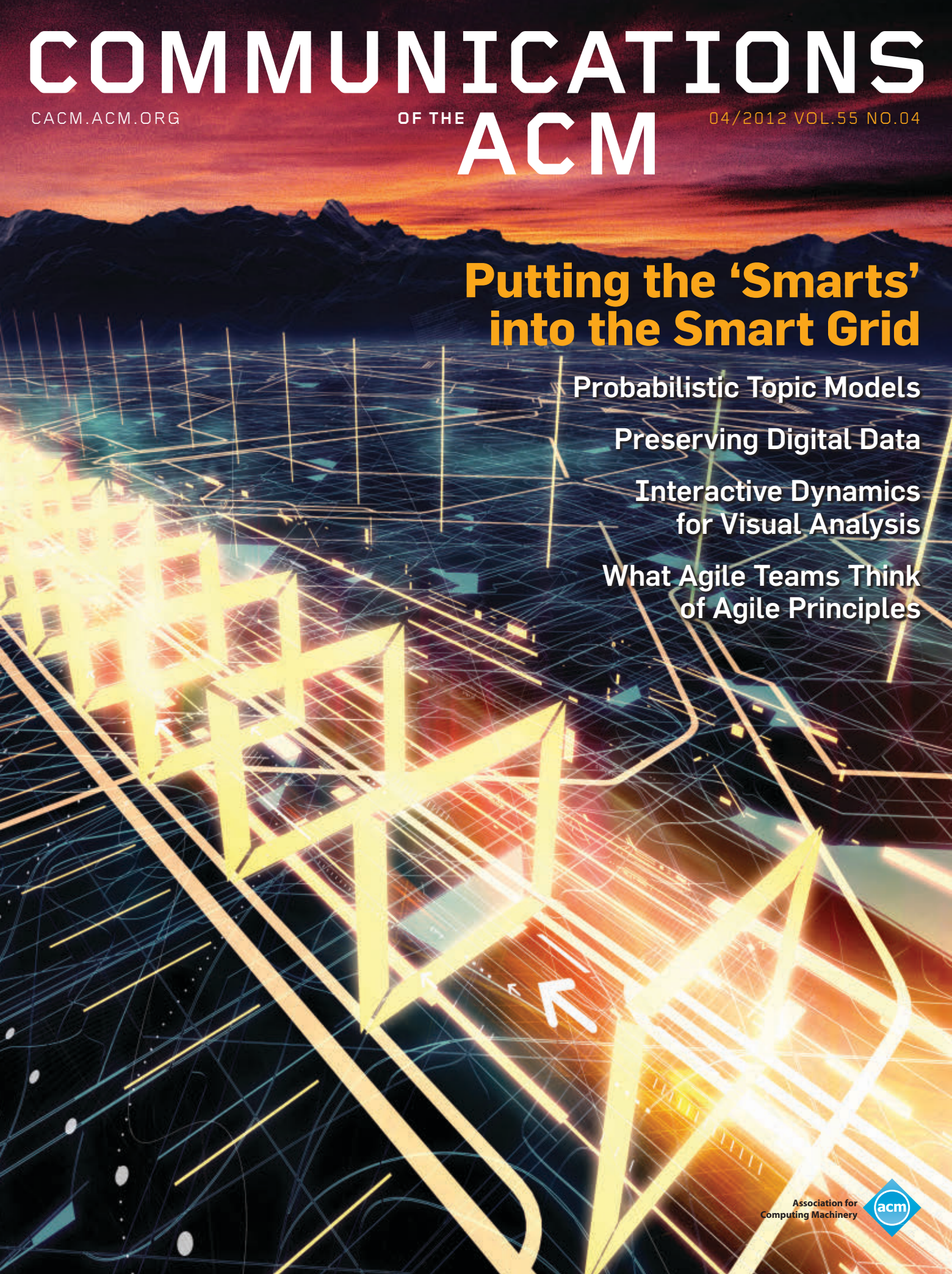
COMMUNICATIONS OF THE ACM
CACM.ACM.ORG
04/2012 VOL.55 NO.04



Putting the 'Smarts' into the Smart Grid

Probabilistic Topic Models

Preserving Digital Data

Interactive Dynamics for Visual Analysis

What Agile Teams Think of Agile Principles

Association for Computing Machinery

NEW!

CLOUD SERVER CONTROL AT YOUR FINGERTIPS

Only pay for what you need.
Change your server specifications anytime!

- Adaptable with up to 6 CPU, 24 GB of RAM, and 800 GB hard drive space
- On-the-fly resource allocation – hourly billing
- Dedicated resources with full root access
- Linux or Windows® operating systems available with Parallels® Plesk Panel 10.4
- Free SSL Certificate included
- 2,000 GB Traffic
- 24/7 Hotline and Support
- 1&1 servers are housed in high-tech data centers owned and operated by 1&1

1&1 DYNAMIC CLOUD SERVER

3 MONTHS FREE!*

Base Configuration, then $49/month

NEW: Monitor and manage servers through 1&1 mobile apps for Android™ and iPhone®.

MEMBER OF united internet

1-877-461-2631 www.1and1.com

1-855-221-2631 www.1and1.ca

1&1®

* 3 months free based on the basic configuration ($49/month) for a total savings of $147. Setup fee and other terms and conditions may apply. Visit www.1and1.com for full promotional offer details. Program and pricing specifi cations and availability subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet, all other trademarks are the property of their respective owners.


Frederick P. BROOKS, JR.


Charles P. (Chuck) THACKER


Charles W. BACHMAN


Stephen A. COOK


Barbara LISKOV


Richard M. KARP


Leslie G. VALIANT


Marvin MINSKY


Judea PEARL


Adi SHAMIR


Richard E. STEARNS


John HOPCROFT


Robert E. KAHN


Vinton G. CERF


Robert TARJAN


Ivan SUTHERLAND


Butler LAMPSON


Juris HARTMANIS


Andrew C. YAO


Donald E. KNUTH


Dana S. SCOTT


Raj REDDY


Fernando J. CORBATÓ


Edward A. FEIGENBAUM


Alan C. KAY


William (Velvel) KAHAN


Frances E. ALLEN


E. Allen EMERSON


Niklaus WIRTH


Ken THOMPSON


Leonard ADLEMAN


Ronald RIVEST


Edmund CLARKE


Joseph SIFAKIS

34 TURING AWARD WINNERS TOGETHER?

WHAT DO YOU GET WHEN YOU PUT

THE ACM TURING CENTENARY CELEBRATION

A ONCE-IN-A-LIFETIME EVENT THAT YOU'LL NEVER FORGET... AND YOU'RE INVITED!

CHAIR:
Vint Cerf, *'04 Turing Award Winner, Chief Internet Evangelist, Google*

PROGRAM CO-CHAIRS:
Mike Schroeder, *Microsoft Research*
John Thomas, *IBM Research*
Moshe Vardi, *Rice University*

MODERATOR:
Paul Saffo, *Managing Director of Foresight, Discern Analytics* and *Consulting Associate Professor, Stanford*

WHEN:
June 15th & June 16th 2012

WHERE:
Palace Hotel, San Francisco CA

HOW:
Register for the event at turing100.acm.org

REGISTRATION IS FREE-OF-CHARGE!

acm | A.M. TURING

THE A.M. TURING AWARD, often referred to as the "Nobel Prize" in computing, was named in honor of Alan Mathison Turing (1912–1954), a British mathematician and computer scientist. He made fundamental advances in computer architecture, algorithms, formalization of computing, and artificial intelligence. Turing was also instrumental in British code-breaking work during World War II.

Departments

Last Byte

News

Viewpoints

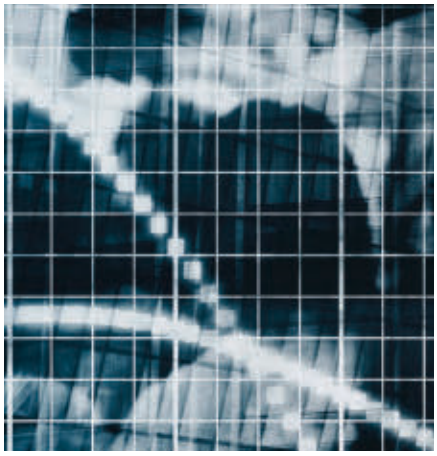
PHOTOGRAPH BY ALICIA KUBISTA

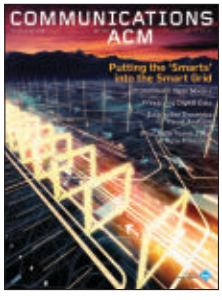
04/2012
VOL. 55 NO. 04

Practice

Articles' development led by acmqueue
queue.acm.org

Contributed Articles





About the Cover:
To build the smart grid of the future will require fundamental rethinking and reengineering of the current electricity grid. Indeed, the prospect of the smart grid poses a grand challenge for artificial intelligence, as explored in this month's cover story beginning on page 86.

Review Articles

Research Highlights

Association for Computing Machinery
Advancing Computing as a Science & Profession

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.

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.

Subscriptions
An annual subscription cost is included in ACM member dues of $99 ($40 of which is allocated to a subscription to *Communications*); for students, cost is included in $42 dues ($20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is $100.

ACM Media Advertising Policy
Communications of the ACM and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current Advertising Rates can be found by visiting http://www.acm-media.org or by contacting ACM Media Sales at (212) 626-0686.

Single Copies
Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

COMMUNICATIONS OF THE ACM

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

POSTMASTER

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

Association for Computing Machinery

Printed in the U.S.A.

DOI:10.1145/2133806.2133807

Yunhao Liu and Vincent Shen

ACM China Council

The ACM China Council was launched in June 2010 as a key component of ACM's China initiative. The goal of ACM China Council is threefold: to increase the number of high-quality ACM activities

in China, to raise ACM's visibility throughout China, and to contribute to advancing computing as a science and profession in China. In the process, we work to increase ACM membership in China.

We can report significant progress in these areas since our 2010 launch. Indeed, before the existence of the ACM China Council, most, if not all, Chinese professionals thought ACM was an American organization. In addition, Chinese membership in ACM was low—less than 1,500 total.

The first two major tasks for ACM China were educating the computing community about ACM—what it is and what it offers—and recruiting more members in China. Early on, ACM China initiated conversations with the China Computer Federation (CCF). With more than 15,000 members, CCF has a dominant position among computer professionals and students in China and is very much like ACM in terms of its mission and focus on publications, conferences, and chapters. The discussions between ACM China and Zide Du, the General Secretary of CCF, and other officials in CCF were quite productive, partly because many of the members-at-large of ACM China are also senior members of CCF.

During CCF's 2010 China National Computer Conference (CNCC) at Hang Zhou, Zhejiang, China, ACM President Alain Chesnais delivered a keynote speech to 1,500 attendees, introducing ACM, its activities worldwide, and the nature of volunteer work within the Association. His talk was well received. ACM CEO John White and CCF's Zide Du then signed a Memorandum of Understanding (MOU) between the two organizations. This MOU established a special 12-month joint membership in ACM for CCF members. In addition, CCF's monthly flagship publication, *Communications of the CCF*, gave ACM two pages per issue to publicize the Association's activities and initiatives. The MOU also called for ACM and CCF to explore joint efforts in publications, conferences, chapters, and awards.

One joint activity has focused on CCF's Young Computer Scientists & Engineers Forum (YOCSEF)—an annual series of academic activities hosted by more than 20 cities. Since early 2011, ACM China has worked with CCF to organize several YOCSEF events where Yunhao Liu (vice chair of ACM China Council) and other ACM China Council members delivered talks to help YOCSEF members learn more about ACM and the ways they can get involved in the Association's activities. As a result of these efforts, five YOCSEF chapters (in Shanghai, Beijing, Jinan, Chengdu, and Hangzhou) will become CCF-ACM chapters this year.

ACM China also invited senior computer scientists to join the CCF@U program, which is similar to ACM's Distinguished Speakers Program. The ACM China contribution to the CCF@U program involves talks to university students to help them develop in their professional careers. This activity also provides another opportunity for students to find out more about ACM. Speakers from this program visited more than 90 universities last year.

ACM co-sponsored CNCC 2011, held at Shenzhen, Guangdong Province last November. ACM Turing Award winner, Joseph Sifakis, and ACM's past president, Dame Wendy Hall, delivered keynote talks; Yunhao Liu addressed the opening ceremony. More than 2,000 computing academics, students, and professionals attended this event. At that conference, the CCF Executive Committee voted to continue the joint-membership arrangement with ACM by increasing the cost of CCF membership to include ACM membership. In late February of this year, more than 10,000 CCF members formally joined ACM.

We believe ACM China has reached the original goals set during its first meeting in 2010. In the next two years, we expect to have 10–20 chapters in China and 15,000 members. We plan to organize more activities, especially to improve communications and outreach in both academic institutions and industry. We also plan to sponsor some joint awards with CCF and to begin translating selected articles from *Communications of the ACM* into Chinese for distribution to members in China. C

Yunhao Liu (yunhaoliu@gmail.com), Vice Chair for Operations of ACM China Council, is a professor in the Department of Computer Science and Engineering at Tsinghua University, Beijing, China. **Vincent Shen** (shen@cse.ust.hk), Vice Chair for Publications and Conferences of ACM China Council, is professor emeritus in the Department of Computer Science and Engineering at Hong Kong University of Science and Technology, Kowloon, Hong Kong. To learn more about ACM China, visit china.acm.org

DOI:10.1145/2133806.2133808

The Beauty of Simplicity

As an admirer of the "artistic flare, nuanced style, and technical prowess that separates good code from great code" explored by Robert Green and Henry Ledgard in their article "Coding Guidelines: Finding the Art in the Science" (Dec. 2011), I was disappointed by the authors' emphasis on "alignment, naming, use of white space, use of context, syntax highlighting, and IDE choice." As effective as these aspects of beautiful code may be, they are at best only skin deep.

Beauty may indeed be in the eye of the beholder, but there is a more compelling beauty in the deeper semantic properties of code than layout and naming. I also include judicious use of abstraction, deftly balancing precision and generality; elegant structuring of class hierarchies, carefully trading between breadth and depth; artful ordering of parameter lists, neatly supporting common cases of partial application; and efficient reuse of library code, leveraging existing definitions with minimum effort. These are subjective characteristics, beyond the reach of objective scientific analysis—matters of taste not of fact—so represent aspects of the art rather than the science of software.

Formalizing such semantic properties is more difficult than establishing uniform coding conventions; we programmers spend our professional lifetimes honing our writing skills, not unlike novelists and journalists. Indeed, the great American essayist Ralph Waldo Emerson (1803–1882) anticipated the art in the science of software like this: "We ascribe beauty to that which is simple; which has no superfluous parts; which exactly answers its end; which stands related to all things; which is the mean of many extremes." It is to this standard I aspire.

Jeremy Gibbons, Oxford, U.K.

Along with the solid rules of programming laid out by Robert Green and Henry Ledgard (Dec. 2011), I add: Since programs are meant to be read, they should also be spell-checked, with each name parsed into separate words that are checked against a dictionary, and common shortenings and abbreviations (such as "num" for "number" and "EL" for "estimated lifetime") included in the dictionary to help standardize ways of expressing names and making programs more readable.

The exception to this spelling rule, as suggested in the article, is the locality of reference, such that index variables like "I" do not have to be spelled out when used locally within a loop. However, newer program constructs (such as `for _ each`) would mostly eliminate the need for such variables. Meanwhile, parameter names should have fuller names, so their intent could be determined by reading the header without also having to refer to the implementation.

Moreover, the code in the article's Figure 13 (an example of the kind of code covered in the article) could have been broken into multiple routines at the points each comment was inserted. This would have separated the flow from the details and made the code easier to understand. This way, each of the smaller functions would have been simpler to test, with testability a proven indicator of code quality.

Ken Pugh, Durham, NC

Still Paying for a C Mistake

In "The Most Expensive One-Byte Mistake" (Sept. 2011), Poul-Henning Kamp addressed the decision taken by "the dynamic trio" Ken Thompson, Dennis Ritchie, and Brian Kernighan when they designed C to represent strings as null-terminated rather than measure them through a preceding length count. Kamp said he found no record of such a decision and no proof it was even a conscious decision in the first place. However, he did speculate it might have been motivated by a desire to conserve memory at a time when memory was an expensive resource.

I fully agree with Kamp that the decision, conscience or not, was a mistake, and ultimately a very costly one. In my own C programming in the 1970s I found it a frequent source of frustration but believe I understand its motivation: C was designed with the PDP-11 instruction set very much in mind. The celebrated C one-liner

```
while (*s++ = *t++) ;
```

copies the string at `s` to the string at `t`. Elegant indeed! But what may have been lost in the fog of time is the fact that it compiles into a loop of just two PDP-11 instructions, where register `R0` contains the location of `s` and register `R1` contains the location of `t`:

```
A  mov  (@R0)++,(@R1)++
   bne  A  test result for nonzero and
           branch
```

Such concise code was seductive and, as I recall, mentioned often in discussions of C. A preceding length count would have required at least three instructions.

But even at this level of coding, the economy of the code for moving a string was purchased for a high price: having to search for the terminating null in order to determine the length of a string; that price was also paid when concatenating strings. The security issues resulting from potential buffer overruns could hardly have been anticipated at the time, but, even then, such computational costs were apparent.

"X-Rays will prove to be a hoax": Lord Kelvin, 1883. Even the most brilliant scientists sometimes get it wrong.

Paul W. Abrahams, Deerfield, MA

Beware This Fatal Instruction?

Regarding the article "Postmortem Debugging in Dynamic Environments" by David Pacheco (Dec. 2011), I have a question regarding the broken code example in the section on

native environments, where Pacheco said, "This simple program has a fatal flaw: while trying to clear each item in the `ptrs` array at lines 14–15, it clears an extra element before the array (where `ii = -1`)." I agree the out-of-bounds access and write on the `ptrs` array could be fatal in some cases, but wouldn't writing to an uninitialized pointer be the true cause of fatality in this case?

I am not familiar with Illumos and do not know what hardware the example was run on, but it seems like writing to an address below the `ptrs` array with the negative index would probably just write to an area of the stack not currently in use, since `ptrs`, `ii`, pushed registers, and previous stack frame all likely exist in memory at addresses above `ptrs[-1]`. However, since the stack is not initialized, `*(ptrs[ii])` will access whatever address happens to be in memory at `ptrs[ii]`, while `*(ptrs[ii]) = 0;` will try writing `0` to that address. Wouldn't such an attempt to write to a random location in memory be more likely to be fatal to the program's execution than writing to an unused location on the stack?

Berin Babcock-McConnell, Tokyo

Author's Response:

The sample program was intentionally broken. If dereferencing one of the array elements did not crash the program immediately (a stroke of luck), then the resulting memory corruption (whether to the stack or elsewhere) might have triggered a crash sometime later. In a more realistic program (where code inspection is impractical), debugging both problems would be hopeless without more information, reinforcing the thesis that rich postmortem analysis tools are the best way to understand such problems in production.

David Pacheco, San Francisco

Who Qualifies?

The idea of establishing a universal index to rank university programs, as discussed by Andrew Bernat and Eric Grimson in their Viewpoint "Doctoral Program Rankings for U.S. Computing Programs: The National Research Council Strikes Out" (Dec. 2011), was apparently first proposed more than 50 years ago in the story "The Great Gray Plague" in sci-fi magazine *Analog* (Feb. 1962, http://www.gutenberg.org/files/28118/28118-h/28118-h.htm). I hope anyone proposing such an index today would first read that story and carefully consider the implications of limiting alternative sources of research. The index algorithm would likely miss the control variables, and different measuring variables would likely be used in studies in different institutions. Over time, the index algorithm would likely focus on the institutions with the highest-ranked indexes and their particular ways of viewing research results—regrettably away from other lines of research that might otherwise yield potential breakthroughs through new theories.

Randall J. Dyck,
Lethbridge, Alberta, Canada

Insight, Not Numbers

In his blog (Sept. 2, 2011), Daniel Reed asked, "Why do we ... have this deep and abiding love of computing?" and "Why do we compute?" His answer, "We compute because we want to know and understand," echoed Richard Hamming of the old Bell Labs, who famously said, "The purpose of computing is insight, not numbers."

But a deeper understanding of our need to compute can be found in the mathematical formalism Gregory Chaitin of IBM calls Algorithmic Information Theory, or AIT. Perhaps the most important insight from AIT is that information is a conserved quantity, like energy and momentum. Therefore, the output from any computation cannot contain more information than was input in the first place. This concept shifts Reed's questions more toward: "Why do we compute, if we get no more information out than we started with?"

AIT can help answer this question through the idea of compression of information. In AIT, the information content of a bitstring is defined as the length of the shortest computer program that will produce that bitstring. A long bitstring that can be produced by a short computer program is said to be compressible. In AIT it is information in its compressed form that is the conserved quantity. Compressibility leads to another answer to Reed'squestions: "We compute because information is often most useful in its decompressed form, and decompression requires computation." Likewise, nobody would read a novel in its compressed .zip format, nor would they use the (compressed) Peano axioms for arithmetic to make change in a grocery store.

Further, AIT also provides novel insight into the entire philosophy of science, into what Reed called our "insatiable desire to know and understand." AIT can, for the first time, make the philosophy of science quantitative. Rather than ask classical questions like "What do we know and how do we know it?," AIT lets us frame quantitative questions like "How much do we know?," "How much can we know?," and "How much do we need to know?"

Unlike most questions in philosophy, these questions have concrete, quantitative answers that provide insight into the nature of science itself. For example, Kurt Gödel's celebrated incompleteness theorem can be seen as a straightforward consequence of conservation of information. AIT provides a simple three-page proof of Gödel's theorem Chaitin calls "almost obvious." And one of the quantitative implications of Gödel's theorem is that a "Theory of Everything" for mathematics cannot be created with any finite quantity of information. Every mathematical system based on a finite set of axioms (a finite quantity of compressed information) must therefore be incomplete. This lack of completeness in mathematics leads naturally to another important quantitative question "Can a Theory of Everything for physics be created with a finite quantity of information?" that can also be explored using the concepts developed in AIT.

Douglas S. Robertson, Boulder, CO

For more on AIT, a quantitative philosophy of science, and the question of why we compute, see http://cires.colorado.edu/~doug/philosophy/

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

BLOG@CACM

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

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

DOI:10.1145/2133806.2133809 http://cacm.acm.org/blogs/blog-cacm

The Power of Computing; Design Guidelines in CS Education

Daniel Reed writes about how computing systems increase human intellect and abilities. Mark Guzdial discusses the need to avoid polarized and extreme positions in education and the trend toward design-based research.

Daniel Reed
"Intellectual Amplification via Computing"
http://cacm.acm.org/blogs/blog-cacm/106540
February 18, 2011

Many years ago, Fred Brooks relayed a tale about how he chose the first application target domain for his computer graphics research. It was not long after he had left IBM and completed his work on the IBM System/360. He had just moved to Chapel Hill and taken a faculty position at the University of North Carolina.

As I remember Fred telling the story, with a bit of a twinkle in his eye, he went to see one of the senior university administrators. He told the administrator that, as a computer scientist, he was in the intelligence amplification business. Who on the campus, Fred wanted to know, might most benefit from having their intelligence amplified?

I have recalled this story many times, always with a smile, as I have reflected on the nature of computing and its power.

DANIEL REED

"Success accrues to the talented with access to the most effective and powerful tools."

Amplification and Universality

Computing systems share many features with other instruments and machines in amplifying human abilities. However, one aspect distinguishes them—namely, their general utility as an intellectual amplifier. Like a universal Turing machine, which can simulate any other Turing machine with arbitrary inputs, computing is broadly—dare I say universally—applicable to human intellectual endeavors, much as all the variants of the inclined plane and lever are applicable to human physical endeavors.

Sir Humphrey Davy could well have been speaking about computing two centuries ago when he said, "Nothing tends so much to the advancement of knowledge as the application of a new instrument. The native intellectual powers of men in different times are not so much the causes of the different success of their labors, as the peculiar nature of the means and artificial resources in their possession."

In a phrase: Success accrues to the talented with access to the most effective and powerful tools.

Supercomputing and its applications to science and engineering have been canonical examples of this universal benefit. Powerful new telescopes advance astronomy, but not materials science. Powerful new particle accelerators advance high-energy physics, but not genetics. In contrast, supercomputing advances all of science and engineering because all disciplines benefit from high-resolution model predictions and theoretical validations.

As exciting as those opportunities remain, new ones are emerging in the world of big data.

The tsunami of structured scientific data produced by a new generation of sensors and the growth of semi-structured and unstructured data from business, entertainment, social networks, and popular culture have created new needs for the creative application of our intellectual amplifier. As the performance of IBM's Watson system on the game show "Jeopardy!" illustrated, the combination of large-scale data, rich algorithm suites, and powerful computing are opening new vistas. Vannevar Bush's 1940s vision of a Memex, a device capable of storing, indexing, and retrieving data from a broad knowledge base, is now within our reach.

It really is about how we use computing as an intellectual amplifier, allowing humans to be more productive and more creative by doing what we do best—asking interesting questions, ones that span multiple disciplines and that illuminate opportunities at their interstices—aided by powerful analytic and computation engines.

