institut Potsdam, Germany

chair@onward-conference.org

**Papers**

Eelco Visser

Delft University of Technology, The Netherlands

papers@onward-conference.org

**Workshops**

Pascal Costanza

Vrije Universiteit Brussel, Belgium

workshops@onward-conference.org

**Essays**

David West

New Mexico Highlands University, USA

essays@onward-conference.org

**Films**

Bernd Bruegge

Technische Universität München, Germany

films@onward-conference.org



Association for  
Computing Machinery

**<http://onward-conference.org/>**