Mark Guzdial
"From 'Must' and 'Unsuitable' to Design Guidelines in Computing Education"
http://cacm.acm.org/blogs/blog-cacm/106838
March 25, 2011

Matthias Felleisen gave a rousing opening keynote at SIGCSE 2011. There was a lot to like about the ideas and insights he presented. I particularly liked the design-oriented topic list for an introductory CS course, versus a language-oriented one. The latter one looks like just about any intro course you've ever seen, which makes his point. What I didn't like about his talk was the tone of the rhetoric. "Lesson 1: Your PL/IDE must support an arithmetic of images." "Must"? I loved what they are doing with images in Racket, but there are other tools for novices that do lead to success but don't support an arithmetic of images. A useful design guideline might be: "Lesson 1: A PL/IDE that supports an arithmetic of image engages and motivates students." I buy that. I also buy that there are lots of ways to engage and motivate students without an arithmetic of images.

Recently, I visited Carnegie Mellon University where I heard about their new introductory curriculum. They introduce computing with an imperative first course using a language called c-0 ("C-nought"), then follow that with a second course using functional programming ML. The courses are quite rigorous in that they place a strong emphasis on verification and proof as part of program development. I love the multilingual, multiparadigmatic model! Object-oriented methods show up in a new course, not part of the core curriculum. Then someone sent me a link to Robert Harper's Existential Type blog, where he claims in the "Teaching FP to freshman" post, "Object-oriented programming is eliminated entirely from the introductory curriculum, because it is both anti-modular and anti-parallel by its very nature, and hence unsuitable for a modern CS curriculum." Really? "*Unsuitable?*" The CMU report, *Introductory Computer Science Education at Carnegie Mellon University: A Deans' Perspective*, describing the rationale for the new approach is more careful in its claims:

Although object-oriented programming (in its myriad forms) remains a dominant theme in industrial software development, the use of object-oriented languages, such as Java, at the introductory level introduces considerable complexity and distracts from the core goals at the introductory level. It seems preferable to give fuller coverage of OO design and implementation methodology to later in the curriculum to allow more focused concentration on basics at the introductory level.

I can buy that. That's a valid criticism, and it does consider where objects do fit into the curriculum. The Harper blog post paints all of OO with a Java brush. Alan Kay recently said in a comment to a post in my Computing Education Blog, "By my original definition of 'Object oriented,' neither Java nor C++ is OO." Message sending in Smalltalk and Self is both object-oriented and easily made parallel. Because Java doesn't work for the purpose, *all* of object-oriented programming is "unsuitable" and must be "eliminated"?

MARK GUZDIAL

"Let's celebrate our successes in computing education without claiming that ours is the only path to that success."

Why do we take such polarized and extreme positions in education? All across the political spectrum, there are complaints about polarization in education. Maybe we fight about it because it is so important. But I would hope the researchers and scientists could be more careful. Could you get a paper published in a programming languages or parallel algorithms conference making statements about "must" and "unsuitable" without proof or evidence?

In education research, there is a trend toward design-based research that I think helps to avoid the polarities. Design-based research is about doing iterative development in real classrooms. You rarely come out with definitive claims supported by statistical significance that researchers expect to generalize. Instead, you end up with statements like, "Under these conditions, we can show that these interventions lead to significant learning gains." Those kinds of statements can guide future design, and can help the teacher, but avoid defining A One, True Way. Rather, we can talk about what works, leaving open the door that some other set of conditions might make another set of interventions successful. We absolutely care about having strong and careful yardsticks, so that we can measure real learning. But we recognize that there are many ways to reach the end of that yardstick.

Human beings are complicated and messy. What works great with one set of students may not work at all with another. Laws like "F=ma" are rare in education. Instead, we need guidelines that inform future efforts. Let's celebrate our successes in computing education without claiming that ours is the only path to that success. C

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

ACM Member News

JUDEA PEARL RECEIVES HARVEY PRIZE

Judea Pearl, professor emeritus of computer science at the University of California, Los Angeles, has received the 2011 Harvey Prize for developing Bayesian networks and the mathematics for calculating cause and effect. The prize, which is presented by the Technion-Israel Institute of Technology, includes an award of $75,000.

Bayesian networks provide a way of structuring data and examining the complex relationships among thousands of variables, and deal with the noise inherent in the data. They allow the calculation of probability distributions that would otherwise be too large to handle, and have been used to understand computer vision, genetics, and relationships in networks. "The key contribution I made is the ability to represent knowledge under uncertainty, so the computer can cope with noisy observations," says Pearl. The main users of this work include "epidemiologists, social scientists, and people who are dealing with data," he says.

Pearl calls Bayesian networks a passive approach, the statistics of observation and extrapolation. The study of causality, on the other hand, examines how actions relate to results. One ultimate goal, Pearl says, is to allow robots to understand causal relationships in the environment so they can cope with the unexpected. For instance, if an industrial process involving several steps suddenly starts producing the wrong results, the robot could work through the steps to diagnose and fix the problem, much the way a human does.

Telling a robot the consequences of one action would be good and another would be bad, and thus helping it to decide which actions to take, could be a powerful way to communicate, Pearl says. "That's how we talk to each other, and that's how we ought to talk to a sophisticated robot."

—Neil Savage

Science | DOI:10.1145/2133806.2133811 Gregory Goth

Preserving Digital Data

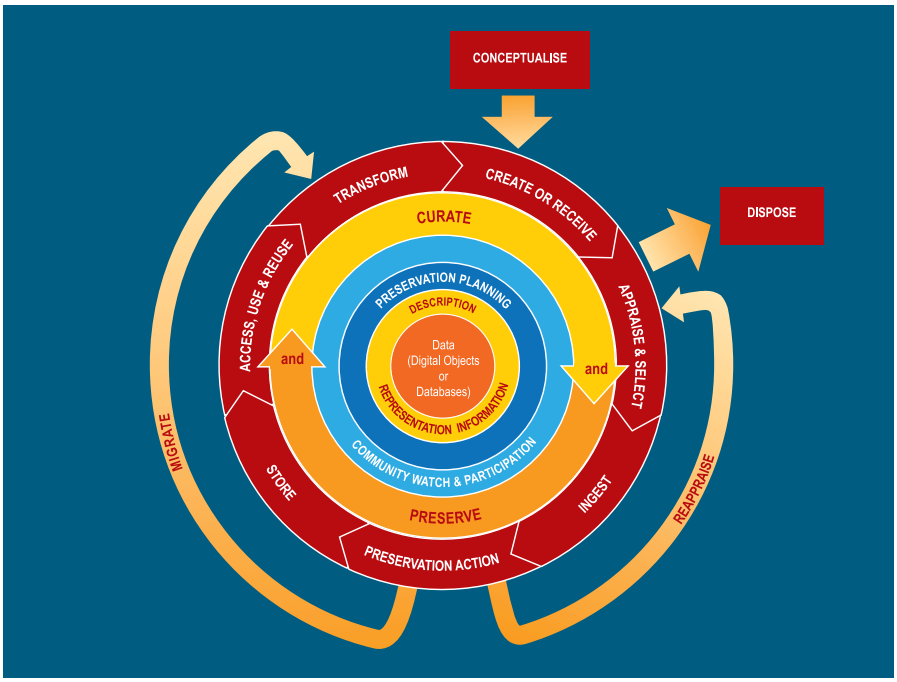
Scientific data is expanding at an unprecedented rate. While new tools are helping preserve this data, funding must be increased and policy coordination needs improvement.

DATA PRESERVATION EXPERT Jeff Rothenberg says digital data has a behavior problem. Rothenberg does not mean digital artifacts are purposely causing mayhem across platforms and applications, but rather the questions of how to make these artifacts useful across space and time are running into perception issues.

For centuries, Rothenberg says, archivists could properly do their jobs by preserving static artifacts such as printed pages and photographs for even the most complex undertakings of a society, but that has irrevocably changed.

On the far end of the spectrum from such "archaic" forms as paper, stone, and wood "is what I call inherently digital artifacts," says Rothenberg, "something that really requires a computer to even render it, and increasingly we are producing things like that. Ph.D. theses, particularly in the sciences, are becoming inherently digital, including not just text but models, which show behavior and are part of the result of the thesis. The real crux of preserving this data is how to preserve the behavioral aspects of artifacts, which generates the image in the first place."

Agencies, researchers, and technologists are beginning to seriously address this crux, but the sheer number of agencies and efforts tackling it also seem to highlight the enormity of the task.



In the U.K., the Digital Curation Center's Curation Lifecycle Model provides a graphical, high-level overview of the necessary stages for the curation and preservation of data.

FIGURE COURTESY OF THE DIGITAL CURATION CENTRE/UNIVERSITY OF EDINBURGH

Data Discovery and Preservation

Perhaps the foremost irony facing those dealing with scientific data preservation is the entrenched cultural emphasis on the next discovery and a less appreciative attitude toward old data.

"Scientists are living in a world that is spinning around very fast, writing papers, publishing results, and going on to the next project, and data is not preserved," says Stefan Winkler-

Nees, program officer in the Department of Scientific Library Services and Information Systems at the German Research Foundation (DFG), the country's largest funder of research. "The big 'A-ha!' effect in Germany was that somebody realized we are losing up to 90% to 95% of the data produced in public science over time because it is not accessible anymore."

Winkler-Nees's observations are echoed by others, including Rothenberg, Gary King, director of the Institute for Quantitative Social Science at Harvard University, and Sylvia Spengler, program director in the U.S. National Science Foundation's Computer Science and Engineering directorate.

The biggest challenge facing data preservationists, King says, is "the infrastructure to make it all happen. It's straightforward to get grants to do research. The agencies that could provide the funding tend to provide short-term funding—primarily for research, sometimes to build infrastructure, but virtually never for preservation. Preservation is the promise of keeping things in perpetuity. That's a long time. Figuring out that is really hard."

"The kinds of domains which do think about long-lived collections have an inevitable tension between the acquisition of the data and the preservation of the data," Spengler says. "And the funds for that, at least currently, are all within one budget. So it has to be an active decision within a community how to deal with the preservation issues for that community."

However, these existing cultural models are being challenged somewhat by more comprehensive national initiatives. In Germany, the DFG recently funded 28 projects with €9.9 million under the aegis of data accessibility, including preservation; in the U.S., the National Science Foundation published requirements effective January 2011 that every funded project must include a two-page data management plan, including retention periods, on a directorate-specific basis; and in the U.K., JISC (originally the Joint Information Services Committee) has long coordinated data management and preservation policies among the nation's funders and research institutions.

Recent research into the efficacy of archived data also indicates a substantial return on investment. Researchers receiving support from the Dryad data repository project estimated the repository could ingest and curate 10,000 publications annually for about U.S. $400,000—and that "a $400,000 investment would contribute to more than 1,000 papers within four years, far greater than the accepted value of a research dollar."

Yet there is little to no national level coordination of data preservation standards, even in the U.K.'s well-documented research council guidelines. The U.K.'s Digital Curation Center, for example, links to each of the country's nine primary research funders' policies; some include very amorphous wording on data preservation while others stipulate a 10-year retention period.

Simon Hodson, program manager for digital infrastructure at JISC's Managing Research Data program, says funders are well aware of the potential for retaining and repurposing data in future experiments, but that there must be some discipline-specific autonomy in setting policy.

"That's going to have to go forward on a case-by-case basis," Hodson says. "You can't keep everything, and I think researchers and institutions have to be in a position where they can make these decisions on what they retain and what they throw away."

Rothenberg says a deep cultural divide between scientists and archivists must be addressed to forge a consensus on how this data should be preserved and curated.

"Scientific data archives are not archives in the same sense as a national archive," he says. "People may have been trained in archival studies, but they don't have the same mind set [as scientists]. Each of those communities has, to some extent, its own history and philosophy and vision. Those may be realigning under the mandate of digital convergence, but it's a slow process."

Gaining Momentum

Fitful as progress may be, the global effort at addressing data preservation is gathering momentum. The International Organization for Standardization (ISO) is actively pursuing a Digital Preservation Interoperability Framework specification; the JISC-sponsored *Keeping Research Data Safe* report, published in two parts in 2008 and 2010 respectively, features numerous best practice recommendations; Rothenberg and RAND Europe researcher Stijn Hoorens in 2010 co-authored a comprehensive report, which was sponsored by the British Library, that explored the library's possible role in preserving scientific, technical, and medical data; and the White House Of-

Technology

Africa Increases Its Internet Usage

Africa's Internet usage has grown dramatically during the last decade due to new information and communication technologies, including improved fiber-optic networks and increased availability of computers and mobile phones.

Africa had nearly 140 million Internet users by the end of 2011, compared to only 4.5 million users at the end of 2000, according to Internet World Stats. The continent's Internet penetration is 13.5%, compared to 26.2% in Asia, 35.6% in the Middle East, 39.5% in Latin America and the Caribbean, 61.3% in Europe, and 78.6% in North America.

The African nations with most Internet users are Nigeria (45 million), Egypt (21.7 million), Morocco (15.7 million), Kenya (10.5 million), and South Africa (6.8 million). Nigeria's 29% Internet penetration is slightly below the global average of 32.7%.

After Google, Facebook is the most popular Web site in Africa. Egypt boasts the largest number of Facebook users (9.4 million), followed by South Africa (4.8 million), Nigeria (4.4 million), Morocco (4.1 million), and Kenya (1.3 million). Africa's Facebook penetration is 3.6%, compared to 4.7% in Asia, 8.4% in the Middle East, 25.5% in Latin America, 27.4% in Europe, and 50.3% in North America. Egypt's 11.4% Facebook penetration is almost equal to the global average of 11.5%.

—Jack Rosenberger

fice of Science and Technology Policy released a Request for Information, soliciting comments regarding public access to digital data resulting from federally funded research in November 2011.

Also, some of the research funded under NSF's Sustainable Digital Data Preservation and Access Network Partners (DataNet) program is beginning to bear significant fruit. Although Spengler says the DataNet projects were and are intended to be exemplars and fairly restrictive prototypes due to limited funding, her NSF colleague Rob Pennington says DataNet awardees are working with other researchers eager to find ways to share data across domains and disciplines.

One standout example of this is iRODS (integrated Rule-Oriented Data System), developed by the Data Intensive Cyber Environments (DICE) research group at the University of North Carolina (UNC) and the University of California, San Diego. Institutions spanning disciplines from climatology to social sciences are adopting the innovative data grid tool. The NSF awarded iRODS developers $8 million in September 2011 to build a policy-driven national data management infrastructure, motivated by the discrete data management requirements of the NSF's Ocean Observatories Initiative, NSF's Consortium of Universities for Advancement of Hydrologic Science, engineering projects in education, CAD/CAM/CAE archives, the genomic databases of the iPlant collaborative, the H.W. Odum Institute for Research in Social Science at UNC, and NSF's Science of Learning Centers.

iRODS has also been adopted by scientific data centers worldwide, including astronomical observatories in Canada and France, climate centers in the U.S., and at the Sanger Institute genomics databases in the U.K. It is also in use in the U.S. National Archives and Records Administration's Transcontinental Persistent Archives Prototype.

iRODS is the successor to the pioneering Storage Resource Broker (SRB) architecture. DICE Director Reagan Moore says iRODS's rule engine-based architecture makes the distributed management of a data grid much simpler than the hard-coded SRB architecture, can serve as the sort of reporting tool that demonstrates researchers are meeting their mandated data management plan outlines—and is also the sort of policy-based system that melds the concepts of data management and data preservation for whatever duration is required.

There is little to no national-level coordination of data preservation standards, even in the U.K.'s well-documented research council guidelines.

"We make the assertion that any data-management application really consists of the policies you're applying in order to validate assertions about what you've done," Moore says. "So if I build a preservation environment, my policies are related to authenticity, integrity, chain of custody, and original arrangement."

Moore says the rule-based iRODS architecture makes it possible for users to tailor which policies apply to a given action without having to rewrite any code, whereas server-side commands in SRB were hard coded. These rules are applied in a platform-agnostic manner through any number of 254 microservices selected as pertinent by any user community.

At least one project has already successfully replicated a recognized sample archive, the Harvard IQSS-developed Dataverse, using iRODS. Researchers at UNC's Odum Institute performed a Dataverse-to-iRODS transfer using the Open Archives Initiative Protocol for Metadata Harvesting and the compatible Data Documentation Initiative standard, plus XML.

"The result is an accurate copy of a Dataverse archive inside iRODS," according to the UNC authors, "which data grid administrators can preserve over the long term by, for example, replicating the information to many geographically distributed storage resources."

Phil Butcher, head of information technology at the Sanger Institute, says the organization's iRODS installation has run smoothly. He believes funding agencies should make themselves aware of the details of such groundbreaking technologies, even if comprehensive national and international management and preservation policies are not possible.

"Unless you start to deploy tools like this, you'll be in trouble," Butcher says. "The funding bodies in particular have a real opportunity to understand a bit more about the technology. If there is a list of two or three tools people could use, the best would come to the surface. We're getting to the point where some of these decisions have to be made. Otherwise a lot of groups will be in a lot of trouble." C

Further Reading

Beagrie, N., Lavoie, B., and Woollard, M.
***Keeping Research Data Safe 2*, JISC, Bristol, U.K., April 2010..**

Chiang, G., Clapham, P., Qi, G., Sale, K., and Coates, G.
Implementing a genomic data management system using iRODS in the Wellcome Trust Sanger Institute, *BMC Bioinformatics 12*, 361, Sept. 9, 2011.

Neuroth, H., Strathmann, S., and Vlaeminck, S.
Digital preservation needs of scientific communities: The example of Göttingen University, *Proceedings of the 12th European Conference on Research and Advanced Technology for Digital Libraries*, Aarhus, Denmark, Sept. 14–19, 2008.

Rothenberg, J. and Hoorens, S.
***Enabling Long-term Access to Scientific, Technical and Medical Data Collections*, RAND Europe, Cambridge, U.K., 2010.**

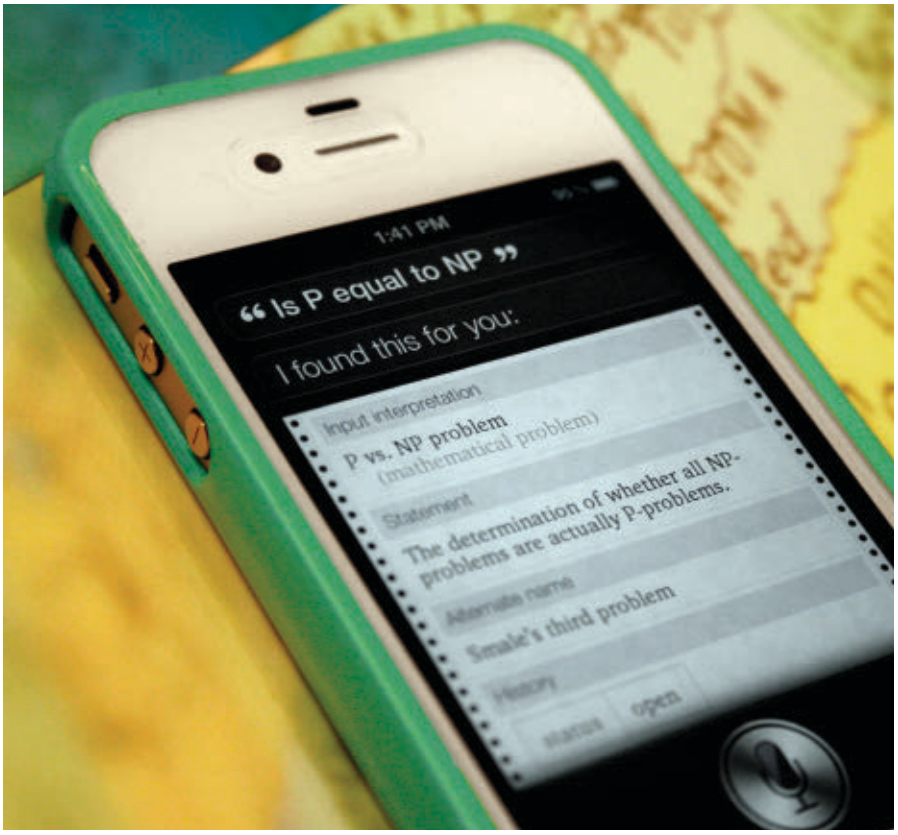
Ward, J.H., de Torcy, A., Chua, M., and Crabtree, J.
Extracting and ingesting DDI metadata and digital objects from a data archive into the iRODS extension of the NARA TPAP using the OAI-PMH, 5th IEEE International Conference on e-Science, Oxford, UK, Dec. 9–11, 2009.

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

Technology | DOI:10.1145/2133806.2133812 Tom Geller

Talking to Machines

Voice recognition programs like Siri are now capable of understanding spoken commands, recognizing a conversation's context, and answering questions in a personable manner.



Siri can answer a wide variety of spoken questions in a conversational manner even in difficult conditions but, like its human inventors, it has yet to solve to the *P* versus *NP* problem.

When Apple integrated Siri into the iOS operating system last October, it spurred iPhone owners to start talking to their phones as well as through them. The program, which converts spoken commands such as "Schedule dinner with Lisa at 6 tonight" into calendar appointments, Web searches, and the like, is the most widely distributed example of a cognitive assistant to date. More than four million iPhone 4S's featuring Siri were sold during its first weekend. Although users might see it as simple speech recognition, its abilities go far beyond simple transcription.

Siri represents an important moment when voice recognition, information management, artificial intelligence, task fulfillment, and user interface marry in a way the general public finds usable and productive. As Wolfram Research executive director Luc Barthelet says, "The news about Siri is that it works. People have tried to get computers to answer questions conversationally for at least 15 years, but only now has the technology reached a threshold where people overall like it." The iPhone's popularity also gives intelligent software assistants wider exposure than they would get otherwise. Roger K. Moore, editor in chief of the journal *Computer Speech and Language*, points out that "the field of research hasn't changed dramatically. What's new is that Siri's brought several complementary technologies together. Our business has been going for many years. Only now, with Siri, everybody knows about it."

Executing Commands

There is a long road between the spoken command and its fulfillment, though. The first step in the process is to convert the audio of speech into meaning. The two main applications of speech recognition—dictation and command recognition—have forced researchers to pursue parallel methods that balance vocabulary, accent, and context needs.

Grammar-based voice recognition is optimized for situations where the program has a very good idea of what the speaker will say. Its most common application is in Interactive Voice Response (IVR) systems, such as those that some airlines use to interpret spoken reservations and requests for information. These are often conversational. A recorded voice asks the speaker a question, then listens for the response. As a result, the system needs to understand only a limited vocabulary. But according to Dan Faulkner, vice president of product and strategy for the Enterprise Business Unit at Nuance, responses can vary widely no matter how restricted the domain is. "When a phone prompt system tells you to 'please say yes or no,' we might be pretty confident the speaker will say 'yes' or 'no.' But 'yes' could be 'yeah,' 'that's correct,' or 'yup,' and in the Southern states some people will say 'yes, ma'am' and 'no, ma'am.' So even for something as simple as 'yes' or 'no,' you need quite an extensive list of phrases."

By contrast, a language-based voice recognition system is optimized for dictation. It makes few assumptions about context, attempting to recognize and transcribe every word it hears. In

PHOTOGRAPH BY ALICIA KUBISTA

gaining a wide vocabulary, it gives up the ability to understand a variety of accents. According to Peter Mahoney, general manager of the division of Nuance that produces its Dragon Naturally Speaking products, modern dictation programs no longer need the 30–60 minute initial training period that earlier versions did. "People now get well above 90% accuracy the first time they use [our dictation programs]; that figure used to be only 75% to 80%."

Although initial training is not as necessary as before, dictation programs still train themselves as you use them. "The program looks at three things to get to know its user," says Mahoney. "First is acoustics. How do you actually say words? That's what the initial training used to focus on. Second is the kind of words you use—your spoken writing style. Third is a variety of user preferences, such as how you say numbers and names, and how you like them to be formatted. So if you were to say, 'I gave you two dollars and forty-two cents,' the program knows to transcribe that as 'I gave you $2.42.'"

These two methods of understanding speech are starting to converge, however. Dictation programs adapt somewhat to the tasks they are performing, for example favoring the phrase "dot com" when the cursor is in an email program's "To" field. And Siri switches from command to dictation mode when appropriate.

The two main applications of speech recognition—dictation and command recognition—have forced researchers to pursue parallel methods that balance vocabulary, accent, and context needs.

Learning and Organizing

But voice recognition is only a small part of the puzzle. Before a cognitive assistant can schedule that dinner with Lisa, it needs to understand that dinner is an event of limited duration, that Lisa is a person whose contact information is found in the device's address book, and so forth.

Some of this understanding came from a Defense Advanced Research Projects Agency (DARPA)-funded project, Cognitive Assistant that Learns and Organizes (CALO), which was part of DARPA's PAL (Personalized Assistant that Learns) program. CALO's focus was not on voice recognition per se, or natural-language understanding, or on human-computer interaction in general. Rather, it was about making computer systems learn in everyday settings, such as learning to recognize concepts. It then had to relate them through an underlying ontology, and trigger desktop applications and Web services.

SRI International principal investigator C. Raymond Perrault describes the challenge of solving for ambiguities in natural language. "Movie titles are typically just phrases in the language," Perrault points out, "so you could say 'Get me two tickets to *The Artist*,' and the system would recognize that phrase as a movie title. On CALO we tried to solve for such ambiguities robustly, and would make it easy to build systems that do as well." Humans resolve these ambiguities using context, while teaching a computer to learn them requires building very long lists of such things as addresses, people, movies, and organizations, along with a solid categorization system to manage them. Ideas from CALO suggested a new approach to the development of a spoken interface to a set of Web services, eventually realized in Siri, which was built by a company that spun out of SRI International.

Neither a robust ontology nor high-speed voice recognition are possible without substantial data as input. Here, the Web's social nature—and some carefully worded clauses in end-user license agreements—allow software assistants to collect acoustic, syntactic, and factual information. "It wouldn't be possible to do something like Siri 10 to 15 years ago because you couldn't get enough data to train the system," says Alan W. Black, associate professor at the Language Technologies Institute of Carnegie Mellon University. "Google started collecting data years ago through its free 411-GOOG informational service. Notably, they advertised on billboards rather than online. What they were actually doing was finding out how ordinary people asked questions." By contrast, Nuance's Faulkner recalls how the company trained its dictation products to understand noisy phone transmissions in past years. "We'd pay people to come into the office and give them scripts. We'd give them a mobile phone, put them in a cab, tell them to call a number, then record their speech."

The resulting collection of data is much too big to fit on today's portable devices, so command agents rely on two other recent developments: ubiquitous high-speed bandwidth and cloud storage. For speed, recognition actually takes place on the server rather than in the device itself. But because the voice channel has a relatively low 8kHz resolution, resulting in noise and distortion that could affect recognizability, the sound is transmitted via the data channel. "We've now reached a point that we put a fairly fast stream of language over an iPhone at 16kHz," says Faulkner. "Being able to capture twice as much data makes a big difference."

On the back end, such programs rely on third-party services. Siri counts among its providers Yelp, OpenTable, StubHub, Rotten Tomatoes, and *The New York Times*. One of its most unusual providers is Wolfram|Alpha, which catalogs general concepts and facts rather than the raw Web site data stored by such search engines as Google. As Wolfram's Barthelet describes, "The Web pushes the world back on you. You're asking, 'What does the world know about this subject?' But often, you just want to know the answer." As with conversational voice recognition systems, Wolfram|Alpha attempts to deliver that answer by first limiting the question's domain based on the questioner's location, previous questions, and other factors.

Measures of Success

Ultimately, these steps serve the single goal of delivering a relevant, true, and useful response with acceptable speed. But as Carnegie Mellon's Black points out, a software agent's job is not done if it only delivers facts. "One standard measure of spoken dialogue systems is task completion," he notes. "Did the user successfully get the weather? But it's clear that that's not the only goal. You can have an interaction that's successful and takes little time, but is unpleasant. So satisfaction is another goal."

Black believes Siri delivers that satisfaction partly through its helpful-yet-sassy tone. "It doesn't just answer questions," he says. "It has a character. It wants to name you, to know who you are. You can tell it to call you "Master" or "Darth Vader" or whatever, but it wants to call you that. It makes things a little more personal, and that's important." Faulkner also points to Siri's many handcrafted, hidden Easter eggs. For instance, if you tell Siri "I'm drunk," it offers to call you a cab.

More importantly, today's software agents have taken context to a level never before attempted, striving to know more about you than you know about yourself—your situation, tastes, and patterns—before running off to find exactly what it believes you want. They paradoxically expand your power by limiting its domain, collapsing infinite possibilities into a single action.

"It wouldn't be possible to do something like Siri 10 to 15 years ago because you couldn't get enough data to train the system," says Alan W. Black.

Still, Moore believes the game is far from over. "The history of this field has always been one of waves of success, followed by going into the doldrums. The joke about our success is, 'Just keep showing the same graph of escalating future returns, but don't put any dates on it.' " He has tested that theory empirically by surveying his colleagues every six years, asking them when voice recognition will hit certain milestones. But every time he resurveys them, "all the dates have moved out another six years! So the future isn't getting any closer."

"Something like Siri appears and people think, 'We've solved it!' But you can't use it in a pub or a train station," says Moore. "Then when you point out all the realities of the fantastic abilities that human beings have at holding conversations in difficult circumstances, you realize we still need to solve artificial intelligence, language, neurocomputing, and so on before we have a truly autonomous agent." C

Further Reading

Baker, J., et al.
Research developments and directions in speech recognition and understanding, part 1, *IEEE Signal Processing Magazine 26*, 3, May 2009.

Baker, J., et al.
Updated MINDS report on speech recognition and understanding, part 2, *IEEE Signal Processing Magazine 26*, 4, July 2009.

Lecouteuxa, B., Linarèsb, G., and Ogerb, S.
Integrating imperfect transcripts into speech recognition systems for building high-quality corpora, *Computer Speech & Language 26*, 2, April 2012.

Moore, R.K.
Progress and prospects for speech technology: Results from three sexennial surveys, INTERSPEECH 2011, Florence, Italy, August 27–31, 2011.

Prasad, R., et al.
BBN TransTalk: Robust multilingual two-way speech-to-speech translation for mobile platforms, *Computer Speech and Language*, Nov. 15, 2011.

Yorke-Smith, S. and Myers, M.
Like an intuitive and courteous butler: A proactive personal agent for task management, *Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS 2009, Budapest, Hungry, May 10-15, 2009.

Tom Geller is an Oberlin, OH-based science, technology, and business writer.

Artificial Intelligence

Computer Passes IQ Tests, Say Researchers

Computers can mimic human intelligence, at least when it comes to scoring on standard IQ tests, if they are programmed to think more like humans, according to researchers in Sweden.

In many IQ test questions, takers are asked to identify the next number or image in a sequence. Mathematical software such as Mathematica and Maple were fed the number sequence problems and got scores below 100, which is considered average intelligence in a human. But Claes Strannegard, a researcher in philosophy, linguistics, and theory of science at the University of Gothenburg, says a program he and his colleagues developed scores 150 on number sequences questions, and 100 on progressive matrix problems, which ask a test taker to select the correct image.

The key to answering the questions lies in identifying a pattern, but in many cases there are several patterns, some very complex, that could fit. Strannegard says his program homes in on those patterns people would notice. "Some patterns are out of reach for humans and those we never consider. We use a cognitive model of a human problem solver and if a certain pattern is too complex for that model, then we reject the pattern," he explains.

The software applies Occam's Razor, giving preference to the more succinct patterns. It also gives greater weight to more general patterns. "The point is to speed up the search for solutions to AI problems by considering only those solution candidates that are accessible to the human mind, with its limitations on the cognitive resources," Strannegard says.

His hope is to develop artificial intelligence programs that are intelligent by the same standards applied to humans. Such "anthropomorphic computing" could be used for tutoring systems, verification tools for electronic systems, and as intelligent agents in games.

—Neil Savage

Society | DOI:10.1145/2133806.2133813 Leah Hoffmann

Open for Business

Should academic articles be available for free on the Web?

The number of open access publications has steadily increased during the last 10 years, even for smaller, specialized journals.

In 2001, 40 members of the editorial board of *Machine Learning*—then published by Kluwer Academic Publishers, now part of Springer—tendered their resignation. The reason behind the mass resignation was Kluwer's policy of restricting online access to *Machine Learning* articles to only subscribers. "Our resignation... reflects our belief that journals should principally serve the needs of the intellectual community, in particular by providing the immediate and universal access to journal articles that modern technology supports, and doing so at a cost that excludes no one," the members wrote in an open letter. Instead, they decided to found the *Journal of Machine Learning Research* (*JMLR*), a peer-reviewed, open access publication whose articles are freely available at the journal's Web site.

"Open access is a model that makes sense," says Lawrence K. Saul, *JMLR*'s editor-in-chief and a professor of computer science and engineering at University of California, San Diego. "You want your research to be read by as many people as possible. You don't want it to be gated by artificial barriers."

The sentiment is common among today's computer scientists. In a paper world, goes the logic, the need of the scientific community to see its research circulate widely was aligned with the business model of commercial publishers, which had a financial incentive to ensure broad distribution. In the Internet age, of course, distribution is as easy as connecting to the Internet. Many researchers have thus grown reluctant to entrust their papers to journals whose online archives are restricted to those who can pay for them.

A variety of issues are bound up in the discussion, from peer review and proofreading to research and the scholarly record. At the core, for many scientists, is copyright, and the conviction that it does not make sense to transfer ownership of their work to organizations that profit from distributing it, especially when the work was given freely, and possibly even based on taxpayer-funded research. Also, the annual profit margins at publishers like Elsevier are more than 35%. The cost of the site licenses that grant access to scholarly journals continues to increase, accounting for an estimated 50%–65% of library budgets. Computer science journals are inexpensive compared to those of other scientific disciplines, but their cost is rising nonetheless. According to a *Library Journal* study, the average cost of an annual subscription in computer science was $1,593 in 2011, up from $1,472 in 2009.

And for many computer scientists, the days when publishers were thought to perform a valuable service are largely gone. "Computer scientists can format their own papers, and there is no need for so-called professional typesetting—we have the tools to do it ourselves," says Yann LeCun, Silver Professor of Computer Science and Neural Science at New York University. Though production delays have fallen, the allure of free, instantaneous online publication is undeniable for fast-evolving scientific disciplines.

The number of open access journals has exploded during the past decade. At press time, the Directory of Open Access Journals (DOAJ) listed more than 7,300 titles, up from 2,750 in 2008 and 560 in 2004. Of them, 317 are computer science titles. The open access model has obviously proven itself for smaller journals like *JMLR*, which continues to thrive more than a decade after it was founded. Nonetheless, questions remain about the future of open access. Is it sustainable in the long run? Will a standard business model emerge? Is there room for both open and paid access publications?

Matters of Money

Nearly all academic journals rely heavily on volunteer labor, from editorial board members to authors and reviewers. Yet there are a variety of additional expenses, from managing the peer review process and maintaining digital infrastructure to archiving old articles,

creating reference links, proofreading, and providing metadata to services like Google Scholar. Commercial and learned society publishers typically incorporate these costs into the price of their subscriptions. At many open access journals, these costs are borne by volunteers, and offset by direct or indirect institutional subsidies. *JMLR*, for instance, is hosted by the Massachusetts Institute of Technology, and the journal's founder and first editor-in-chief, Leslie Pack Kaelbling, personally paid for its domain registration and operating costs. The journal now covers such expenses through two grants from Google and Microsoft. Some open access journals receive support from their national governments. Others charge authors publication fees to offset their administrative costs. These fees vary from journal to journal, but typically range from $1,500 to $5,000 per accepted article. The practice is more common in some fields than in others, though it is not, on the whole, as widespread as it is often thought to be. In 2009, Stuart Shieber, the James O. Welch, Jr. and Virginia B. Welch Professor of Computer Science at Harvard University, found that only 23%–30% of open access journals listed in the DOAJ charge publication fees. On the other hand, as the University of Quebec at Montreal psychology professor Stevan Harnad points out, the majority of top open access journals—those with an impact factor greater than one, like *PLoS Biology*—do charge publication fees.

How open access will evolve in the future is unclear. In the meantime, many traditional publishers have adopted a similar scheme. Under this hybrid model, authors can pay a fee to release their articles from the constraints of site licensing terms. Since subscription prices usually are not affected, the practice remains controversial (it is known pejoratively as double dipping), although up to 40% of authors participate in some cases. "It can be a very successful commercial model, particularly with high-volume operations," says Bernard Rous, director of publications at ACM. "It also introduces a significant bias toward heavily funded research areas and rich institutions, where there tends to be a lot of grant money. Without institutional underwriting, you're left with an unfair burden on individuals that have to pay." For that reason, ACM decided against the idea during a recent review of its policies.

The Directory of Open Access Journals contains more than 7,300 titles, up from 2,750 in 2008 and 560 in 2004.

What ACM does offer authors is the ability to self-archive their work on personal Web sites and institutional repositories. Previously, scholars could post the author-prepared version of an article, after peer review. Now, thanks to a new service called Author-Izer, they can create a link that grants free access to the definitive version in the ACM Digital Library. Harnad calls self-archiving "green open access" (as opposed to "gold open access," such as publishing in an open access journal). According to statistics he has compiled, more than 90% of scholarly journals enable authors to self-archive preprint versions, and more than 60% of them enable authors to self-archive the refereed final draft upon acceptance for publication. "Almost all publishers are already green," explains Harnad.

Wouldn't it be easier to let authors retain copyright and publish their work directly on their Web sites? Not necessarily, says Rous. "When some publishers shift from copyright to license, they put a lot of conditions in an exclusive publishing license that can leave the author very little room to exercise the copyright they retain," he explains. "On the other hand, nonexclusive publishing licences may undermine the subscription model since authors may publish the same work in other titles and grant permission for inclusion in any and all aggregations." The fear may seem unrealistic to senior scientists, although it is perhaps not completely unfounded in the current publish-or-perish climate, especially

With Author-Izer, ACM offers authors the ability to self-archive their work on personal Web sites and institutional repositories.

among younger researchers who are looking to inflate their publication records. "Except for the mind-set that if you retain copyright, it's yours, authors are actually in a much better position to reuse the materials and do selective reposting and distribution under ACM's copyright policy than under many of the licenses examined in our recent review of policy," says Rous.

While self-archiving has long been popular among computer scientists, it does not satisfy all scholars. Many would like to reform publishing and peer review in one fell swoop, starting from scratch and building a new system from the ground up. Their proposals often try to harness the power of the Internet to accelerate the review process. LeCun, for example, envisions a centralized online repository such as ArXiv, a site used mainly by physicists to distribute pre-peer-review versions of their papers, with an additional layer for collective review. "You still need a system to count points for tenure and promotions," LeCun says. "But if we, as a community, cannot define our own standards, then who can?"

Others are more focused on reforming distribution alone. "I strongly support open access, but I believe there's room for reader-pays journals as well," says Eric Van de Velde, a technology consultant and former Library IT Director at the California Institute of Technology. "What's wrong with the current system is the inefficiency that's involved. People don't realize how many middlemen there are between the reader and the publisher. A library doesn't buy directly from a publisher. It buys from an aggregator, so of course the aggregator gets a cut. Then the library negotiates consortium deals and bundle deals. It gets ridiculously complicated." Van de Velde would like to see publishers create a national, iTunes-like system for scholarly articles, freed from the market distortions of site licensing subscriptions. "You could pay for each article; that might be one model. Another model could be a monthly subscription that gives you access to download X number of articles. Once you put this national distribution infrastructure in place, you can experiment with all kinds of models at an individual basis."

Harnad, on the other hand, feels that broader adoption of green open access is sufficient to address scholars' goals. "Speed is a red herring. Having a central repository is a red herring. Self-archive the minute your article comes back from peer review, and CiteSeer will harvest it immediately—you can't get it any faster than that. The only thing standing between us and 100% open access for every one of the 2.5 million scholarly articles that are published every year is a few keystrokes by the author."

Of course, not all of those authors may be willing to take the additional steps to make their articles available via open access. One thing, however, is clear: Open access is growing fast in both recognition and popularity, making it a force to be reckoned with in the future of academic publishing. C

Further Reading

Bosch, S., Henderson, K., and Klusendorf, H.
Periodicals price survey 2011: Under pressure, times are changing. *Library Journal*: April 14, 2011.

Delman, S.
ACM offers a new approach to self-archiving, *Communications of the ACM 54*, 11, Nov. 2011.

Harnad, S.
The green road to open access: a leveraged transition, *The Culture of Periodicals from the Perspective of the Electronic Age.* Gacs, A. (Ed.), L'Harmattan, Paris, France, 2007.

Rous, B.
Electronic publishing models and the public good, *Nature*, Feb. 13, 2012.

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

Milestones

IEEE Awards

IEEE recently announced the recipients of the 2012 IEEE medals and service awards. John L. Hennessy, Stanford University, received the IEEE Medal of Honor "for pioneering the RISC processor architecture and for leadership in computer engineering and higher education"; Leonard Kleinrock, University of California at Los Angeles, received the Alexander Graham Bell Medal "for pioneering contributions to modeling, analysis, and design of packet-switching networks"; Faqir Chand Kohli, Tata Consultancy Services, received the IEEE Founders Medal "for early vision and pioneering contributions to the development of the IT industry in India"; William Whittaker, Carnegie Mellon University, received the Simon Ramo Medal "for pioneering contributions to mobile autonomous robotics, field applications of robotics, and systems engineering"; Edward McCluskey, Stanford University, received the John Von Neumann Medal "for fundamental contributions that shaped the design and testing of digital systems"; Mark Handley, University College London, received the Internet Award "for contributions to Internet multicast, telephony, congestion control, and the shaping of open Internet standards and open-source systems in all these areas"; Jean Walrand, University of California, Berkeley, received the Koji Kobayashi Computers and Communications Award "for contributions to the theory and algorithms for high-speed switching and network resource allocation"; Fred B. Schneider, Cornell University, received the Emanuel R. Piore Award "for contributions to trustworthy computing through novel approaches to security, fault tolerance, and formal methods for concurrent and distributed systems," Bernard Roth, Stanford University, received the Robotics and Automation Award "for fundamental contributions to robot kinematics, manipulation, and design"; and Vladimir N. Vapnik, Columbia University, received the Frank Rosenblatt Award "for development of support vector machines and statistical learning theory as a foundation of biologically inspired learning."

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:

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

STUDENT MEMBERSHIP:

- o **ACM Student Membership: $19 USD**
- o **ACM Student Membership plus the ACM Digital Library: $42 USD**
- o **ACM Student Membership PLUS Print *CACM* Magazine: $42 USD**
- o **ACM Student Membership w/Digital Library PLUS Print *CACM* Magazine: $62 USD**

All new ACM members will receive an ACM membership card.
For more information, please visit us at www.acm.org

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

RETURN COMPLETED APPLICATION TO:

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

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

Satisfaction Guaranteed!

payment:

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

o Visa/MasterCard o American Express o Check/money order

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

o ACM Digital Library ($99) $ _____

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

Total Amount Due $ _____

Card # Expiration date

Signature

DOI:10.1145/2133806.2133814

Michael A. Cusumano

Technology Strategy and Management

Can Services and Platform Thinking Help the U.S. Postal Service?

How the U.S. Postal Service might improve the efficiency of its delivery platform.

The United States Postal Service (USPS) dates back to a department created in 1775. In 2010, it had revenues of $67 billion and some 650,000 employees (down from 800,000 several years ago). But it has been in the news for years because of mounting deficits, including at least a $5 billion operating shortfall in 2011 and another $5 billion in annual obligations for retiree health benefits. The major "product" of the USPS is first-class mail. This is in a long-term decline as customers have switched to electronic substitutes. The worse news is the Postal Service expects first-class mail volume to continue dropping by nearly 50% over the next decade.[1]

What we have here is an organization disrupted by technological change, especially in information technology. The problems began with fax machines, widely introduced in the 1980s, followed by Internet-based email, which became ubiquitous in the 1990s, and then online bill paying, which has risen from 5% in 2000 to approximately 60% in 2011, as well as other electronic means of communicating.[2] Even party and wedding invitations, birthday cards, and holiday greetings are now frequently distributed via email. The Post Office still delivers packages, and this business is rising a couple percentage points each year. But the USPS has to compete with efficient private companies, led by FedEx and the United Parcel Service (UPS). The Postal Service also delivers paper-based advertising (often called "junk mail"). While this market also continues to rise by modest amounts, it could decline as well in the future as more advertising moves to the Web or mobile phones.

The USPS has special problems, too. It is obligated by federal law to deliver mail to each address in the U.S. (some 150 million locations), a costly endeavor. The price of a first-class U.S. stamp is also the same whether the mail is going one mile or several thousand miles. Moreover, unlike private companies, the Postal Service faces

ILLUSTRATION BY ALICIA KUBISTA

limits in the types of businesses it can go into. But, since 2006, the USPS has been under increased pressure to operate like any other firm—that means it must be profitable or at least break even and can no longer rely on government loans.

The Postal Service is not alone in seeing customers move to new technology-based alternatives that are cheaper and faster. Book and music publishers as well as retailers, video rental shops, travel agencies, stock brokerages, and even consumer banks represent just a few industries disrupted by the Internet. But, the financial situation has become desperate for postal workers. In December 2011, the USPS proposed to cut regular mail delivery back to five days a week, from six days; eliminate most next-day first-class mail delivery; and close another 3,700 of its 32,000 post offices as well as more than half of its 487 mail-processing centers.[2] These and other measures scheduled to go into effect over the next several years are expected to eliminate $100 billion in costs and reduce headcount by another 100,000. Such drastic cutbacks may get the Postal Service on a sounder financial footing temporarily. But if the organization continues to cut costs as first-class mail declines, eventually the Postal Service and its infrastructure will wither away.

Services and Platforms

We could simply ask private companies to deliver whatever physical mail and packages remain, though this would require a change in federal law and might also prove to be far more expensive. Sending a letter via UPS or FedEx today costs the same as sending a package and that is many times the price of a first-class stamp. But there should be another alternative, such as to grow revenues! One way to do this would be to leverage two related concepts I have been discussing in this column and in other publications for nearly a decade: *services* and *platforms*. For example, we have seen many different firms shift to offering value-added services as their basic products have become commoditized or outmoded (see "Finding Your Balance in the Products and Services Debate," *Communications*, March 2003). We have also seen firms utilize their physical or Web infrastructure as a platform for diversification and for establishing new revenue-generating partnerships with other organizations (see "The Evolution of Platform Thinking," *Communications*, Jan. 2010).[a]

The Postal Service is not alone in seeing customers move to new technology-based alternatives that are cheaper and faster.

Sometimes we see value-added service offerings that are closely related or complementary to a company's basic product or standardized service, such as customized features, expert advice, training, or integration with other companies' products and systems. Other services may be unrelated to the basic product but still leverage the same customer demand in a kind of diversification or "economies of scope" strategy. We also see some companies offering service-like versions of their products, which provide additional convenience and potentially lower costs to the customer. These "servitized" products include Software as a Service (SaaS) and Cloud Computing applications instead of packaged software. We could also consider ZipCar's transportation by the hour a service-like substitute for automobile product purchases. iTunes is a service-like substitute for purchasing music and videos. Even Rolls Royce—with its Power by the Hour program—rents the use of its aircraft engines to airlines or leasing companies that prefer not to pay the full cost of buying the planes and engines outright. So the idea here would be for the USPS and future partners to offer a wide range of mail-related and unrelated services that either complement or replace physical mail products as well as leverage the postal network of offices and trucks and their customer traffic.

Congress needs to respond here by loosening the legal constraints on the types of business into which the USPS can enter. The USPS leadership also would need to change the way it thinks about the network of post offices, mail trucks, and people. At present, these assets seem to be a costly liability that must be cut back—drastically. Reducing the scale and scope of the USPS, however, could also lower the potential value of the network as a service-delivery platform. The USPS should be able to use its vast infrastructure to create and deliver new products and services on its own as well as through partnerships. The USPS platform and future "ecosystem" of partners would resemble similar retail and service-delivery platforms such as at Amazon.com, which now sells everything from books to electronics and hosts a variety of small business partners and cloud-computing service users, or even Walmart, CVS, and Walgreens, which partner with different firms to provide a variety of products and services from their store locations.

My class at MIT did a brainstorming session and came up with a variety of ideas, several of which the USPS has already implemented on its own. In the *unrelated* to mail category, financial services such as banking (savings accounts in particular, which are not costly to maintain) and insurance (such as simple term policies or product warrantees) come immediately to mind. The USPS could provide greater access to its infrastructure to other businesses as well as other government agencies, local, state, and national—a potentially enormous market. For example, postal workers could conduct the census as they deliver mail, or process and deliver official documents, such as licenses and inspection permits, perhaps while offering notary services. Postal workers could become meter readers where this is necessary, or help the elderly pay utility bills through mobile payment devices. The USPS could put advertising on trucks or allow mobile cellphone operators to mount antennas. The trucks could carry air quality monitoring equipment.

There are also many possible val-

a Platforms and services are also the focus of the first two chapters of M. Cusumano, *Staying Power* (2010).

ue-added services *related* to mail and package delivery. The USPS could digitize and sort mail and then send it out electronically to customers who prefer not to have paper (this is already under way). Maybe the best way to charge for this would be like a SaaS or subscription service, related to volume. It could take a cue from UPS and offer advanced logistics consulting for its customers as well as handle business inventory and deliveries—become the logistics arm of local businesses. It already has alliances with FedEx and UPS for "last mile" deliveries instead of viewing these companies as competitors. The local post offices could also learn from Craig's List or Angie's List and go a step further. The offices could leverage the intimate knowledge they have of local customers and help them find local business services while helping local businesses advertise more effectively and reach the right customers.

The USPS could also create platform-based services that generate positive "network effects," that is, the services increase exponentially in value as more people sign on for them. For example, there could be a service that lets only registered users store and send paper copies of photos, greeting cards, or postcards through the mail to other registered users after sending them to a USPS Web site. The USPS would then print the photos and cards (or subcontract the printing) according to customer instructions and then deliver them. Another service could be similar to the above but target secure documents, stored electronically and valid only when delivered in physical form (official copies of birth and death certificates, marriage licenses, divorce documents, stock certificates, deeds, permits, contracts, and so forth). The USPS could create its own social media network by itself or through partnerships with firms such as Facebook. Registered users could send photos, postcards, and invitations to friends and family members. This service would generate network effects as more users and family members joined, which should allow the USPS to charge fees or monetize the service through advertising.

If these ideas sound strange to Americans, they should be familiar to residents of other countries. Post offices around the world provide all types of non-mail and mail-related services to their customers. Japan's postal network has been offering basic banking services for decades (though the financial services arm has been separated administratively from the postal arm). Germany's post office bought DHL and now competes in the international package delivery market. The Swiss postal service already has "digital mail"—it will scan your letters and send them to your computer as well as eliminate junk mail. Sweden's post office has an app that creates postcards from photos and allows users to send the postcards directly from their cellphones, without stamps.[3] The Indian postal service is also pursuing a variety of financial services that it would make available to poor rural customers.

Conclusion

In short, a more creative services and platform strategy could help reverse the USPS's financial woes and provide a more positive way forward—generate new revenue rather than continue to reduce the scale and geographic scope, and thus the intrinsic value, of the network. While additional cuts in locations and headcount may still be necessary to get costs in balance with short-term revenues, the goal would be to halt the downward spiral of continually reducing the physical footprint of the USPS that ultimately could destroy its potential value as a service-delivery platform. How much would a bank pay for the privilege of locating ATM machines or customer service agents in every post office in the U.S.? Probably a lot, and maybe enough to solve the Postal Service deficit problem in one or two bold moves. At least, the services and platform alternative seems worth serious consideration.

References

1. Donahoe, P. (U.S. Postmaster General) United States Postal Service: Changing from 6-day to 5-day delivery. Unpublished memo for the MIT Sloan Fellows Program, Seminar in Leadership, Fall 2011.
2. Greenhouse, S. Next-day mail faces postal service cuts. *The New York Times* (Dec. 5, 2011).
3. Leonard, D. The postal service is running out of options. *Bloomberg Business Week* (June 5, 2011).

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

Calendar of Events

April 15–16
Annual Computing Conference,
Springfield, MA,
Contact: Mark E. Hoffman,
Email: mark.hoffman@quinnipiac.edu

April 16–17
International Cross-Disciplinary Conference on Web Accessibility Conference,
Lyon, France,
Contact: Vigo Markel,
Email: markel.vigo@manchester.ac.uk

April 16–20
21st World Wide Web Conference 2012,
Lyon, France,
Contact: Elod Egyed-Zsigmond,
Email: elod.egyed-zsigmond@insa-lyon.fr

April 18–20
Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks,
Tucson, AZ,
Contact: Professor Marwan Krunz,
Email: krunz@ece.arizona.edu

April 18–21
International Conference on Web Information Systems and Technologies,
Porto, Portugal,
Contact: Jose Cordeiro,
Email: jcordeiro@insticc.org

April 18–21
2nd International Conference on Cloud Computing and Services Science,
Porto, Portugal,
Contact: Tony Shan,
Email: tonyscshan@yahoo.com

April 20–21
Annual Computing Conference
Canyon, TX,
Contact: Alex Rajon,
Email: ralex@wtamu.edu

April 25–27
9th USENIX Symposium on Networked Systems Design and Implementation,
San Jose, CA,
Contact: Steven Gribble,
Email: gribble@cs.washington.edu

April 29–May 2
Learning Analytics and Knowledge,
Vancouver, BC Canada,
Contact: Dawson Shane,
Email: shane.dawson@ubc.ca

DOI:10.1145/2133806.2133815

Richard Heeks

Emerging Markets
Information Technology and Gross National Happiness

Connecting digital technologies and happiness.

What would you do if you received (as I recently did) the following assignment: to help Bhutan plan the use of information and communication technologies to maximize happiness? Bhutan is a small landlocked Himalayan kingdom, sandwiched between India and China. Where other nations—explicitly or implicitly—search for wealth as their goal, Bhutan is set on a different course. Its objective is GNH not GNP: gross national happiness rather than gross national product.

Steeped in the values of Mahayana Buddhism, Bhutan has a population of approximately 700,000 in a country the size, and with similar topography, of Switzerland. It has always been different. Until the 1960s Bhutan had no schools, no hospitals, no paved roads. The first radio station was only licensed in the 1980s; television was only introduced in 1999; tobacco and plastic bags are still banned. The country's one set of traffic lights in the capital, Thimphu, were removed and replaced by a police officer when drivers deemed the lights unfriendly and frustrating.

Reflecting those sorts of views, Bhutan is also trying to take its own path when it comes to national development. The idea of GNH was first broached by King Jigme Singye Wangchuck in 1972. What was initially not much more than a passing remark slowly gathered momentum. It first grew to become a defining characteristic of both national planning and then national identity, with Bhutan now a global promoter of the happiness concept, and pushing for it to be ratified by the United Nations as a development goal. This has synchronized well in the last 10 years with the global trend of growing interest in happiness, positive psychology, well-being, and related concepts. That trend argues happiness is central to human existence and purpose.

Children in Thimphu, the capital city of Bhutan.

PHOTOGRAPH BY LAI RYANNE/OLPC

The Technology-Happiness Disconnect

Yet, so far, there seem relatively few connections made between information and communication technologies (ICTs) and happiness. Affective computing looks more at the recognition and simulation of human emotions, though could clearly extend its scope to encompass the impact of computing on happiness: one Taiwan-

ese research group's work on so-called "orange computing" is a step on this path.[4] There has been more work done on the relationship between Internet use and specific emotions, such as depression or social isolation, though with somewhat mixed conclusions.[2]

One can also see the relative disconnect between ICTs and happiness within Bhutan. The Internet and mobile phones were first permitted at the start of the 21st century, and have grown rapidly since. International Telecommunication Union figures indicate approximately 400,000 mobile subscriptions, with annual growth rates of more than 50%; and 14 Internet users per 100 population (nearly twice the level of neighboring India). That growth has been enabled by a series of policy statements and legislation, most especially the 2004 BIPS: Bhutan Information and communication technology Policy and Strategies.

But BIPS—while mentioning Gross National Happiness—makes no direct link between new technologies and happiness. This has frustrated senior staff in the Ministry of Information and Communication, leading two years ago to a vision statement that sought to assert the linkage between Bhutanese values and ICTs, and more recently inspiring the 2011 policy workshop on "ICTs for Gross National Happiness."

Finding a way forward, though, is not easy. Most ideas and most advice that circulate are of the cookie-cutter variety: standard templates that link ICTs to economic growth, profits, organizational goals, and so forth. Happiness is never mentioned. Bhutan itself seems sometimes like a young, innocent gazelle surrounded by the slavering hyenas of the corporate IT world, just waiting to pounce. Instead, it would like to live up to its billing as the land of the thunder dragon by roaring to impose its own unique agenda.

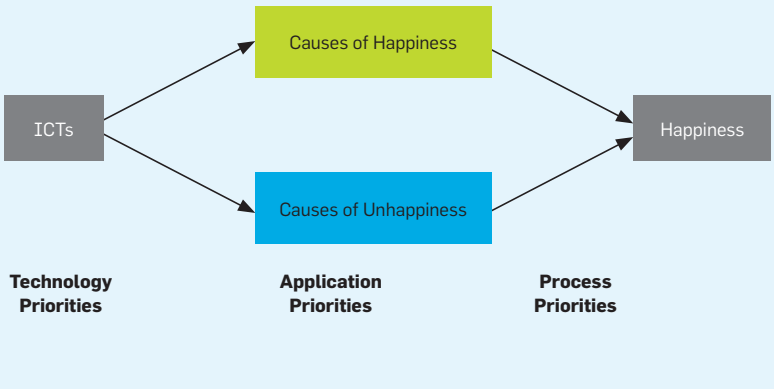
Linking ICTs and Happiness

So how could we fill in the rather blank space on the map, and relate digital technologies to happiness? An obvious starting point would be a definition of happiness. As befits a topic first discussed hundreds of years B.C. by Aristotle, there is no single agreed definition, though there is some consensus that happiness should be understood as a relatively persistent state of well-being rather than as a passing emotion or a fleeting euphoric sensation.

If defining happiness is tricky, measuring it is even more so.[1] The word rarely translates easily from English, and happiness is a social construct that varies from society to society. As an example, U.S. surveys show higher levels of happiness than those conducted in Japan. It is not that Americans are innately happier, but they inflate their responses because of their social convention that to be happy is important: "the pursuit of happiness" is, after all, a right enshrined at the start of the Declaration of Independence. Conversely, in Japan, there is no such convention; replaced instead with a certain modesty about oneself and one's own fulfillments.

It seems somewhat easier to agree on the substrates of happiness and unhappiness, and these also appear—while acknowledging some local variation—to be more cross-culturally resilient. If we are to take the connection between computing and happiness seriously, then, we would adopt something like the model shown in the accompanying figure. In the remainder of this column, I will discuss ways in which it might be applied, using Bhutan as an example.

ICTs and happiness.



Causes of Happiness

Focusing on the application priorities, what are some of the key causes of happiness, and how do they relate to ICTs? Two stand out: jobs and relationships.

A meaningful source of employment and income is central to happiness.[a] What would that mean for a developing country like Bhutan? First, that it should make a strong connection between ICTs and job creation. In many poorer countries, that connection is weak, with ICTs linked more to social than economic outcomes. But Bhutan has pushed for a place in the IT outsourcing market, and recently inaugurated the Thimphu TechPark (see http://www.thimphutechpark.com).

Alongside the generic draw of low-cost, tech-skilled graduates, one can identify a number of Bhutanese unique selling points that may help the country find its own particular niche, and create jobs from technology:

- Green IT power: Bhutan is perfectly designed for hydropower. With only 5% of potential sites developed, hydropower already provides well over half of the country's GDP thanks to exports to India,[b] and largely from environmentally friendly "run-of-river" rather than dam-based power plants.
- Pervasive English: all schoolchildren are taught in English from elementary school onward, so capabili-

a At least up to a certain level. The still-debated "Easterlin paradox" finds happiness in the U.S. (and other nations of the global North) has flatlined for the past 60 years despite growth in incomes; an outcome attributed to adaptation of expectations and to people comparing themselves relative to others rather than focusing on their absolute levels of material well-being.

b Hydropower also explains Bhutan's amazing growth rates—averaging approximately 7% per annum for the past 30 years, and pulling Bhutan up from one of the poorest countries in the world to being about halfway up the economic "league table," at least in purchasing power parity terms; now akin to countries like Jordan and Sri Lanka.

ties in English are well above those of most nations.

▸ Low attrition: the IT sector is characterized by "boomerang Bhutanese"—those who go to train and work in India for a couple of years, but then want to return home to the comforts, values, and career stability of Thimphu.

▸ Service mentality: it is easy to fall into stereotypes of orientalism or Buddhism, but Bhutan does exude a strong service culture. (A mentality that extends to driving: if there is another country in the world where slower-moving vehicles always pull over to safely let others go past, I have yet to visit it.)

Of course, Bhutan's high-tech dreams may turn out to be only that, rather than realities, and ICT jobs—even if we extend the boundary to include those running cybercafes, selling cellphones, or teaching hands-on skills—will only ever deliver well-being to a minority of the population. Countries like Bhutan will therefore also need to get serious about e-agriculture, since farming still forms the source of livelihood for the majority of the population. For poor farmers, driving up income does correlate with increased happiness, so ICTs can be introduced to support agricultural extension work—providing information on better planting, cropping, and animal husbandry that drives up yields—and to help monitor market prices so farm outputs can be sold for the best price.

Close relationships with family and friends also create happiness. ICTs clearly have the potential to disrupt these relations—ask any parent of a teenager who has seen their kid disappear for several years into the virtual worlds of Azeroth, Liberty City, or the like. Yet social media, Skype and other applications can help maintain—perhaps even strengthen—social capital at a distance. It is questionable whether any active invention or intervention is needed to help foster ICT-enabled links between migrants and their families. However, connections to "back-home" communities can be developed using applications like StoryBank (see http://www.cs.swan.ac.uk/storybank) built around digital storytelling within communities and capturing both age-old stories alongside the specific narratives of current lives.

In particular—and acknowledging the Good Samaritan—helping others seems to make us happy. This means that in Bhutan and elsewhere, we should be bringing to the fore the kind of e-mentoring, e-volunteering, e-assistance applications that have somewhat languished on the sidelines of digital economy discussions. The same is true for collective altruistic endeavors seeking to harness the power of new technology. We have patchy examples of these from around the world, such as Colombia's "digital brigades" or the Random Hacks of Kindness events. But how can such initiatives be more widely replicated, especially within the environment of developing countries like Bhutan?

Causes of Unhappiness

If the causes of happiness are multiple so, too, are the causes of unhappiness. We can use tools like the Holmes/Rahe scale of life stressors to identify poor health as a key source of grief; highlighting the importance of health informatics and, most notably for developing countries, the need for further development of m-health applications.

Empowering citizens via ICTs means, unfortunately, empowering them to do bad things as well as empowering them to do good things. There is evidence that e-pornography, online gambling, and computer crime lead to more unhappiness than happiness: "Internet use for the purpose of mischief [*is*] associated with lower levels of happiness."[2,5] So it seems the watchwords will need to be "regulated empowerment"; with countries like Bhutan not being afraid to block and control—as they have continued to do with television—and working to ensure ISPs see themselves as co-regulators, taking action against those who misuse.

An actuality or perception of social exclusion fosters unhappiness.[3] The specifics of social exclusion may vary from society to society, but there are recurrent categories of excluded individuals—the elderly, the disabled, the incarcerated. As yet, these groups rarely appear at the top of the list for development of new ICT applications; reflecting their marginalized status. But using happiness as a guide would change that. For example, given its Buddhist foundations, Thailand has also sought to integrate happiness into its development and ICT policies. It recently began an "IT for inmates" initiative, with the specific intention of using new technology to improve the aspirations and prospects of the country's prisoners.

A Challenge to ACM Professionals

Lastly—and looking at the process priorities by which technology is harnessed to make relevant applications—a recurring theme has been the need for new applications to be written. With that in mind, I would like to issue a challenge to ACM professionals: to design the best app to advance human happiness; a H-*app*-iness Challenge, if you will. It is an ideal remit for a BarCamp or for crowdsourcing from a group of graduate students (as well as for late-night conference conversations). One would likely go back to first principles—as I have started to here—of what causes happiness and unhappiness in a particular context like Bhutan's, and then design something practical to address one of those factors.

The pursuit of happiness is a founding principle of the U.S., a guiding principle for Bhutan, and a matter of ever-greater discussion worldwide. It is time we IT professionals got more involved, with some thinking outside-the-box to consider our work's contribution—or otherwise—to gross national happiness. C

References

1. Hoellerer, N.I.J. The use of qualitative and ethnographic research to enhance the measurement and operationalisation of Gross National Happiness. *Journal of Bhutan Studies 23* ,(2010), 26–54; http://www.bhutanstudies.org.bt/pubFiles/V23-2.pdf.
2. Mitchell, M.E. et al. Internet use, happiness, social support and introversion. *Computers in Human Behavior 27* (2011), 1857–1861.
3. Myers, D. *The Pursuit of Happiness*. Morrow, New York, 1992.
4. Wang, J.-F. et al. Orange computing: Challenges and opportunities for affective signal processing. In *Proceedings of the IEEE Conference on Signal Processing, Communications and Computing* (Xi'an, China, Sept. 2011).
5. Zillman, D. Effects of prolonged consumption of pornography. In *Pornography: Research Advances and Policy Considerations*, D. Zillmann and J. Bryant, Eds., Lawrence Erlbaum, Eds. Hillsdale, N.J., (1989), 127–157.

Richard Heeks (richard.heeks@manchester.ac.uk) is the director of the Centre for Development Informatics at the University of Manchester, U.K.; http://www.cdi.manchester.ac.uk.

DOI:10.1145/2133806.2133816

George V. Neville-Neil

Article development led by acmqueue
queue.acm.org

Kode Vicious
The Network Protocol Battle

A tale of hubris and zealotry.

Dear KV,

I have been working on a personal project that involves creating a new network protocol. Out of curiosity, I tried to find out what would be involved in getting an official protocol number assigned for my project and discovered it could take a year and could mean a lot of back and forth with the powers that be at the IETF (Internet Engineering Task Force). I knew this would not be as simple as clicking something on a Web page, but a year seems excessive, and really it's not a major part of the work, so it seems like this would mainly be a distraction. For now, I just took a random protocol number that I know does not conflict with anything on my network—such as UDP or TCP—and things seem to work fine. I guess my real question is why would anyone bother to go to the IETF to ask for this unless they were a company that could waste someone's time on an email campaign to get a properly assigned number?

Waiting

Dear Waiting,

Let me begin by complimenting you on the fact that you actually went so far as to find out how one might do this correctly. (I am sure many readers have just spit coffee on their screens, because none of them can remember the last time I complimented a writer in this column.) I compliment you because just recently I came across some-

one who knew the right thing to do, and then did exactly the opposite.

Because of some of the assumptions present in the original design of the Internet, some parts of the IPv4 packet header are far more precious than others, and, while the limitations of the 32-bit network address get the largest amount of attention, the 8-bit protocol field is equally as important, if not more so. With an 8-bit field, we can layer only 255 possible protocols on top of IPv4, which may seem like a lot, and since most people assume that all IP packets carry only TCP, protocol 6, there is plenty of space. It turns out that more than half of the numbers have been used for one protocol or another, leaving only 109 for use by authors of new protocols. Another problem is that IPv6, the nominal savior of the Internet, with its wider network addresses, still uses an 8-bit protocol field, so we are not getting any more space anytime soon.

The protocol field can be seen as a commons for the Internet, so let me tell you a tragedy, and one that did not have to happen. It is a story of hubris and zealotry, and unsurprisingly, involves the collision between corporations and open source.

Sometime in the late 1990s, a group of companies got together and proposed a protocol that would be standardized within the aegis of the IETF. It is not particularly important what the protocol does, but it is called VRRP (Virtual Router Redundancy Protocol) and exists so that two or more routers can act as peers in a fail-over scenario. If one router fails, another router discovers this via a means described in the protocol and takes over for the failing router. After the standard was published, two companies—Cisco and IBM—both claimed to have patents to some of what the protocol did. Cisco released its claimed intellectual property under a RAND (reasonable and nondiscriminatory) license. In nonlegal terms this means people could implement VRRP, and Cisco would not chase them down with expensive claims. RAND licenses are often used in software standardization processes.

Unfortunately, there is a segment of the open source community that is incapable of playing well with others, when those others don't play the way they want them to. For those who have not had to deal with these people,

Figure 1. One view of an IPv4 packet.

```
0x0000:  4500 0045 bf49 0000 ff11 7902 c0a8 010a
0x0010:  c0a8 0101 d688 0035 0031 8a10 39e6 0100
0x0020:  0001 0000 0000 0000 0c70 3034 2d75 6269
0x0030:  7175 6974 7906 6963 6c6f 7564 0363 6f6d
0x0040:  0000 0100 01
```

Figure 2. An easier-to-read view.

```
14:19:02.997326 IP (tos 0x0, ttl 255, id 48969, offset 0, flags
[none], proto UDP (17), length 69)
192.168.1.10.54920 > clearspot.domain:
[udp sum ok] 14822+ A? p04-ubiquity.icloud.com. (41)
```

it is a bit like talking to a four-year-old child. When you explain checkers to your niece, she might decide she does not like your rules and follows her own rules. You humor her, she is being creative, and this is amusing in a four-year-old. If you were playing chess with a colleague who suddenly told you that the king could move one, two, or three places in one go, you would be upset, because this person would obviously be trying to confuse, or insane.

Have I lost my mind?! What does this have to do with VRRP or network protocols?

The OpenBSD team, led as always by their Glorious Leader (their words, not mine), decided that a RAND license just was not free enough for them. They wrote their own protocol, which was completely incompatible with VRRP. Well, you say, that's not so bad; that's competition, and we all know that competition is good and brings better products, and it is the glorious triumph of Capitalism. But there is one last little nit to this story. The new protocol dubbed CARP (Common Address Redundancy Protocol) uses the exact same IP number as VRRP (112). Most people, and KV includes himself in this group, think this was a jerk move. "Why would they do this?" I hear you cry. Well, it turns out they believe themselves to be in a war with the enemies of open source, as well as with those opposed to motherhood and apple pie. Stomping on the same protocol number was, in their minds, a strike against their enemies and all for the good. Of course, it makes operating devices with both protocols in the same network difficult, and it makes debugging the software that implements the protocol nearly impossible.

In the end the same thing is going to happen as happens when your four-year-old niece upends the checkers game in frustration. She runs away crying, and you are left to pick up the pieces. A few of us now have to take this protocol and actually get it a proper protocol number and then deal with the fact that legacy devices are still using the old, incompatible protocol.

Now I think you see why I wanted to compliment you. Doing the right thing in the commons is good for all of us.

KV

Dear KV,

One of my coworkers seems to be completely allergic to useful tools. For example, rather than extend a program to interpret new data properly in one of our log generators, he instead memorizes the format and reads it off the screen when something breaks. He seems to take great pride in this, but I am not really sure why. Wouldn't a tool make more sense?

Hexed by Dumps

Dear Hexed,

I am not sure how many variations on the "Tools are Useful" theme I will eventually write, but clearly my message has not penetrated to the depth I would like. In part, I think the problem is that geeks—and I include myself in this group—enjoy the feeling they know things—in particular, things they think others do not know. You can see this in the in-jokes we tell one another, such as the number of times the number 42 appears in code. It is this love of arcana, I think, that leads people to be proud of the fact they can get along by reading hex dumps.

Of course, the problem is that some ways of looking at data are more prone to error than others. For instance, as someone who looks at a lot of network packets, I can tell you the example shown in Figure 1 is an IPv4 packet, and I can then scan through it to find the IP addresses and other bits I need to know to diagnose what is wrong with a host's network connection.

Compared with Figure 1, though, I would rather read the example shown in Figure 2. Sure, Figure 2 is also gobbledygook to the average person, but it is far easier to read and understand for the professional who is trying to get a job done. It is also just as compact as the first example, so an argument cannot even be made that the hex dump is quicker to read. The real issue is that getting a job done is what so many geeks lose sight of when they are preening themselves over their 1337 hex-dump reading skills; it is not about knowing the arcana, it is about solving a problem. If you want to learn and work purely in arcana, then you should study the Kabbalah, or go on "Jeopardy!", where you can now be beaten by a computer. If you want to get a job done, then use or build the tool that will give you the clearest view of the problem.

KV

Related articles on queue.acm.org

Network Front-end Processors, Yet Again
Mike O'Dell
http://queue.acm.org/detail.cfm?id=1530828

You Don't Know Jack about Network Performance
Kevin Fall, Steve McCanne
http://queue.acm.org/detail.cfm?id=1066069

The Next Big Thing
Kode Vicious
http://queue.acm.org/detail.cfm?id=1317398

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

DOI:10.1145/2133806.2133817

Jill Ross, Elizabeth Litzler, J. McGrath Cohoon, and Lucy Sanders

Broadening Participation
Improving Gender Composition in Computing

Combining academic and industry representation, the NCWIT Pacesetters program works to increase the participation of girls and women in computing.

"WHAT GETS MEASURED gets done." Taking this maxim to heart, a group of highly committed member organizations from the National Center for Women & Information Technology (NCWIT) formed a nationwide U.S. program "Pacesetters." Twenty-four academic and corporate organizations committed to adding 1,000 "Net New Women" to the U.S. computing talent pool by 2012.

Net New Women are technical women who would otherwise not have pursued or remained in computing careers. These are the women who have little or incorrect information about computing careers; who never experienced an engaging introduction that sparked their interest in computing; who had an interest, but left because no one encouraged them and told them they could succeed in computing; or who switched to a different career after tiring of the isolation or career stagnation they experienced. Understanding the urgency of this situation, NCWIT Pacesetters are creating a model for change that they hope will finally "move the needle," both within organizations and on a national scale, in real and quantifiable ways.

NCWIT is a rapidly growing coalition of more than 300 corporations, academic institutions, government agencies, and non-profit organizations devoted to women's full participation in computing (see http://www.ncwit.org).

Representatives from 14 Pacesetters organizations are featured in video stories promoting their results as part of the Sit With Me campaign in use by over 300 NCWIT organizations (see http://sitwithme.org/tag/pacesetters/).

PHOTOGRAPHS COURTESY OF SIT WITH ME

The Difference

Many successful programs work to enrich women's experiences with computing. For example, CRA-W offers a host of professional development programs that spark women's interest and skills in research careers. In addition to building women's sense of belonging to a community, mentoring and peer support are hallmarks of programs like ACM-W student chapters, ABI's Grace Hopper Celebration of Women in Computing, the regional Celebrations of Women in Computing, and the NCWIT Award for Aspirations in Computing.

The Pacesetters program is the first of its kind where organizations come together, work across corporate and academic organizational boundaries, and identify effective ways to recruit or retain a specific number of technical women, all within an aggressive timeframe and holding shared accountability to themselves and the public for achieving a common quantifiable goal.

How the Program Works

Pacesetters combine top-down and bottom-up approaches for progress based on both research findings about organizational change and observations of what works for NCWIT members.[1–6] Each Pacesetters organization must have meaningful participation by executive leaders who work top-down, and change leaders who work bottom-up. Together they build internal teams, develop and fund the needed programs, and share their results.

This program design has several advantages. Executive leaders actively engage and can influence people, policy, and resources within the organization, while providing visible endorsement. Change leaders complement these executive efforts by building out an extended team, including people in a variety of key roles across the organization. Together they develop a broadly shared vision that takes into account group norms and specialized knowledge. This results in a set of organizational approaches for collectively reaching a quantified overall goal of more women added to the technical talent pool.

Pacesetters' approaches include actively recruiting graduate and undergraduate students; retaining them in the major through curricular, pedagogical, and community innovations; developing and raising awareness of mid-career options; and fostering technical innovation by facilitating women's contributions. For the most part, approaches rest on research-informed promising practices that are tailored to the particular conditions at a Pacesetters organization. Early evaluation shows tangible value in building this type of Pacesetters learning community around a shared and urgent goal (see http://www.ncwit.org/work.pacesetters.html).

NCWIT hosts annual Pacesetters Roundtables that bring executive leaders face to face with team change leaders from each of the 24 participating organizations for focused working sessions. The momentum generated by the roundtables is further spurred through NCWIT's leadership visits with each Pacesetters organizations' executives or deans to discuss their specific approach, leverage NCWIT and other research-based interventions, and encourage an accelerated pace for change.

Pacesetters organizations include:

- AT&T
- ATLAS Institute
- Bank of America
- Boehringer Ingelheim
- Cal Poly at San Luis Obispo
- Carnegie Mellon
- Georgia Tech
- Google
- IBM
- Indiana University Bloomington
- Intel
- Microsoft
- Pfizer
- Qualcomm
- Santa Clara University
- UC Irvine
- UC Santa Cruz
- University of Colorado Boulder
- The University of Texas at Austin
- University of Virginia
- University of Washington
- Villanova University
- Virginia Tech

Organizational Results

Pacesetters use innovative change strategies to reach their goal. Some specific examples include:

- UT Austin developed an "in-reach" strategy to target undeclared freshman women already on campus. They doubled the number of new female students in one year by requesting 40 slots for focused recruiting, targeting students from the First Bytes summer camp, providing faculty and student mentoring for new freshman women, and offering NSF scholarships to selected freshmen women.
- Indiana University used a change model by NCWIT called the Strategic Planning for Retaining Women in Undergraduate Computing Workbook and focused faculty on best practices in pedagogy. They doubled the number of female undergraduate majors in the School of Informatics and Computing, from 75 to 150 in 18 months.
- Virginia Tech decided to connect personally with women residents of on-campus housing. They sent teams of CS faculty, advisors, and students to interact as mentors and peers with female potential and current students. New "designer minors" are offered that combine CS with other disciplines. They reported a 56% increase in the number of female high school students who met with school representatives and showed interest in their programs.
- Intel helped mid-career technical women navigate the company culture and build confidence. They piloted a program called Command Presence Workshop where senior technical women facilitated half-day sessions on successful presenting to decision-making audiences. The workshop gave this specialized training to over 100 mid-level technical women and they continue to measure the impact.
- Google helps undergraduate women prepare for the technical work force. They built a new program for college women, brought them to Google's offices and held a career development panel with engineers. The women participated in mock interviews. As a result, the number of applicants grew and Google doubled the number of women software engineering summer interns in 2011 compared to 2010.

These examples from the Pacesetters pilot demonstrate that with focused effort, executive commitment, and by working across organizations, significant results are possible in a short timeframe.

National Results

Pacesetters set a goal of recruiting or retaining 1,000 technical women in the U.S. computing work force by 2012 and reported 568 Net New Women in May

2011. Evaluations are continuing to assess progress toward the goal.

As Pacesetters organizations worked toward their goals, a strong need arose for a national platform to publicly share results. Together with NCWIT and brand marketing firm BBMG, Pacesetters helped conceive a national advocacy campaign called Sit With Me. The invitation, "Will You Sit With Me?" gives everybody (men, women, technical, non-technical) the chance to take a small, but symbolic, action to "sit" in solidarity with women in computing, raise visibility for their contributions, and ask others to do the same. A Web site, Facebook page, and Twitter stream encourage action around the theme "Sometimes you have to sit to take a stand." Sit With Me is currently in use by over 300 NCWIT members' organizations. Representatives from 14 Pacesetters organizations are featured in video stories publicly promoting their results (see http://sitwithme.org/tag/pacesetters/).

Lessons Learned

Achieving results requires significant participant effort, and each Pacesetter struggled with parts of the process. Perhaps most of all, time was a challenge. The Pacesetters program work overlays many other professional and personal responsibilities; these are busy people with many demands on their time, and change takes time.

Most Pacesetters agree that the timeline for this pilot was too short (24 months), yet they also reported that the short length led to more urgency and as a result, accelerated the pace of change within their organizations. It took most organizations a year to confirm their goals and strategies, and not all of them did. Every Pacesetter has unique conditions to accommodate, so practices had to be tailored to fit their specific situation, requiring even more time. Goal achievement was also negatively affected by turnover of key personnel.

Going forward, subsequent cohorts may need to be launched more quickly, or the cohort timeline may need to be extended to allow more time for implementation. It is also clear from this pilot program that additional consulting services are needed for Pacesetters during their strategic planning phases.

Not surprisingly, the academic and corporate Pacesetters progressed toward their goals at different rates due to their dissimilar organizational structures. Many academic Pacesetters had NCWIT Extension Services Consultants who provided guidance and support that enabled them to move quickly to articulate a goal. Some corporations struggled with aligning their organizational and Net New Women goals. Once aligned and focused, however, corporate Pacesetters will see larger numbers of technical women participating, supported by their well-established human resources policies and procedures.

Benefits

Data from program evaluation surveys suggests 86% of Pacesetters derive new ideas and fresh perspectives on diversity issues at their organizations as a result of Pacesetters participation. More academic institutions than corporations reported benefits from Pacesetters participation (92% and 78%, respectively). Both agreed that quantifiable goals increased their focus on "the big picture." They thought NCWIT leadership visits served as a catalyst for their efforts and helped legitimize staff time spent on the program.

The annual Pacesetters Roundtable also has a positive impact on participants. One Pacesetter said. "I think the best thing about Pacesetters was the chance to meet and network with other academic institutions particularly who have some of the same concerns about getting women into computing, and to learn about different approaches that people are taking and what kind of success they're having. Overall, I think the meetings were very useful for networking and for information from speakers who touched a nerve and gave us ideas."

Both academic and corporate Pacesetters describe the impact of the program as a stimulus for action within their organizations. In a recent evaluation survey academic Pacesetters wrote, "While we have always been interested in increasing our female population, being a part of Pacesetters keeps this goal at the forefront, and encourages us to make and measure concrete goals." and "Pacesetters' goals, resources, and contacts have provided structure, support, and accountability, which are helping us define and pursue our goals." Corporate Pacesetters expressed similar views, "Having goals, a strategic focus, key company stakeholders involved, and a strong NCWIT team leading the way and pushing a bit from behind is crucial and leads to action!" Pacesetters said overall the program increased their visibility, awareness, and connections to people both within and across organizations. A Pacesetter commented, "In addition to raising our own awareness, one of the unexpected benefits has been the relationship building with other Pacesetter organizations."

Next Steps

NCWIT's continuing support for Pacesetters' efforts enables an even greater impact as some organizations have already set new goals. For example, Indiana University committed to doubling the number of women in their programs again this year. A new cohort of Pacesetters will be recruited to continue national progress. As successive cohorts of Pacesetters organizations contribute Net New Women to the national talent pool, the representation of women in computing should move toward gender balance. More information about Pacesetters and how your organization can become involved is available at info@ncwit.org. C

References

1. Damanpour, F. and Schneider, M. Phases of the adoption of innovation in organizations: Effects of environment, organization and top managers. *British Journal of Management 17*, 3 (Mar. 2006), 215–236.
2. Fernandez, S. and Rainey, H.G. Managing successful organizational change in the public sector. *Public Administration Review 66*, 2 (Feb. 2006), 168–176.
3. Meyer, C.B. and Stensaker, I.G. Developing capacity for change. *Journal of Change Management 6*, 2 (Feb. 2006), 217–231.
4. Murray, E.J. and Richardson, P.R. Fast Forward: A new framework for rapid organizational change. *Ivey Business Journal* (2003), 1–5; Reprint #9B03TB03.
5. Stainback, K., Tomaskovic-Devey, D., and Skaggs, S. Organizational approaches to inequality: Inertia, relative power, and environments. *Annual Review of Sociology, 36* (2010), 225–247.
6. Williams, D.A. and Clowney, C. Strategic planning or diversity and organizational change. *Effective Practices for Academic Leaders 2*, 3 (Mar. 2007), 1–16.

Jill Ross (jill.ross@colorado.edu) is the director of the Workforce Alliance and Pacesetters Program at NCWIT in Boulder, CO.

Elizabeth Litzler (elitzler@u.washington.edu) is the director for research at the University of Washington Center for Workforce Development.

Joanne McGrath Cohoon (jmcohoon@virginia.edu) is an NCWIT senior research scientist and a member of the Department of Science, Technology, and Society at the University of Virginia.

Lucy Sanders (lucinda.sanders@colorado.edu) is CEO and cofounder of NCWIT in Boulder, CO.

DOI:10.1145/2133806.2133818

Selma Tekir

Viewpoint
Reading CS Classics

Revisiting required reading.

We often focus so much of our attention on our particular research areas that we do not fully utilize the potential coming from the core theoretical computer science. We lack the fundamental theoretical knowledge of the field. Moreover, the computer science classics are unknown to many computer scientists. Knowledge of the theories of computer science helps in understanding the limitations of the field. This directly influences your ongoing research by providing you with new perspectives and insights. In addition, the stories of the pioneers of the field inspire young professionals, provide a common history to unite the community, and facilitate the recognition of computer science as an independent science and profession.

With these ideas in mind, I organized CS classics meetings in my computer engineering department during the last summer term. Our group selected a subset of classics to initialize the project. The selected classics and their respective ordering reflected our personal interests; in the end, they become part of a coherent whole.

It can be a good practice for CS professionals to compile their own list of classics that highlights some key scientific concepts of the field. Such an attempt improves the understanding of the field and serves as a valuable source of reference, as this Viewpoint attests. Our group discussed these CS classics:

- "The Emperor's Old Clothes," C.A.R. Hoare
- "An Axiomatic Basis for Computer Programming," C.A.R. Hoare
- "Gödel's Undecidability Theorem," S.F. Andrilli
- "Computing Machinery and Intelligence," A.M. Turing
- "Reflections on Trusting Trust," K. Thompson
- "The Humble Programmer," E.W. Dijkstra
- "An Interview with Edsger W. Dijkstra," P. Frana
- "Computer Programming as an Art," D. Knuth
- "The 'Art' of Being Donald Knuth," E. Feigenbaum
- "Donald Knuth: A Life's Work Interrupted," E. Feigenbaum

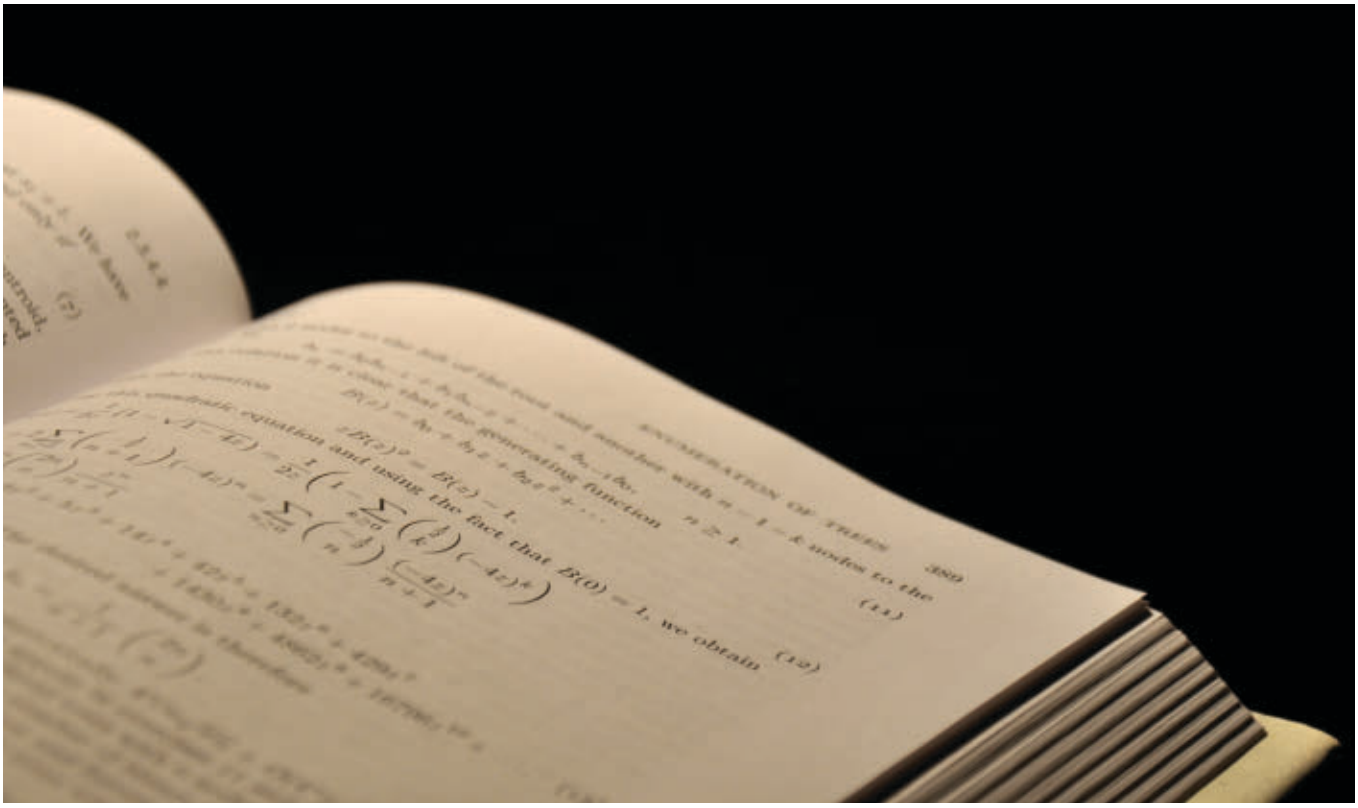
We found these intellectual gatherings quite useful and subsequently decided to make the CS classics group reading a regular activity of our academic environment. Here, I give an overview of the classics we discussed and encourage further reading.

Knowledge of the theories of computer science helps in understanding the limitations of the field.

Classics Overview

In reading Hoare,[6,7] you learn about the computing industry of the 1960s and 1970s in Britain. The programming languages community of those years was also well described in the reading. Hoare wrote a more efficient sort algorithm than the one invented by D.L. Shell.[9] When he had the opportunity to hear about the recursive procedures in an ALGOL 60 course, Hoare realized this mechanism is the right way of expressing his new sort algorithm, which is the original QuickSort. The moral of this example is that one should communicate with people to seek better solutions to the problems at hand and extend the existing solutions. His remark on simplification is of high importance as well. A simple, reliable core is critical for a programming language, an operating system, and even for any software product. With this realization, Hoare provides a foundation for the formal proofs of programs by an algebraic assertions-based approach, which is named as "An Axiomatic Basis for Computer Programming."[7]

Gödel's undecidability theorem[1] states that any mathematical system containing all the theorems of arithmetic is an incomplete system. This opens the way for Turing to introduce the famous halting problem: There is no general algorithm that can always correctly predict whether a randomly selected computer program will run or not.[11] Before knowing about Gödel and his undecidability theorem, Turing stands out as the most prominent figure in computer science. After you hear about Gödel's work, you realize Turing is standing on the shoulders of giants. The proof of the undecidability theorem has important implications



for computer science by introducing the Gödel numbering scheme, which introduces unique numbering to each symbol, formula, or proof in the system. This system is the basis of the computer numbering systems that provide unique representation to every programming construct: due to this property, code can be treated as data.

The idea of this unique numbering system can be better explained by the challenge of writing a source program that, when compiled and executed, will produce as output an exact copy of its source. It is a Turing machine SELF that is printing itself. The SELF machine is constructed such that it contains two concatenated machines and one of them is the Gödel number equivalent of the other. Such a self-reproducing program is introduced by Ken Thompson in "Reflections on Trusting Trust"[10] as the most primitive version of today's trojans. When you see the scientific layers on top of each other like the one presented, you begin to appreciate the real beauty of science and the scientific developments.

By reading Dijkstra independent from his contemporary Hoare you have information about the computing environment of that era. To make a correct assessment of that time period and the products that were launched, independent but consistent views are required. In this sense, Dijkstra and Hoare's identical views on ALGOL 60 help us appreciate this programming language. Additionally, the realization of the recursion mechanism by both is spectacular—a good example of the axiom "great minds think alike."

Dijkstra's dialogue with his professor cannot be overlooked. Most significant is his realization of the high intellectual challenge of programming and the professor's encouragement that made him one of the greatest minds of computer programming.[5]

One lesson comes from the huge abstraction capability/potential inherent in computer science. Abstraction is extending the viewpoint in a way that the specificities of the problem can be reflected in a better way rather than being vague. The tools we work with can then have vital importance in abstracting. Dijkstra's comment on computing tools is remarkable in this sense:[2] he states that computing tools have direct influence on the thinking habits of their users. If you constrain yourself with one specific tool, your thinking becomes constrained in the boundaries of this tool. You continue to stay at the same level of thinking as the creator of this tool in accordance with the famous quote by Einstein: "The significant problems we face cannot be solved at the same level of thinking we were at when we created them."

Reading CS classics widens your perspective by introducing stable, timeless ideas.

Donald Knuth is extraordinary with his perspective on computer programming.[3,4] His definition of programming identifies the right balance between conceptual clarity and implementation efficiency. He says: "Programming is the art of telling another human being what one wants the computer to do."

In understanding the importance of this definition, one should realize it is beyond the traditional definition of the task of telling a computer what to

PHOTOGRAPH BY GEET DUGGAL

do. Knuth's viewpoint is more encompassing and is helpful in understanding the diversity and convergence of programming languages. Moreover, it points out an important trade-off between conceptual clarity and implementation efficiency. When the task is to define a job to a computer, low-level instructions are better in terms of execution efficiency. However, people have difficulty in understanding such written code. When you try to describe a task to a human being, you can skip some steps because humans are good at filling in the blanks; machines have difficulty doing this. The best is to compromise: to discuss the task at a high level but in a manner that can be converted into a machine-processable format as indicated by Knuth's prodigious statement.

Knuth's opinions about tools are similarly noteworthy.[8] Like Dijkstra, he thinks the tools we utilize have direct influences on what we accomplish. He puts emphasis on the artistic aspect of programming. According to him, the beauty and aesthetics of tools improves the enjoyment of users and enhances their thinking habits. Combining the assessments of Dijkstra and Knuth, what we (plan to) do is not independent of how we (plan to) do it. The process is a good indicator of the resultant product most of the time.

Conclusion

Reading CS classics widens your perspective by introducing stable, timeless ideas. You escape the popular themes of your times and evaluate the field from a more literal position. You learn about the qualities that make a person a great scientist. You realize those people are delighted to think over problems. By learning the history of computers and studying the lives and works of eminent computer scientists we all recognize the true merit of being part of such a respectful profession and privileged community.

I hope this Viewpoint raises readers' interest in CS classics, causes CS professionals to revise their reading lists to include these books and articles, and inspires them to further extend their classics library. Time spent on the classics is not wasted but is an investment in your career as a researcher as well as an educator. C

References

1. Andrilli, S.F. Gödel's Undecidability Theorem. *Applications of Discrete Mathematics*. J.G. Michaels and K.H. Rosen, Eds. McGraw-Hill, 1991.
2. Dijkstra, E.W. The humble programmer. *Commun. ACM 51*, 10 (Oct. 1972), 859–866; DOI: 10.1145/355604.361591.
3. Feigenbaum, E. Donald Knuth: A life's work interrupted. Shustek, L., Ed. *Commun. ACM 51*, 8 (Aug. 2008), 31–35; DOI: 10.1145/1378704.1378715.
4. Feigenbaum, E. The 'art' of being Donald Knuth. Shustek, L., Ed. *Commun. ACM 51*, 7 (July 2008), 35–39; DOI: 10.1145/1364782.1364794.
5. Frana, P. An interview with Edsger W. Dijkstra. T.J. Misa, Ed. *Commun. ACM 53*, 8 (Aug. 2010), 41–47; DOI: 10.1145/1787234.1787249.
6. Hoare, C.A.R. The emperor's old clothes. *Commun. ACM 24*, 2 (Feb. 1981), 75–83; DOI: 10.1145/358549.358561.
7. Hoare, C.A.R. An axiomatic basis for computer programming. *Commun. ACM 12*, 10 (Oct. 1969), 576–580; DOI: 10.1145/363235.363259.
8. Knuth, D.E. Computer programming as an art. *Commun. ACM 17*, 12 (Dec. 1974), 667–673; DOI: 10.1145/361604.361612.
9. Shell, D.L. A high-speed sorting procedure. *Commun. ACM 2*, 7 (July 1959), 30–32; DOI: 10.1145/368370.368387.
10. Thompson, K. Reflections on trusting trust. *Commun. ACM 27*, 8 (Aug. 1984), 761–763; DOI: 10.1145/358198.358210.
11. Turing, A.M. I—Computing machinery and intelligence. *Mind LIX*, 236 (1950), 433–460; http://mind.oxfordjournals.org/content/LIX/236/433.full.pdf+html

Selma Tekir (selmatekir@iyte.edu.tr) is a postdoctoral instructor in the Department of Computer Engineering at Izmir Institute of Technology in Turkey.

I would like to thank Burcu Külahçioğlu, Murat Özkan, and Serap Şahin for the joyful CS classics meeting we had.

DOI:10.1145/2133806.2133819

Daniel Soper

Viewpoint
Is Human Mobility Tracking a Good Idea?

Considering the trade-offs associated with human mobility tracking.

Two years ago, the *Communications* Web site featured a news item reporting on the advances being made in the area of human mobility research.[8] With increasing numbers of smartphones incorporating GPS capabilities, it is now possible to accurately track the locations and movements of individual human beings on a large scale. While modeling and predicting human movement patterns may yield great benefits for mankind, it also has the potential to influence our lives in unexpected and possibly undesirable ways. For this reason, the computing community must carefully weigh the costs and benefits of human mobility tracking, and consider what implications this increasingly common activity may have for our lives and our communities.

A detail of Eric Fischer's visualization depicting MapMyRun public GPS logs from June 13 through August 9, 2011, in San Francisco, CA.

VISUALIZATTOIN BY ERIC FISCHER

An Invitation to Dinner

Consider the following scenario: After a long workweek, a few of Cho's colleagues invite her to visit their favorite restaurant for a relaxing dinner. Cho tells her friends she will think about their offer, and returns to her office. As the end of the workday nears, a message from an unknown sender appears on Cho's smartphone. She warily opens the message, and is pleased to find a digital coupon that entitles her to a free appetizer at the very restaurant her friends were planning to visit. "Looks like I'm going to dinner!" she thinks to herself. With her digital coupon in hand, Cho walks down the hallway to meet up with her colleagues.

On the surface, this appears to be a winning scenario for everyone involved—Cho gets a free appetizer to share with her friends, the restaurant gets her business for the evening, and the unknown sender of the message earns a modest profit for its targeted advertising efforts. Upon deeper reflection, however, you may feel there is something unsettling about what has happened to Cho. How did the sender know Cho might be visiting the restaurant after work? Has her privacy been violated? Is it unethical or even illegal for an unknown entity to profit from being able to predict her movements?

Tracking Human Mobility

At present, mobile phones provide the best means of gathering information about individual human movements on a large scale.[4] Whenever your mobile phone is on, your wireless service provider records the cellular tower that is currently assigned to handle

your requests. Because the service provider knows the location of each tower, it also knows within a certain margin of error what *your* location is at that moment in time. By keeping track of your location, the provider can then easily build a model of your movement patterns.

Until recently, the distance between cellular towers meant that location information gathered from mobile phones was only accurate to within a few hundred meters. The good news for those who wish to protect the privacy of their location and movements is the top four wireless service providers in the U.S.—AT&T, Verizon, Sprint, and T-Mobile—all currently have policies in place to protect customer location information. Indeed, protecting this information may be a point of competition among wireless providers. The latest smartphones, however, include GPS capabilities that allow their locations to be pinpointed with precision. Savvy developers can easily write mobile apps that tie into your smartphone's GPS capabilities, thus allowing parties other than your wireless provider to know your location.

Human mobility tracking is currently a divisive and highly controversial issue.

With accurate GPS data, any of these parties could track, model, or predict your movements, using the results to their own advantage. As the CEO of Google—which makes several GPS-enabled mobile apps—remarked, "We know where you are. We know where you've been. We can more or less know what you're thinking about."[13] It is not difficult to imagine this is what happened in the example scenario involving Cho.

How Predictable Are You?

Many of us like to think we are spontaneous and unpredictable, but in reality this does not appear to be the case. In a study involving 50,000 mobile phone subscribers, researchers found they could, on average, accurately predict the movements of individuals 93% of the time.[12] Surprisingly, the movements of every single person in their study could be predicted with at least 80% accuracy, supporting the conclusion that most of us follow simple, highly predictable movement patterns. Given that this study relied on cellular tower data rather than GPS coordinates, we can soon expect GPS-based models of human mobility that are even more accurate.

This ability to track, model, and accurately predict human movements has important implications for your personal privacy. The reason for this is simple—knowing where you are or where you will be in the near future has value to others beside yourself. Whether it is scientists interested in improving society, corporations interested in profit, or governments interested in enhancing security, knowledge of your movements is a valuable commodity. Given the approximately five billion mobile phone subscribers worldwide,[6] these concerns are by no means inconsequential. It is therefore incumbent upon the computing community to carefully consider how and for what human mobility tracking should be used, as it is we who will largely shape the destiny of this increasingly common activity.

The Good, the Bad, and the Ugly

Human mobility tracking and modeling has great potential to improve the lives of people everywhere. When studied collectively, information about our locations and movements can be used for such noble purposes as reducing traffic congestion, improving urban planning, arresting the spread of disease, or studying interpersonal interactions.[4,9] When coupled with miniature environmental sensors, mobile computing devices with location tracking capabilities can be transformed into nodes on a wide-area sensor network, thus allowing chemical, biological, or radiological hazards to be identified and addressed.[5] We can also benefit directly as individuals from human mobility tracking. Concerned parents, for example, might find comfort in knowing the locations and movements of their children. Similarly, emergency responders could use the GPS coordinates of our mobile phones to come to our aid if we become lost or injured.

Despite the many potential benefits of human mobility tracking, information about our movements could also be used for more controversial purposes. Law enforcement agencies, for example, might track the movements of peaceful protesters, while lawyers might track spouses suspected of infidelity. Insurance companies might adjust your rates based on how often you visit fast-food restaurants, or on how often you exceed the speed limit while driving. Employers might be interested in knowing where their employees go while not at work. As was the case in the example with Cho, marketing companies could also use knowledge of our whereabouts to craft targeted advertising campaigns intended to influence our behavior.

Governments, of course, are also highly interested in knowing our locations and movements. In the U.S., for example, the government has been tracking citizens' international travel patterns for years.[1] Despite several

past failures,[11] federal attorneys continue to argue that the government should be granted access to mobile phone location information without probable cause.[2] Although the courts have consistently rejected these arguments, the implication is clear—governments place a great deal of value on knowing our whereabouts and movement patterns, and are actively seeking a legal pathway to gaining such information.

To Track or Not to Track

As with any new technology that has implications for our personal privacy, human mobility tracking is currently a divisive and highly controversial issue. The future of this activity will therefore depend on how we as a society collectively judge its costs and benefits. On the one hand, concerned citizens, privacy advocates, and organizations such as the ACLU recognize the immense potential for government malfeasance and corporate abuse of this technology.[10] Such concerns should not be ignored or easily dismissed, because the current federal law that governs electronic privacy—the Electronic Communications Privacy Act—does not specifically address human mobility tracking. On the other hand, many scientists, corporations, and individual consumers see immense social and monetary value in this technology. Evidence for this can be seen in the rapid growth of GPS-based services such as Foursquare, Gowalla, and Facebook Places, and in the fact that revenue from human mobility tracking is expected by some researchers to reach nearly $13 billion by 2014.[7]

When coupled with demonstrable scientific and social benefits, the potential for companies to earn so much money from human mobility tracking makes it nearly impossible to imagine a future without it. If the expansion of human mobility tracking is indeed inevitable, then we must collectively be willing to accept the responsibilities that come along with it. First, I believe we owe it to ourselves to further study the impacts of this technology. Further, we must recognize the privacy implications of human mobility tracking, and act ethically when developing location-based services. Finally, we must all remain eternally vigilant against any corporations, governments, or individuals who might seek to misuse or abuse this technology at our expense.

In a sense, we all *choose* to allow these parties to gather information about us.

The Path Ahead

Given the many potentially unethical or otherwise questionable uses to which human mobility tracking might be applied, these activities seem a natural target for legal regulation. Direct government regulation of private-sector human mobility tracking could, however, impede the many efforts in this area that are being directed at genuinely altruistic ends. For this reason, I believe all wireless service providers, mobile app developers, and other computing professionals involved in GPS-based human mobility tracking should take proactive steps toward self-regulation. A good place to start would be the voluntary adoption of a set of principles designed to protect customer location data, such as those proposed in the Wireless Association's *Best Practices and Guidelines for Location Based Services*.[3] Further, although mobility tracking is just beginning to enter the public consciousness, researchers in ubiquitous computing have been studying these issues for several years, and their efforts may prove to be a wellspring of useful ideas. If we as a global community of computing professionals can agree to behave responsibly, then it may be possible to help protect our collective privacy, stave off government regulation, and allow human mobility modeling to develop at a natural pace.

Whether or not researchers, corporations, and governments are able to acquire and benefit from knowledge about our individual locations and movements is largely up to us. In a sense, we all *choose* to allow these parties to gather information about us. By opting to use the mobile technologies and apps that enable our locations and movements to be recorded, we are agreeing, either explicitly or implicitly, to allow others to benefit from our personal information. Once we have lost ownership of our location information, another party may, within the boundaries of the law, use or sell that information for profit without our permission. While for now we might take some comfort in knowing we can flip the switch to "off," the increasingly ubiquitous nature of mobile computing technologies implies they will soon become difficult to avoid.

Conclusion

Ultimately, it appears that if we want to enjoy the many benefits afforded by mobile computing technologies, we must be willing to give up at least some of the privacy that was enjoyed by previous generations. This is, it would seem, the price we all must pay to live in a modern, technology-driven world. C

References

1. Barabasi, A.-L. *Bursts: The Hidden Pattern Behind Everything We Do.* Penguin Group, New York, 2010.
2. Buchanan, M.B. and Eberhardt, R.E. In the matter of the application of the United States of America for an order directing a provider of electronic communication service to disclose records to the government. United States Court of Appeals for the Third Circuit, 2010.
3. CTIA. *Best Practices and Guidelines for Location Based Services.* The Wireless Association, Washington, D.C., 2010.
4. González, M.C., Hidalgo, C.A., and Barabási, A.-L. Understanding individual human mobility patterns. *Nature 453* (2008), 779–782.
5. Hill, J., Horton, M., Kling, R., and Krishnamurthy, L. The platforms enabling wireless sensor networks. *Commun. ACM 47*, 6 (June 2004), 41–46.
6. ITU. *Measuring the Information Society.* International Telecommunication Union. Geneva, Switzerland, 2010.
7. *Mobile Location Based Services: Applications, Forecasts & Opportunities.* Juniper Research Ltd, Basingstoke, Hampshire, U.K., 2010.
8. Modeling human mobility (ACM News). *Commun. ACM* Web site (Apr. 28, 2010).
9. Mitchell, T.M. Mining our reality. *Science 326* (2009), 16441645.
10. Ozer, N.A., Conley, C., O¹Connell, H., Ginsburg, E., and Gubins, T. *Location-Based Services: Time for a Privacy Check-In.* American Civil Liberties Union (ACLU), San Francisco, CA, 2010.
11. Singel, R. U.S. cell-phone tracking clipped. *Wired* (2005); http://www.wired.com/news/print/0,1294,69390,00.html.
12. Song, C., Qu, Z., Blumm, N., and Barabási, A.-L. Limits of predictability in human mobility. *Science 327* (2010), 10181021.
13. Thompson, D. Google's CEO: "The laws are written by lobbyists." *The Atlantic* (Oct. 1, 2010).

Daniel Soper (dsoper@fullerton.edu) is a member of the faculty of the information systems and decision sciences department at California State University, Fullerton.

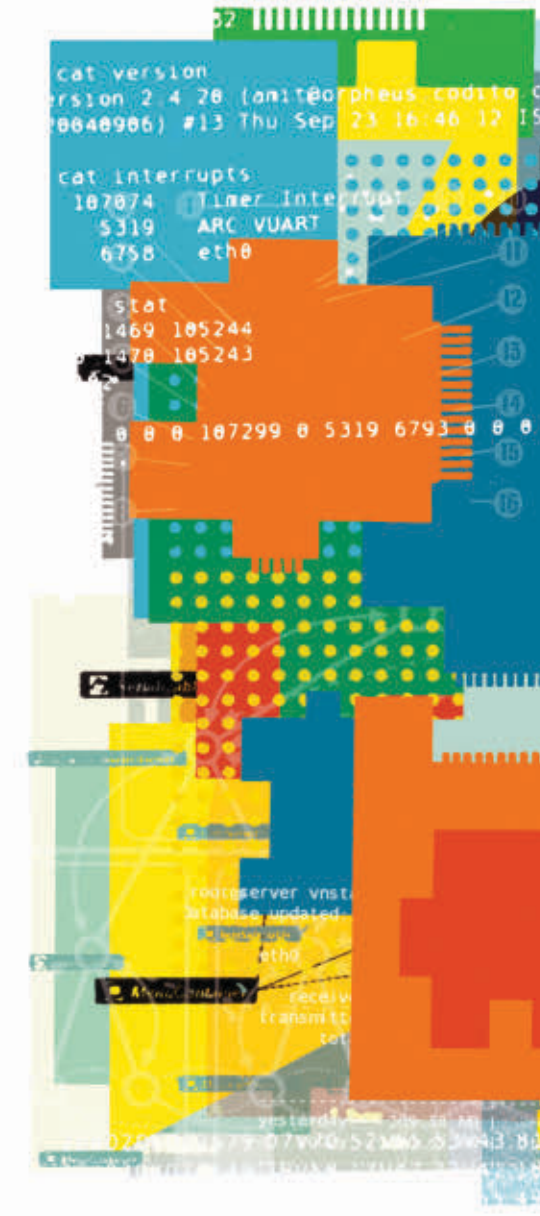
DOI:10.1145/2133806.2133820

Article development led by acmqueue
queue.acm.org

The benefits of composability are becoming clear in software engineering.

BY BRIAN BECKMAN

Why LINQ Matters: Cloud Composability Guaranteed



COMPOSABILITY IS AN aspect of component design that addresses the freedom to select and combine generic components in nearly arbitrary configurations to support a wide variety of applications, even applications that were not anticipated by the designers of the components. For example, electrical components are routinely designed so that switches, junctions, and loads can be configured in almost any order to fulfill an enormous variety of applications. Component designers do not need to know the specifics of an application, and component users are not overly constrained by arbitrary choices made by designers. Similarly, in mechanical engineering many applications can be built entirely of generic brackets, hinges, fasteners, and so on, designed at the outset to be configured in any order and layout made possible by standardization of sizes, screw pitches, strength-of-materials, among others.

In this article we use Language-integrated Query (LINQ) as the guiding example of composability. LINQ is a specification of higher-order operators designed specifically to be composable. This specification is broadly applicable over anything that fits a loose definition of "collection," from objects in memory to asynchronous data streams to resources distributed in the cloud. With such a design, developers build up complexity by chaining together transforms and filters in various orders and by nesting the chains—that is, by building expression trees of operators.

Encoding and transmitting such trees of operators across tiers of a distributed system has many specific benefits, most notably:

- Bandwidth savings from injecting

ILLUSTRATION BY ALEX WILLIAMSON

filters closer to producers of data and streams, avoiding transmission of unwanted data back to consumers.

▸ Computational efficiency from performing calculations in the cloud, where available computing power is much greater than in clients.

▸ Programmability from offering generic transform and filter services to data consumers, avoiding the need for clairvoyant precanning of queries and data models at data-producer sites.

This article also covers lifting composability from programs into the cloud via REST (representational state transfer), mapping between expressions in programs and resource specifications in URIs; and it addresses the subtle hazards of designing for composability: seemingly unimportant, arbitrary choices can render a design fundamentally uncomposable.

Meta-Design Principle

Designing for composability is not *merely* separating interface and implementation. It also means a certain uniformity in the interfaces: *a meta-design principle*. It wouldn't do much good if every electrical socket-and-plug design were of a custom and idiosyncratic shape, even if the design were sufficient to carry the required current. It is precisely because sockets and plugs are standardized in shape that we get the flexibility to combine them in nearly arbitrary configurations.

The physical engineering disciplines (for example, electrical and mechanical) have a well-established history of designing for composability because the benefits are so obvious and overwhelming, but the benefits are becoming increasingly clear in software engineering as well. Software designers have long accepted blackboxing: the need to encapsulate implementation details inside interfaces that expose minimal, precise, complete contracts. Designers are now beginning to abstract over the interfaces themselves and to realize the combinatorial benefits of composability.

The discipline of pure functional programming, exemplified by the programming language Haskell (http://www.haskell.org) and its predecessor Miranda (http://en.wikipedia.org/wiki/Miranda_programming_language), brought the absolute composability guarantees of mathematical functions into the world of programming. Their influence, combined with imperative and object-oriented programming traditions, represents a recent, deep, and important consolidation: one big concept, *function*, incorporates several

smaller concepts as cases. Relations, objects, states, and streams all have natural representations in functional style. The increasing favor of this style is evidenced by the following:

- The incorporation of higher-order functions (http://en.wikipedia.org/wiki/First-class_function; http://en.wikipedia.org/wiki/Higher-order_function) for transforming, filtering, and aggregating data into workaday imperative programming languages, most notably C# and JavaScript.
- The general recognition of immutability as a benefit, especially in concurrent and distributed systems; see Eric Lippert's comments (http://blogs.msdn.com/b/ericlippert/archive/2007/11/13/immutability-in-c-part-one-kinds-of-immutability.aspx) and a relevant thread in the Python user's forum (http://groups.google.com/group/comp.lang.python/browse_thread/thread/29c62cbee7a6b598/df5b676f6f695eb9).
- The spread of libraries such as LINQ (http://en.wikipedia.org/wiki/Language_Integrated_Query) and underscore.js (https://github.com/documentcloud/underscore).
- Whole languages such as Scala, F#, and Clojure.
- Language overlays such as coffeescript (https://github.com/jashkenas/coffee-script).

LINQ as a Multidomain Composability Pattern

LINQ is a case study in taking the composability of functions one step further by specifying a collection of composable higher-order functions—the SQOs (standard query operators; http://en.wikipedia.org/wiki/Language_Integrated_Query#Standard_Query_Operators). This specific design generalizes over a loose abstraction of a collection and covers a comprehensive variety of domains:

- Data structures in memory (for example, lists, trees, graphs, queues).
- Tables in a database (even with primary- and foreign-key constraints).
- Asynchronous data streams (RX; http://msdn.microsoft.com/en-us/data/gg577609).
- Slash-separated terms in URIs; thus, resources in the cloud.
- Signal-processing primitives (convolutions and Fourier transforms).
- CodeDOMs (code document object models); expression trees; ASTs (abstract syntax trees)—that is, over programming languages themselves.
- HTML, XML, and JSON (JavaScript Object Notation) documents.
- Continuations, exceptions, alternatives, and more.

This design brings the same thorough collection of composable operators to all these domains. You could say that LINQ brings *composability itself* to these domains. If you implement the SQOs, then you are guaranteed composability. LINQ was not the first and certainly is not the only way to do it, but it is exemplary enough to support the other observations in this article, namely:

- Lifting composability in programs to composability in the cloud.
- Illustrating the hazards facing would-be composable designs.

From Programs to the Cloud

LINQ's SQOs, then, serve as our exemplar of the composability-in-itself meta-design principle. As typically implemented in programs, the SQOs apply functions to collections in various ways, recognizing *functions as the natural interaction convention between composable components in programs*.

Leaping from composability within programs to composability in the cloud, you must "implement" SQOs over *the natural interaction convention between composable components in the cloud*, whatever that may be. This is just another domain for LINQ, viewed as a specification for composable operators over anything that satisfies LINQ's loose abstraction of a collection. The cloud is just a loose collection of resources specified by URIs.

What is "the natural interaction convention between composable components in the cloud?" Perhaps the most important such interaction convention today is REST (http://en.wikipedia.org/wiki/REST), in which client and server interact via HTTP, by opaque URIs that represent *resources*. REST specifies no semantics on URIs, only a very light postfix syntax of slash-separated terms. Protocols such as OData (http://www.odata.org/) use URI encoding and REST to provide data frameworks to the cloud, specifically by empowering a client to specify the result of a desired computation or query as a resource. To this let's add the following ideas:

- **Radical composability.** Users build complexity via chaining sequences of SQO expressions in arbitrary order rather than via a rich syntax over a presumed data model, as in OData.
- **Bidirectional mapping.** This is mapping between expression chains in programs and resource specifications in URIs. Given such a mapping, you may reason over expressions in programs and resources in cloud interactions as if they were the same.
- **Injectability.** The receiver of an expression chain may maintain standing operations over multiple communications so as to save bandwidth. A client requesting, say, a stream of "Chinese restaurants near me," may inject a filter one time to the server, which then avoids sending other irrelevant business records to the client over future sequences of push notifications. A server, flooded by too-frequent location updates from a client, may inject a filter into the client that says "only every tenth update," which the client applies just before performing physical communication.
- **Broad applicability.** The same composable design works over static data resources and time-dependent, asynchronous data streams (and other domains).

At this point, assume a natural correspondence between composable operator-chain expressions in programs and composable resource specifications in the cloud. A specific way to mechanize that correspondence is pre-

Figure 1. LINQ expressions including a filter and a projection.

```
Customers
    .Where(customer =>
        (DateTime.Now - customer.Orders.Last().dateTime)
            < TimeSpan.FromDays(365))
    .Select(customer => customer.PrimaryContactPhone)
    ;
```

sented in the following sections, but the correspondence itself allows you to reason the same way over what works in programs and what works in the cloud.

Embedding LINQ in URI Syntax

The typical implementation of LINQ is embedded in a programming language that supports higher-order functions. Adopting LINQ, however, only means accepting the specification of the SQOs. LINQ, as a design, does not need an ordinary programming language at all. By noting that a nested tree of operators can be converted to a pure-postfix *chain* of operators; and a chain of operators separated by dots in a programming language is formally isomorphic to a chain of terms separated by slashes in a URI, you can embed any LINQ chain in URI syntax and use HTTP as an "expression-embedding language" in RESTful Web services.

For example, imagine the expression illustrated in Figure 1, in pseudo C#, that represents the phone numbers of customers with recent purchases:

It says, "Keep the customers `Where` the difference between the date and time now and the date and time of the last purchase is fewer than 365 days, then `Select` the primary phone contact number from each customer in the resulting collection of customers." `Where` and `Select` are both LINQ SQOs. Each one takes two arguments—a data collection argument to the left of each dot and a higher-order function inside parentheses—and each produces a new collection. That is the essence of their chain composability: each SQO expression represents a data collection ready to be dotted into the next SQO down the chain in left-to-right order.

The higher-order-function argument of the `Where` SQO is a predicate written as a lambda expression:

```
customer => …function-body
expression depending on
customer…
```

Read this as "the function of *customer* that performs the computation specified by the function-body expression and produces the resulting value of the expression." Since this particular lambda expression represents a predicate, it should produce a Boolean value. In JavaScript, you would write exactly the same thing as

```
function(customer) {return …
function-body expression…;}
```

The higher-order-function argument of the `Select` SQO is another lambda expression that just picks out the phone-number field from a customer record, but it could do arbitrary computation.

Most languages or libraries call this `Select` operator map, but LINQ's designers chose the name `Select` to appeal to SQL developers. In some theoretical discussions, the operator is also called `Project`. Let's stick with `Select` for now.

Since the expression chain contains no nested operators, it is already in *shallow postfix* form if we consider the lambda expressions atomic:

```
Customers.Where(…lambdaW…)
.Select(…lambdaS…)
```

Figure 2. AST showing imploded lambdas.

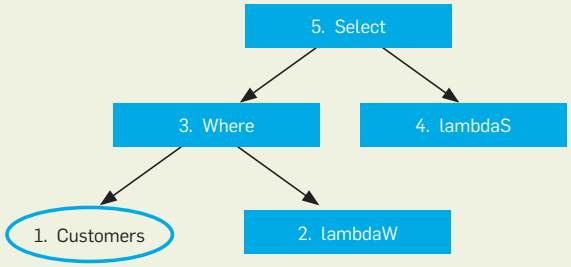
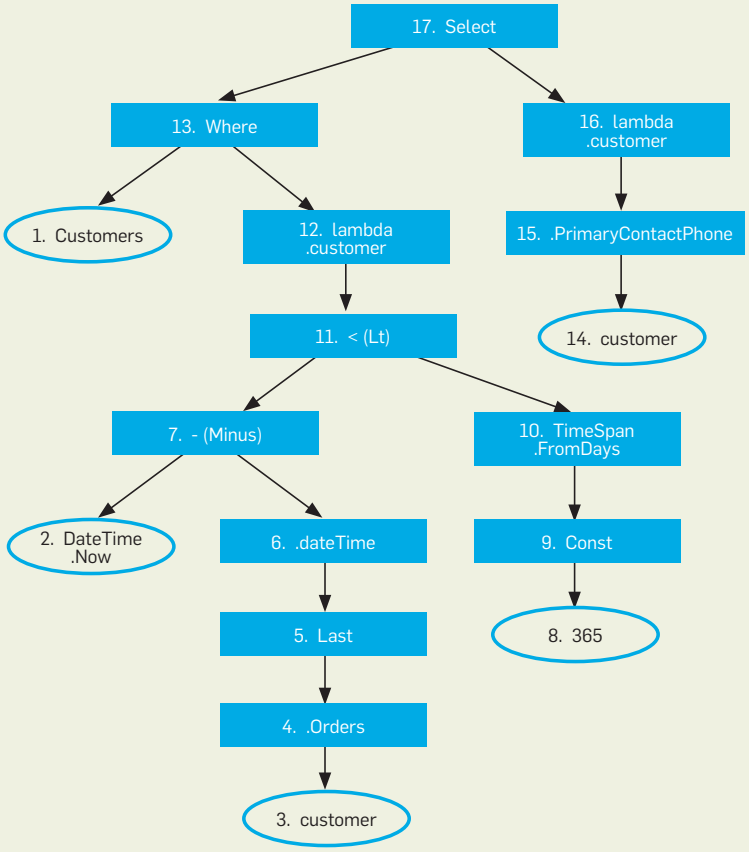


Figure 3. AST with exploded lambdas.



Just prepend a protocol and domain, and replace the dots with slashes:

```
https://myQueries.com/Customers/
Where(…lambdaW…)/Select(…lambdaS…)
```

Then URL-encode the lambdas, ending up with something like this:

```
https://myQueries.com/Customers/
Where(customer%3D%3E(DateTime.
Now-customer.Orders.Last().
dateTime)%3CTimeSpan.FromDays(365))/
Select(customer%3D%3Ecustomer.Pri-
maryContactPhone)
```

Thus, you have a representation of the entire expression in a URI. A RESTful server can interpret expressions such as this in a security sandbox and provide a very general query service without having built-in, precanned queries.

If you cannot or do not want to consider the lambdas as atomic, then you can go all the way to *deep postfix* form, writing the entire AST in postfix and encoding it in a URI.

You can do this in two stages: first keep the lambdas imploded so you can see the difference between deep prefix and shallow prefix; then explode the lambdas. Begin by rewriting the query first in prefix form, dragging everything that was on the left of a query-operator dot over to the right inside the parentheses of the operator in first position. This turns the expression inside out like the sleeve of a sweater, as follows:

```
Select(Where(Customers, …lamb-
daW…), …lambdaS…)
```

Now, it is easy to see that each level is a binary operator with a data collection in the first argument slot and a lambda in the second slot. This corresponds to the AST in Figure 2. Without exploding the lambdas there, left-to-right, depth-first traversal yields a deep postfix form, still with lambdas imploded:

```
https://myQueries.com/Custom-
ers/2.lambdaW/Where/4.lambdaS/
Select
```

It looks similar to the shallow postfix form, just with the SQOs *post-lambda*, as it were. The SQOs appear after the lambdas instead of hosting lambdas inside their parentheses. In deep postfix form, there are *never* any parentheses, and that's the whole point. This should remind you of PostScript or Forth or RPN (reverse Polish notation) calculators, because they are all deep postfix (http://en.wikipedia.org/wiki/Concatenative_programming_language).

Now, explode out the lambdas into the AST shown in Figure 3. With no parentheses, its RESTful encoding in a URI is trivial:

```
https://myQueries.com/Customers/
DateTime.Now/customer/.Orders/
Last/.dateTime/-/365/Const/TimeS-
pan.FromDays/Lt/lambda.customer/
Where/customer/.PrimaryContact-
Phone/lambda.customer/Select
```

The remaining dots denote property-accessor calls, system calls, or lambda-variable declarations, as opposed to SQO calls. URI encoding replaced all dots preceding SQO calls with slashes.

Note that there are other options for encoding lambdas in postfix. This example uses a representation in which variables appear before they are declared. This requires an execution model that saves variables and dependent operations symbolically on the stack, to be resolved later when the lambda term declaring the variable arrives. Other competent choices abound. For example, PostScript and Factor encode lambda expressions in arrays where the variable declarations precede the function bodies. This amounts to a cheater prefix notation embedded in otherwise postfix languages.

We have departed the realm of human-readable syntax (except for lovers of Forth) but have found a URI encoding of ASTs that is trivial for machines to read and write. We probably do not need to go quite this far, even for nested LINQ. Parenthesis counting can just as easily manage nesting while retaining human readability.

Figure 4. Example LINQ expression with nested operators.

```
Customers
    .Where(customer => customer.Orders
        .SelectMany(order => order.LineItems)
        .Sum() > 1000)
    .SelectMany(customer => customer.Orders)
    .SelectMany(order => order.LineItems)
    ;
```

Save some ink by replacing lambda variable `customer` with the shorter `c` and lambda variable `order` with the shorter `o`. You can write the URI form without hesitation:

```
https://myQueries.com/Customers/Where(c%3D%3Ec.Orders/SelectMany(o%3D%3E
o.LineItems)/Sum()%3E1000)/SelectMany(c%3D%3
Ec.Orders)/SelectMany(o%3D%3Eo.LineItems)
```

Then you can convince yourself that parenthesis counting reveals the nesting in this and every case.

Figure 5. Example that produces residual unwanted nesting.

```
Customers
    .Where(customer => customer.TotalSpending() > 1000)
    .Select(customer => customer.Orders)
    ;
```

Figure 6. Example of *SelectMany* that avoids unwanted nesting.

```
Customers
    .Where(customer => customer.TotalSpending() > 1000)
    .SelectMany(customer => customer.Orders)
    ;
```

For an example with nested SQOs, imagine the expression in Figure 4, again in pseudo C#, that represents the line items for high-value orders from a list of customers.

Hazards in Designing for Composability

Whereas LINQ is a fine example of a composable design that works equally as well over objects in memory as over resources in the cloud, it's certainly not the only one. Designers who embrace radical composability, however, will confront some hazards. If they get something just slightly wrong in the design, then users can no longer specify what they need within the design.

In a typical example, suppose you need a collection of all orders from big-spending customers. Your library provides a way to filter out low-spending customers, and it provides a *composable* way to get the orders for each of those customers. So far, so good. In pseudo-C#, you might try the example shown in Figure 5.

This does not yield the required collection of orders, however; instead it yields a collection of collections of orders—one internal collection of orders for each customer. You need to flatten the top-level collection. You must leave the library and write custom code.

In a cloud setting, leaving the library means proliferating custom code for every uncovered case, restricting the ability to build novel expressions just from composable building blocks. Anywhere you have an expression *like* the one here, you must do something out-of-band to remove the unwanted extra structure.

In a program setting, leaving the library means risking insertion of brittle workaround code in various places in the system. Perhaps the programmer knows that `Orders` are arrays, so writes block-copy operations to flatten them. Later, when `Orders` changes implementation from array to gzipped XML, this code fails unexpectedly.

If, however, the library had provided the required `SelectMany` in the first place, then the programmer would have written as shown in Figure 6, which has only one difference from the previous code (underlined).

It is somewhat amazing that a single specification of operators such as LINQ's SQOs applies equally as well to asynchronous data streams as to objects in memory—*but it does*.

Breadth and Depth of Composability

It is somewhat amazing that a single specification of operators such as LINQ's SQOs applies equally as well to asynchronous data streams as to objects in memory—*but it does*. Users can specify arbitrary transformations not only on pulled data resources, but also on pushed asynchronous data streams using the same set of SQOs. In both cases, expression components can be transparently injected upstream to the data-producer machines, whether client or server, transforming and filtering data at the headwaters for bandwidth savings and reduction of "chattiness."

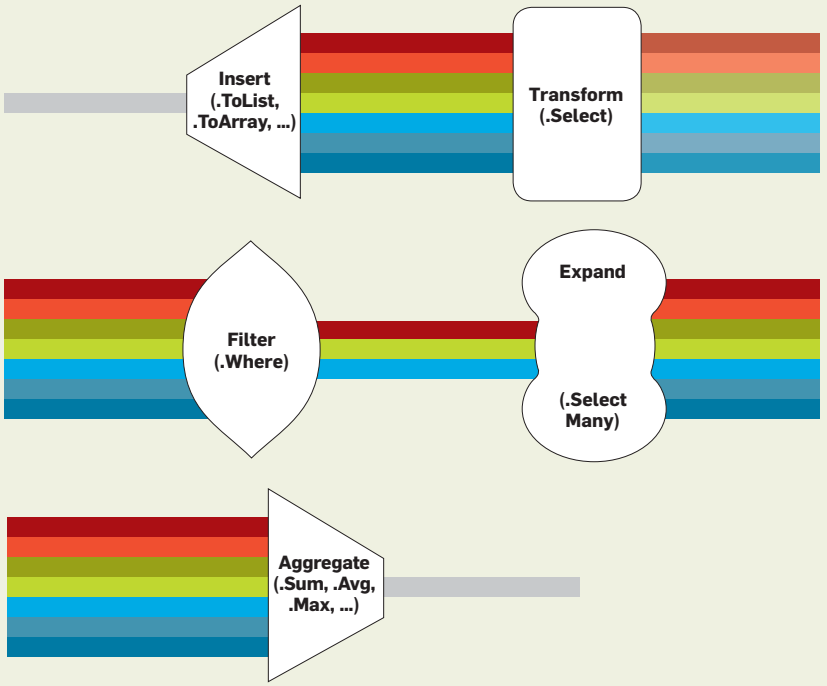
In a kind of self-referential joke, however, the same set of SQOs that works over functions in a program works over resources distributed in the cloud. Thus, there is an octuplet application of the same SQO design:

- SQOs operating on data within programs or on data over the cloud.
- Data distributed in space or with the data distributed in time.
- Transforms and filters executed at the data-producer site versus at the data-consumer site, respecting concerns such as bandwidth, latency, and security.

LINQ borrowed its essential concepts once again from Haskell, specifically from its initialization library called the Prelude. The essence of the technique is *a loose abstraction of a collection* of things—for example (don't yawn), customers, orders, items. Each customer has a number of orders, and each order has a number of items. From the point of view of the pattern, it does not matter whether these customers are in memory, or delivered *n* at a time to callbacks, or threaded through continuations, or in offline document stores accessible by Web-service calls. What matters is just that there is an abstraction representing them somewhere in your system where you want to manipulate them. The required composable operators fall into categories (with examples from LINQ and the Haskell Prelude):

- INSERT elements into collections (for example, `new List(...)` / `return`). This is how you get into a collection. For example, convert a customer record into a one-record table; make a

Figure 7. Categories of composable operators.



singleton list (containing a single customer object).

▸ TRANSFORM elements and produce a new collection (such as, `.Select` / `map`). For example: produce a list of customers' primary phone numbers from a list of customers, one phone number for each customer.

▸ FILTER elements out of the collection based on a predicate (such as, `.Where` / `filter`). For example, produce a list of customers with big orders; there will be fewer of these, in general, than in the original collection.

▸ EXPAND elements from one collection into new subcollections and combine or concatenate the new collections (such as, `.SelectMany` / `concatMap`, `bind`, `>>=`). For example, get all first-degree friends of the customers; get all orders from all customers; get all line items from all orders.

▸ AGGREGATE elements and produce a result that depends on all the elements (such as, `.Aggregate` / `fold`; and `.Take` and `.Skip` / `take`, `drop` also fall in this category). This is how you get out of the collection (technically, this is a co-monadic operator, EXTRACT). For example, get the average spending over all customers.

These are fundamental. If a collection abstraction supports appropriate primitives in each of these categories, then composability is guaranteed.

How to Ruin a Library

You do not have to write *exactly* the LINQ SQOs; there is plenty of freedom if you understand the fundamentals. One big way to blow it, however, is to overlook the EXPAND category, which designers often do because it does not fit the map/filter/fold mantra familiar in some quarters. The EXPAND category is essential; in fact, it is the most important one. Without it, the library can not represent the most general kind of collection and thus can not be extended to new settings such as asynchronous data streams or resources in the cloud. With it (and with INSERT), all the other categories can be simulated exactly (for more information, see http://channel9.msdn.com/Shows/Going+Deep/Bart-De-Smet-MinLINQ-The-Essence-of-LINQ).

The diagrams in Figure 7 demonstrate the categories of operator. Together, the diagrams show why the EXPAND category is most essential. It is the necessary twin of FILTER, even though it is more general.

Another way to ruin a library is to get the order of arguments wrong. To chain operators, let alone to embed them in URIs and REST, you want the collection argument in first position, always. Traditionally, dialects of Lisp put the *function* argument first, at least for `map`, because it makes expressions read, "Map this function over that collection." Library designers with Lisp backgrounds might just reflexively write `map` that way. This minor syntactic lapse means that `map`, a.k.a. `Select`, disrupts the pattern and cannot be composed in dotted chains except at the beginning.

Conclusion

LINQ's pattern is bulletproof and pervasive, applying to a surprisingly broad array of software domains such as asynchronous streams, stateful computations, I/O, exceptions, alternatives—anything that fits a loose abstraction of "collection of things." The pattern is always composable in *expression trees*, where dependent operators follow each other in nested sequences. It can be encoded equally as well in ordinary programming-language syntax as in pure-postfix notations such as URIs.

Because the cloud fits the loose definition of a "collection of things," LINQ's pattern of composable transforms covers distributed resources in the cloud as well as it covers locally addressable resources in programs. Packaging subexpressions and injecting them closer to data-production sites allows you to gain bandwidth, chattiness, and security benefits in RESTful Web services. C

Related articles on queue.acm.org

The World According to LINQ
Erik Meijer
http://queue.acm.org/detail.cfm?id=2024658

Securing Elasticity in the Cloud
Dustin Owens
http://queue.acm.org/detail.cfm?id=1794516

Why Cloud Computing Will Never Be Free
Dave Durkee
http://queue.acm.org/detail.cfm?id=1772130

Brian Beckman currently works in Microsoft's Bing on Maps and Signals. He has held many positions at Microsoft since 1992, from Crypto (SET) to Biztalk to research in functional programming. He wrote the first version of the Time Warp Operating System on the Caltech Hypercube 1984–1989.

DOI:10.1145/2133806.2133821

Article development led by acmqueue
queue.acm.org

A taxonomy of tools that support the fluent and flexible use of visualizations.

BY JEFFREY HEER AND BEN SHNEIDERMAN

Interactive Dynamics for Visual Analysis

THE INCREASING SCALE and availability of digital data provides an extraordinary resource for informing public policy, scientific discovery, business strategy, and even our personal lives. To get the most out of such data, however, users must be able to make sense of it: To pursue questions, uncover patterns of interest, and identify (and potentially correct) errors. In concert with data-management systems and statistical algorithms, analysis requires contextualized human judgments regarding the domain-specific significance of the clusters, trends, and outliers discovered in data.

Visualization provides a powerful means of making sense of data. By mapping data attributes to visual properties such as position, size, shape, and color, visualization designers leverage perceptual skills to help users discern and interpret patterns within data.[4] A single image, however, typically provides answers to, at best, a handful of questions. Instead, visual analysis typically progresses in an iterative process of view creation, exploration, and refinement. Meaningful analysis consists of repeated explorations as users develop insights about significant relationships, domain-specific contextual influences, and causal patterns. Confusing widgets, complex dialog boxes, hidden operations, incomprehensible displays, or slow response times can limit the range and depth of topics considered and may curtail thorough deliberation and introduce errors. To be most effective, visual analytics tools must support the fluent and flexible use of visualizations at rates resonant with the pace of human thought.

The goal of this article is to assist designers, researchers, professional analysts, procurement officers, educators, and students in evaluating and creating visual analysis tools.

We present a taxonomy of interactive dynamics that contribute to successful analytic dialogues. The taxonomy consists of 12 task types grouped into three high-level categories, as shown in the accompanying table: data and view specification (visualize, filter, sort, and derive); view manipulation (select, navigate, coordinate, and organize); and analysis process and provenance (record, annotate, share, and guide). These categories incorporate the critical tasks that enable iterative visual analysis, including visualization creation, interactive querying, multiview coordination, history, and collaboration. Validating and evolving this taxonomy is a community project that proceeds through feedback, critique, and refinement.

Taxonomy of interactive dynamics for visual analysis.

Data and View Specification	**Visualize** data by choosing visual encodings.
	Filter out data to focus on relevant items.
	Sort items to expose patterns.
	Derive values or models from source data.
View Manipulation	**Select** items to highlight, filter, or manipulate them.
	Navigate to examine high-level patterns and low-level detail.
	Coordinate views for linked, multidimensional exploration.
	Organize multiple windows and workspaces.
Process and Provenance	**Record** analysis histories for revisitation, review, and sharing.
	Annotate patterns to document findings.
	Share views and annotations to enable collaboration.
	Guide users through analysis tasks or stories.

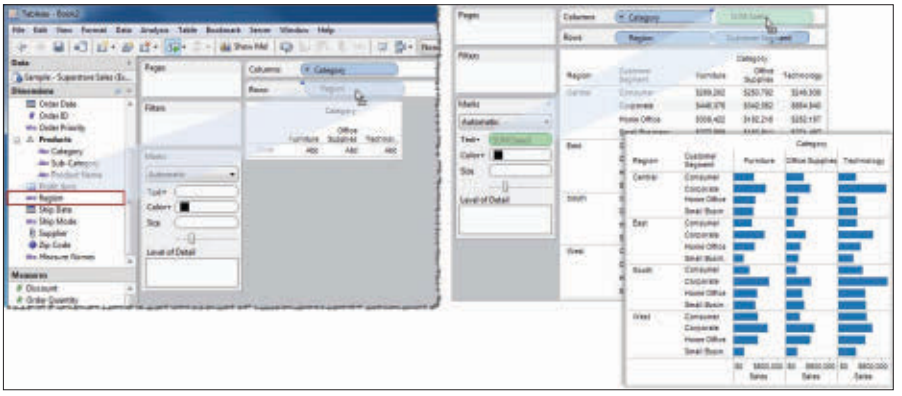


Figure 1. Visual encoding via drag-and-drop actions in Tableau.

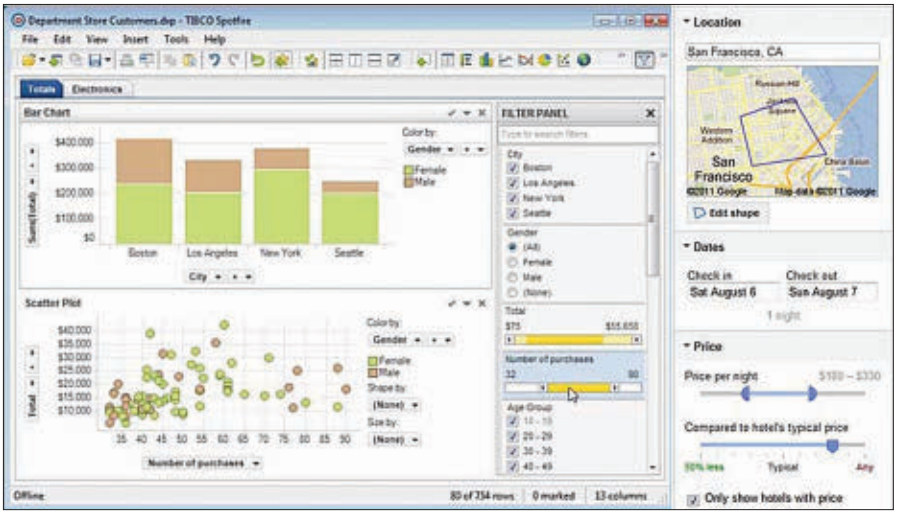


Figure 2. Examples of dynamic query filter widgets from Spotfire (left) and Google Hotel Search (right).

Our focus on interactive elements presumes a basic familiarity with visualization design. The merits and frailties of bar charts, scatter plots, timelines, and node-link diagrams, and of the visual encoding decisions that underlie such graphics, are certainly a central concern, but we will largely pass over them here. A number of articles and books address these topics in great detail,[4,5,20] and we recommend them to interested readers.

Within each branch of the taxonomy, we describe example systems that exhibit useful interaction techniques. To be clear, these examples do not constitute an exhaustive survey; rather, each is intended to convey the nature and diversity of interactive operations. Throughout the article the term *analyst* refers to someone who uses visual analysis tools and not to a specific person or role. Our notion of analyst encompasses anyone seeking to understand data: traditional analysts investigating financial markets or terrorist networks, scientists uncovering new insights about their data, journalists piecing together a story, and people tracking various facets of their lives, including blood pressure, money spent, electricity used, or miles traveled.

Data and View Specification

To enable analysts to explore large datasets involving varied data types (for example, multivariate, geospatial, textual, temporal, networked), flexible visual analysis tools must provide appropriate controls for specifying the data and views of interest. These controls enable analysts to selectively *visualize* the data, to *filter* out unrelated information to focus on relevant items, and to *sort* information to expose patterns. Analysts also need to *derive* new data from the input data, such as normalized values, statistical summaries, and aggregates.

Visualize. Perhaps the most fundamental operation in visual analysis is to specify a visualization of data: analysts must indicate which data is to be shown and how it should be depicted. Within user interfaces, such visualization "widgets" are often presented in a *chart typology*, a palette of available vi-

sualization templates (bar charts, scatter plots, map views.) into which analysts can slot their data. This method of interaction will be immediately familiar to users of spreadsheet programs: users select a chart type and assign data variables to visual aspects such as the X/Y axes and the size or color of visualized marks. A chart typology has the benefits of simplicity and familiarity, but it also limits the types of possible visualizations.

Visualization system designers have explored alternative approaches. Classic scientific visualization systems[1] use *data-flow graphs*, in which the visualization process is deconstructed into a set of finer-grained operators for data import, transformation, layout, or coloring. However, novel designs often require programming expertise to develop new operators for the system. Other systems are based on *formal grammars* that succinctly describe how data should be mapped to visual features. This approach is used by a number of popular data-visualization frameworks such as Leland Wilkinson's *Grammar of Graphics*,[24] ggplot2 for the R statistical analysis platform, and Protovis for HTML5. Formal grammars can be augmented with automated design facilities: a system can generate multiple visualization suggestions from a partial specification. Tableau (née Polaris[19]) enables visualization specification by drag-and-drop operations: analysts place data variables on "shelves" corresponding to visual encodings such as spatial position, size, shape, and color (see Figure 1). This specification is then translated into an underlying formal grammar that determines both the visualization design and corresponding queries to a database.

Fortunately, these methods are not mutually exclusive. Analysts can apply a data-flow system or formal grammar to define new components to include within a chart typology, leveraging the improved expressiveness of the former and the ease of use of the latter. Novel interfaces for visualization specification are still needed, as new tools requiring little to no programming might place custom visualization design in the hands of broader audiences.

Filter. Filtering of data values is intrinsic to the visualization process, as analysts rarely visualize the entirety of a data set at once. Instead, they construct a variety of visualizations for selected data dimensions. Given an overview of selected dimensions, analysts then often want to shift their focus among different data subsets—for example, to examine different time slices or isolate specific categories of values.

Designers have devised a variety of interaction techniques to limit the number of items in a display. Analysts might directly select (for example, "lasso") items in a display and then highlight or exclude them; we discuss these forms of direct view manipulation later. Another option is to use a suite of auxiliary controls, or *dynamic query widgets*,[17] for controlling item visibility (see Figures 2 and 3). The choice of an appropriate widget is largely determined by the underlying data type. Categorical or ordinal data can be filtered using simple radio buttons or checkboxes (when the number of distinct items is small), or scrollable lists, hierarchies, and search boxes with autocomplete (when the number of distinct items is large or contains arbitrary text). Ordinal, quantitative, and temporal data can also be filtered using a standard slider (for a single threshold value) or a range slider (for specifying multiple endpoints). When coupled with real-time updates to the visualization, these widgets allow rapid and reversible exploration of data subsets. In Figure 2, Spotfire

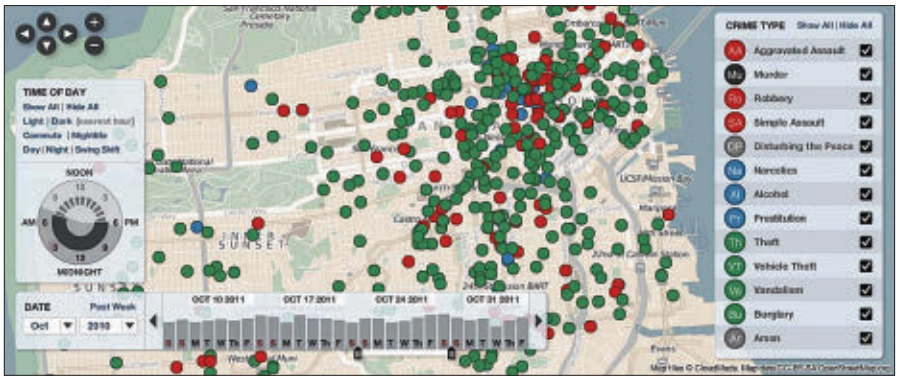


Figure 3. Zoomable map from CrimeSpotting.org.

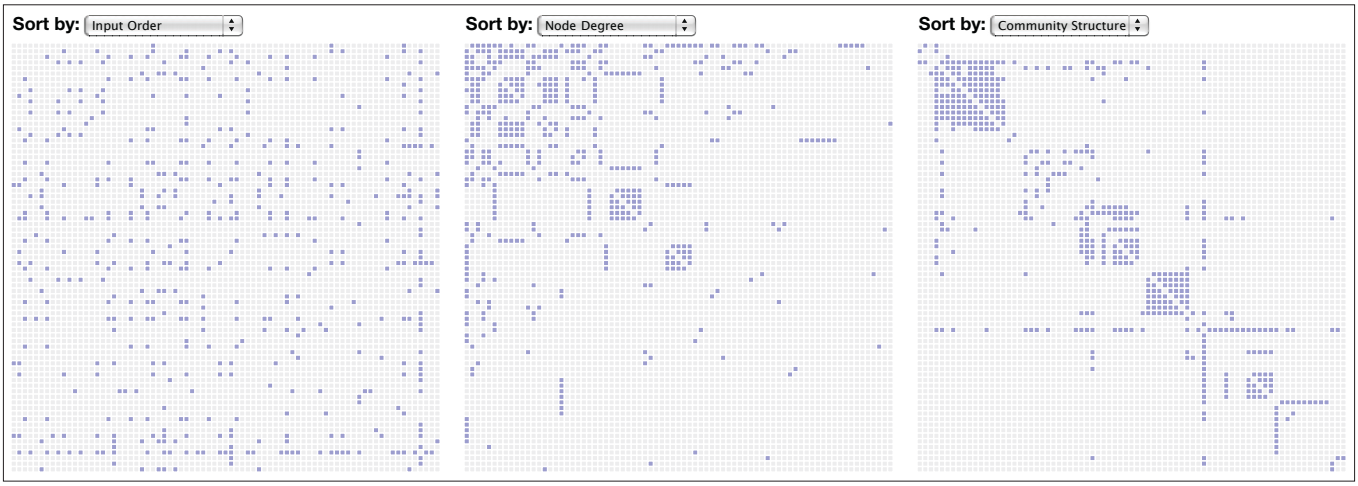
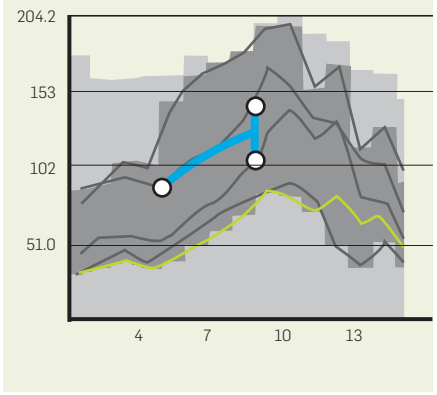


Figure 4. Reorderable matrices.

(left) provides a variety of controls for filtering visualized data: checkboxes and radio buttons filter categorical variables, while range sliders filter numerical values; on the right, Google Hotel Search provides widgets for geographic, date, and price ranges. Query controls can be further augmented with visualizations of their own: Figure 3 includes a range slider for dates augmented with a histogram of underlying values.

Expert analysts also benefit from more advanced functionality. For example, a search box might support sophisticated query mechanisms, ranging in complexity from simple keyword search to a full-fledged query language. Filtering also interacts with other operations: filtering widgets may operate over data sorted in a user-specified manner (see the next section), or users might create derived values (as we will discuss) and filter based on the results.

Figure 5. Querying time-series by slope in TimeSearcher.[12]



Sort. Ordering (or sorting) is another fundamental operation within a visualization. A proper ordering can effectively surface trends and clusters of values or organize the data according to a familiar unit of analysis (days of the week, financial quarters, and so on). The most common method of ordering is to sort records according to the value of one or more variables. Ordering becomes more complicated in the case of multiple view displays, in which both entire plots and the values they contain may be sorted to reveal patterns or anomalies. Sorting values consistently across plots (for example, by their marginal mean or median values) can reveal patterns while facilitating comparison among plots.

Some data types (for example, multivariate tables, networks) do not lend themselves to simple sorting by value. Such data may require more sophisticated *seriation* methods[24] that minimize a distance measure among items. The goal is to reveal underlying structure within the data. Figure 4 shows a matrix-based visualization of a social network. On the left, a matrix plot of a social network conveys little structure when the rows and columns (representing people) are sorted alphabetically. Interactively reordering the matrix by node degree reveals more structure (center). Permuting the matrix by network connectivity reveals underlying clusters of communities (right).

Derive. As an analysis proceeds in iterative cycles, users may find that the input data is insufficient: variables may need to be transformed or new attributes derived from existing values. Common cases include normalization or log transforms to enable more effective value comparisons. Derived measures are often used to summarize the input data, ranging from descriptive statistics (mean, median, variance) to model fitting (regression curves) and data transformation (group-by aggregation such as counts or summations). Often this functionality is provided via a *calculation language*, similar to those found in spreadsheets or database query languages.

Improved derivation methods present a promising frontier for visual analytics research. How can visual tools support flexible construction of more advanced models or derived values? Analysts might define patterns using programming-by-demonstration

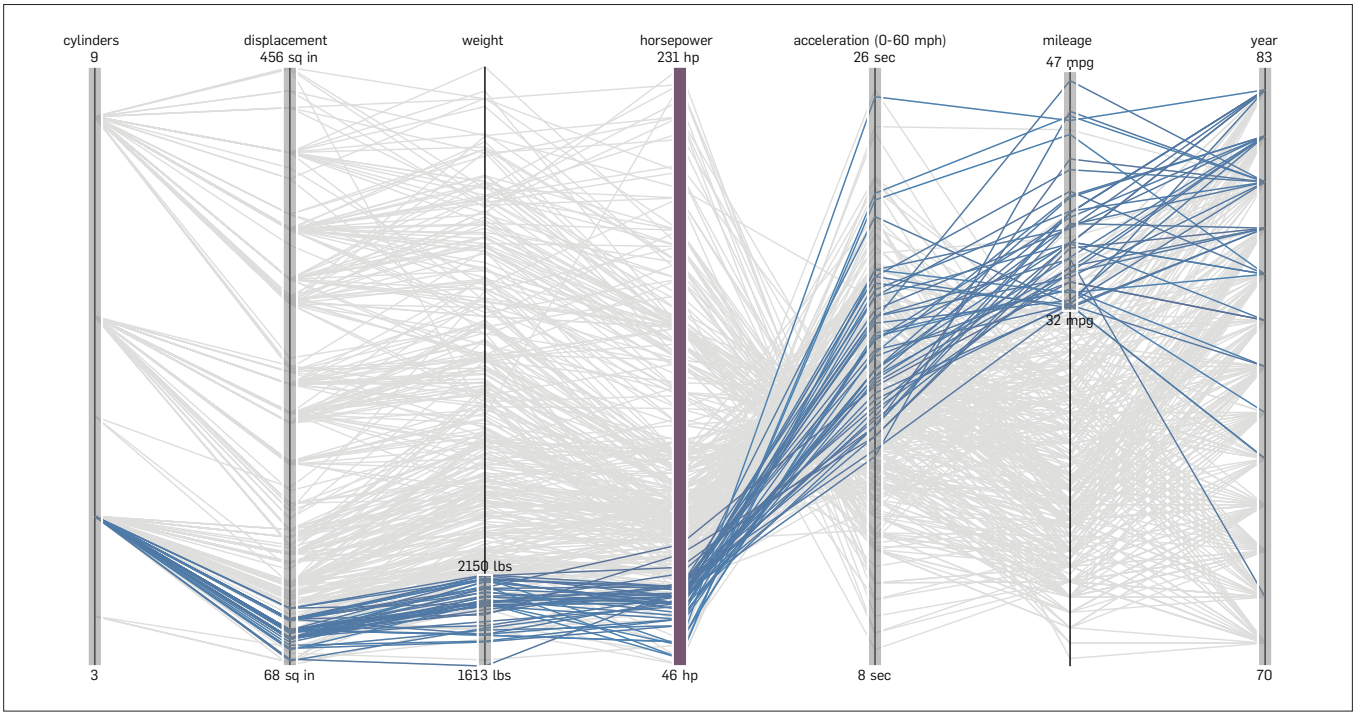


Figure 6. Selection queries in parallel coordinates.

methods. Or, visual tools might automatically fit applicable statistical models to the data based on the current visualization state. For example, the nesting of variables within common "pivot" displays could be mapped to the structure of a linear model. More principled frameworks that wed visualization to modeling and forecasting are still emerging.

View Manipulation

Once analysts have created a visualization, they should be able to manipulate the view to highlight patterns, investigate hypotheses, and drill down for more details. Analysts must be able to *select* items or data regions to highlight, filter, or operate on them. Large information spaces may require analysts to scroll, pan, zoom, and otherwise *navigate* the view to examine both high-level patterns and fine-grained details. Multiple, linked visualizations often provide clearer insights into multidimensional data than do isolated views. Analysis tools must be able to *coordinate* selections across multiple views and *organize* the resulting dashboards and work spaces.

Select. Pointing to an item or region of interest is common in everyday communication because it indicates the subject of conversation and action. In the physical world, people coordinate their gestures, gaze, and speech to indicate salient items. For example, different hand gestures can communicate angle (oriented flat hand), height (horizontal flat hand), intervals (thumb and index finger in "C" shape), groupings (circling a region), and forces (accelerating fist).[11] In visual analysis, reference (or *selection*) remains of critical importance, but is realized through a more limited set of actions, such as clicking or lassoing items of interest.

Common forms of selection within visualizations include mouse hover, mouse click, region selections (for example, rectangular and elliptical regions, or free-form "lassos"), and area cursors (for example, "brushes"[2] or dynamic selectors such as the bubble cursor,[7] which selects the item currently closest to the mouse pointer).

Selections can vary in terms of their expressive power. Most interfaces support selections of a collection of items. Though this approach is easy to implement, it does not allow analysts to specify higher-level criteria. A more powerful, albeit more complex, approach is to support selections as queries over the data. Maintaining query structure increases the expressiveness of visualization applications. For example, drawing a rectangle in a chart may specify a range query over the data variables represented by the x and y axes. The resulting selection criteria can then be saved and applied to dynamic data (updating items may enter or exit a query region) or to a completely different visualization. Examples include querying stock-price changes in TimeSearcher[12] (see Figure 5) and attribute ranges in parallel coordinates displays[13] (Figure 6). In Figure 5 an angular selection tool specifies a target slope (rate of change) and tolerance for a collection of stock prices. All time series with a similar slope over the queried time range are selected; shaded regions show envelopes of minimum and maximum values. The widget operates directly on the visualization: dragging the widget from left to right interactively queries other time windows. In Figure 6 parallel coordinates plot multidimensional data as line segments among parallel axes. Here, an analyst has dragged along the axes to create interactive selections that highlight automobiles with low weight and high mileage.

Designing more expressive selection methods remains an active area of research. Enhanced selections might incorporate data semantics (for example, values, hierarchy, or clusters) to enable guides or "snap-to" actions. Nuanced selections might be specified with more fluid gestures. Of course, selection need not be limited to the mouse and keyboard: input modalities such as touch and speech might enable new, effective forms of selection.

Navigate. How analysts navigate a visualization is in part determined by where they start. One common pattern of navigation adheres to the widely cited visual information-seeking mantra: "Overview first, zoom and filter, then details-on-demand."[18] Analysts may begin by taking a broad view of the data, including assessment of prominent clusters, outliers, and potential data-quality issues. These orienting actions can then be followed by more specific, detailed investigations of data subsets. A common example is geographic maps. The map in Figure 3 depicts criminal activity by time and region. It shows all crimes committed after dark during the last week of October 2011. Dynamic query widgets enable filtering by time of day (left), date span (bottom), and type of crime (right). Pan (drag) and zoom (buttons and scroll wheel) controls enable view navigation. As an analyst zooms in on the map, the circular crime markers gain detailed labels—a form of *semantic zooming*.

Of course, starting with an expansive overview is not always advisable. A legal analyst researching for an upcoming trial may be wise to forego an overview of the entire history of U.S. court decisions. Instead, the analyst might start with the legal decisions most relevant to the current case, perhaps determined by keyword search, and expand the investigation to other, cited decisions. This form of navigation can be summarized as "Search, show context, expand on demand."[21]

In either case, visualizations often function as manipulable *viewports* onto an information space. Common examples include scrolling or panning a display via scrollbars or mouse drag, and zooming among different levels using a zoom slider or scroll wheel (Figure 3). Zooming need not follow a strict geometric metaphor: semantic zooming[3] methods can modify both the amount of information shown and how it is displayed as analysts move among levels of detail.

To aid navigation further, researchers have developed a variety of *focus plus context* methods. These displays provide a detailed view of a high-interest data region while retaining surrounding context to help keep analysts oriented. A second key idea is the use of *overview and detail* displays. For example, a geographic visualization might include a large zoomed-in map (the detail), while a smaller, zoomed-out map includes a rectangle showing the position of the zoomed-in view within the broader terrain (the overview). In this case, the detail view provides the focus, and the overview provides context.

A different approach is to use *distor-*

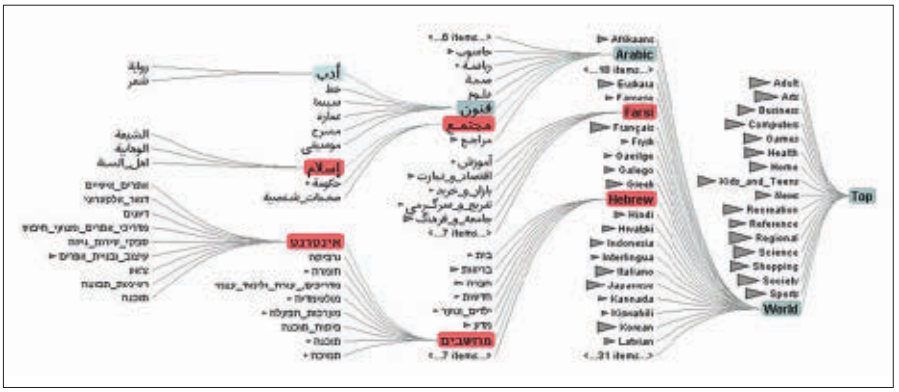


Figure 7. Degree-of-interest tree of a taxonomy with 600k items.[8]

tion techniques to demagnify contextual regions. A simple example is the Mac OS X dock, which uses 1D fisheye distortion to show common applications; more sophisticated methods employ distortion in multiple dimensions. While often visually intriguing, complex distortion methods have yet to prove their worth in real-world applications: viewers can become disoriented by nonlinear distortions, which show no significant performance improvement over simpler methods such as zooming.

In addition to manipulating display space, focus-plus-context methods can be applied directly to the data. The goal is to identify which data items are currently of high interest (focus), which are of high importance regardless of the current focus (context), and which can be safely removed from view. *DOI* (*degree-of-interest*) functions[8,21] calculate scores for information content based both on general importance (for example, top-level categories in a hierarchy or high-centrality nodes in a graph) and current interest (for example, as indicated by mouse clicks, search queries, or proximity to other high-interest items). The distribution of DOI scores can then be used to control the visibility of items based on the current view size and context of interaction, as in Figure 7. As analysts click on or search for different items, the DOI scores dynamically update to reveal relevant unseen data or hide irrelevant detail. A model of the analyst's current interest filters the display to the most relevant items. Low-interest items are elided but still accessible through aggregate representations. The interest estimates update as an analyst explores the taxonomy, initiating animated transitions between different views of the data.

Visualizations can also provide cues to assist analysts' decisions of where and how to navigate. An important challenge is to show selected items, even when they are not in view. For example, the results of a text search that are not currently in view might be shown by markers in the scrollbar or the periphery of the display.

Coordinate. Many analysis problems require *coordinated multiple* views that enable analysts to see their data from different perspectives. A public policy analyst studying educational attainment might produce a bar chart of people's ages, a map of locations, a textual list with education history, and a scatter plot showing income vs. education. By selecting a single item or a group in one view, analysts might see related details or highlighted items in the other views.

Multiview displays can facilitate comparison. For example, Edward Tufte[20] advocates the use of *small multiples*: a collection of visualizations placed in spatial proximity and typically using the same measures and scales. Small multiples enable rapid comparison of different data dimensions or time slices.

Alternatively, multiple view displays can use a variety of visualization types—such as histograms, scatter plots, maps, or network diagrams—to show different projections of a multidimensional data set. An analyst constructs a complex patchwork of interlinked tables, plots, and maps in Figure 8 to analyze the outcomes of elections in Michigan.[23] Annotations indicate how selected data items correspond between visualization views. Accompanying items such as legends, histogram sliders, and scrollbars with highlighting markers also provide views onto the data.

Multiview displays also enable multidimensional exploration. *Brushing and linking* is the process of selecting (brushing) items in one display to highlight (or hide) corresponding data in the other views.[2] In Figure 9, a baseball analyst makes selections in one plot and corresponding items highlight in the others. On the left, selecting high-income players (top-right plot) shows little dependence on career length or fielding ability, but correlates with hitting performance. On the right, selecting the cluster of players who make more assists than putouts (middle-left plot) reveals a strong dependence on position. Each visualization can thus serve as an input channel for revealing patterns across a data set. By allowing analysts to assess how patterns in one view project onto the others, linked selection enables multidimensional reasoning. Analysts may wish to coordinate views in variety of ways: selecting items in one view might highlight matching records in other views, or instead provide filtering criteria to remove information from the other displays. Linked navigation provides an additional form of coordination: scrolling or zooming one view can simultaneously manipulate other views.

Though comparing multiple visualizations requires viewers to orchestrate their attention and mentally integrate patterns among views, this process is often more effective than cluttering a single visualization with too many dimensions. Future studies of how analysts construct multiview displays and specify coordination behaviors (for example, highlighting, filtering) could provide designers with an understanding of how to build more effective tools. In addition, if designers ensure that rich multiview displays stay understandable, analysts are more likely to make compelling insights. Newcomers to an analysis, or even seasoned analysts simply returning from a coffee break, may become confused by the number of views and the potentially complicated set of coordinated queries between them. Vi-

sual analytics systems that provide access to coordination settings and replay the history of view construction can enhance understanding.

Organize. When analysts make use of multiple views they must manage a collection of visualizations. As in traditional window-based interfaces, analysts may wish to open, close, maximize, and lay out different components. As manual manipulation can be tedious, well-designed visual analytics tools simplify the organization of visualization views, legends, and controls. A tiled layout approach allows analysts with sufficiently large displays to see all the information and selectors at once, minimizing distracting scrolling or window operations. The coordination across windows means that slider movements or checkbox selections will cause all views to update, allowing rapid exploration.

Typical systems allow analysts to add views in ways that make modest changes to the existing window organization. An alternative approach is to add a new tab that contains a new plot, so analysts can switch between the first and second set of windows. A common feature is to add trellised views, so multiple visualizations can be created at once—for example, separate bar charts showing age distributions in different regions.

More advanced systems might aid this process through automated support that enables multiple windows to be opened/closed as a group and lays them out in orderly ways. Useful methods include standard scatter-plot matrices (showing all pairs of scatter plots) or custom generation of related views of interest (for example, of data variables correlated to the visualized attributes). Desirable features are automatic (re)sizing as views are added or removed and layout routines to place related views in spatial proximity.

As larger and multiple displays become more common, layout organization tools will become decisive factors in creating effective user experiences. Similarly, the demand for tablet and smartphone visualizations will promote innovation in layout organizations that are compact and reconfigurable by simple gestures. Zooming, panning, flipping, and sequencing strategies will also improve analyst experiences and facilitate effective presentations to others.

Process and Provenance

Visual analytics is not limited to the generation and manipulation of visualizations—it involves a process of iterative data exploration and interpretation. As a result, visual analytics tools that provide facilities for scaffolding the analysis process will be more widely adopted. Tools should preserve analytic provenance by keeping a *record* of analyst actions and insights so the history of work can be reviewed and refined. If analysts can *annotate* patterns, outliers, and views of interest, they can document their observations, questions, and hypotheses. Analysts should be empowered to *share* results and discuss with colleagues, coordinate the work of multiple groups, or support processes that may take weeks and months. Moreover, analysis tools can explicitly *guide* novices through common analysis tasks, provide progress indicators, or lead viewers through an analysis story.

Record. When analyzing data with visualizations, users regularly traverse the space of views in an iterative fashion. Exploratory analysis may result in a number of hypotheses, leading to multiple rounds of questions and answers. To support iterative analysis, visual analysis tools can record and visualize analysts' *interaction histories*. At a minimum, applications should provide basic undo and redo support. By modeling the space of user actions (view specifications, sorting, filtering, or zooming), rich logs can be constructed and visualized.[6,9] Common visual representations of analytic actions include both chronological ("timeline") and sequential ("comic strip") views. As shown in Figure 10, a "comic strip" display retraces the

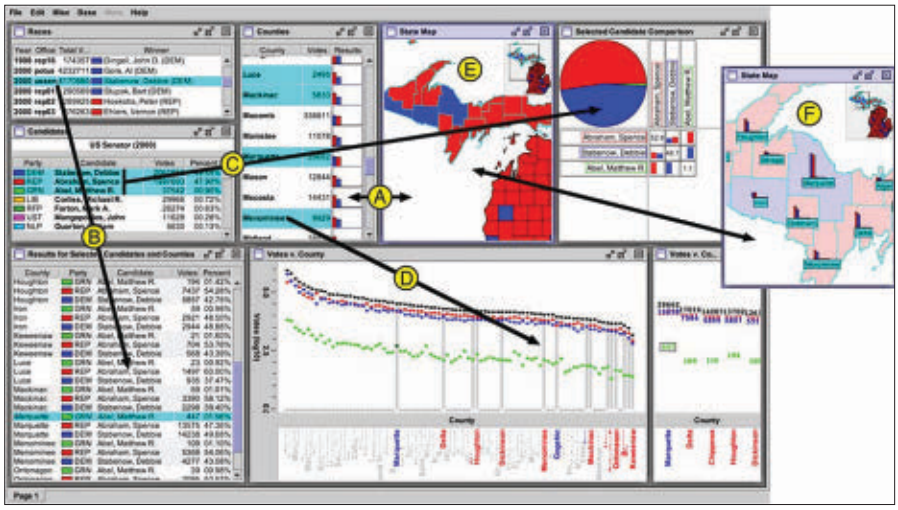


Figure 8. Multiple coordinated views in Improvise.[23]

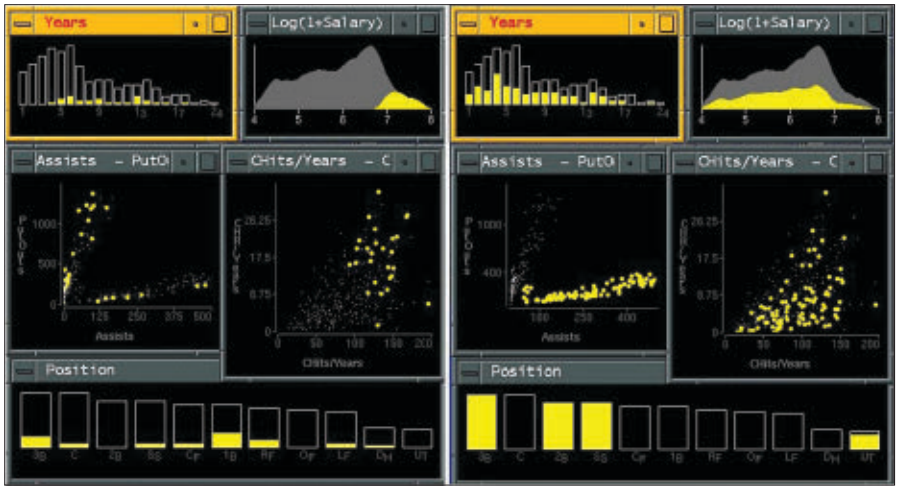


Figure 9. Brushing and linking of baseball statistics in GGobi.

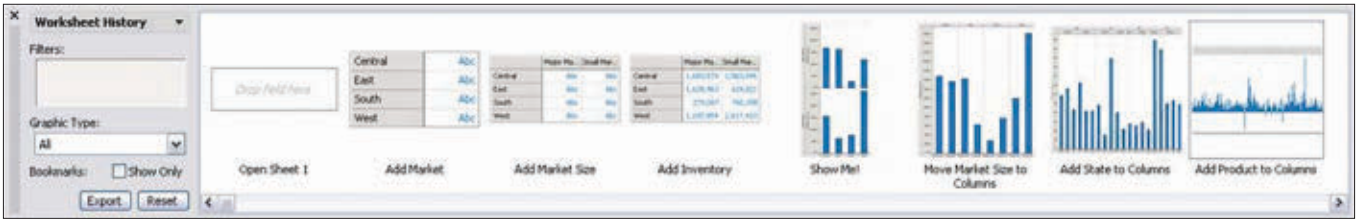


Figure 10. Visual analysis history.[9]

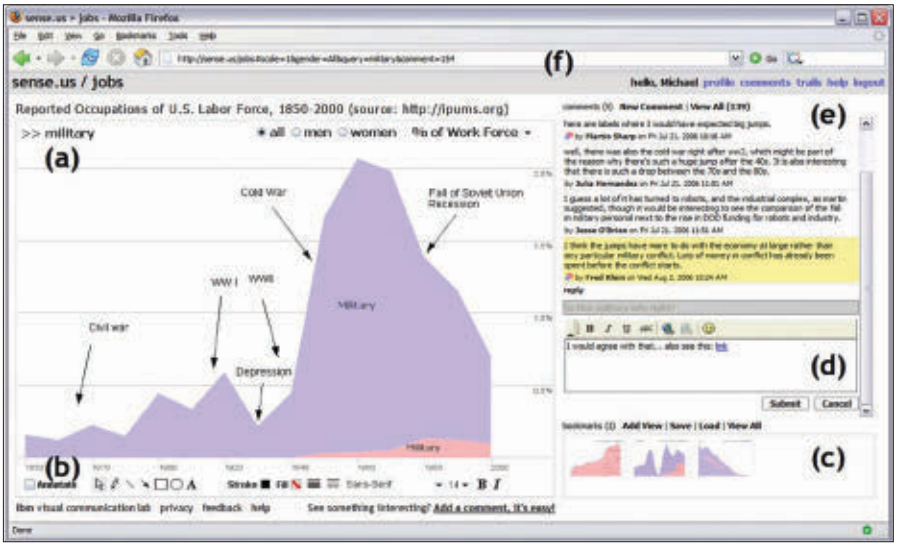


Figure 11. Collaborative visual analysis in Sense.us.[10]

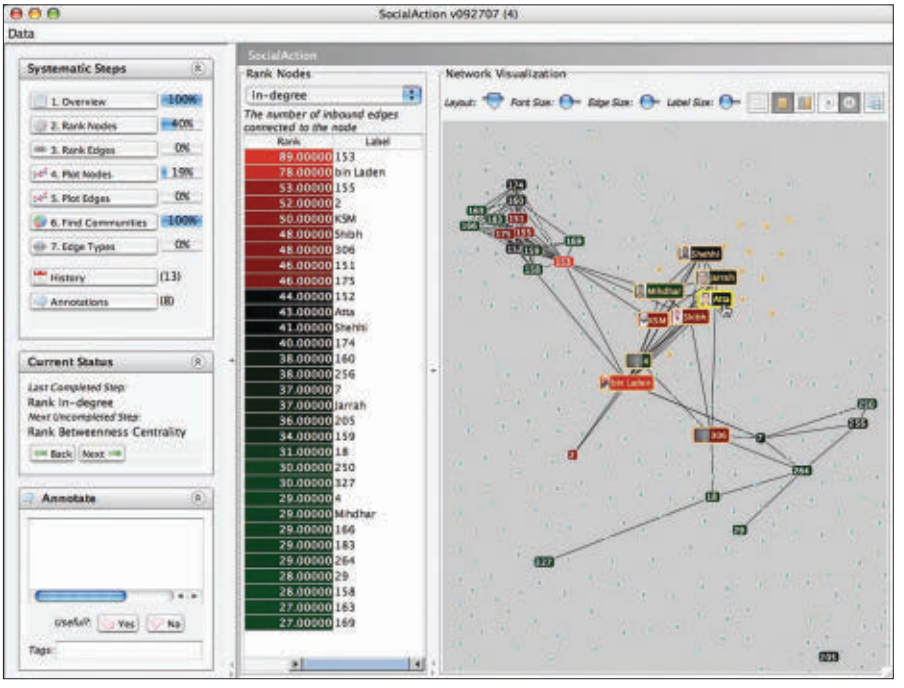


Figure 12. Systematic yet flexible analysis in SocialAction.[15]

steps taken in a visual analysis of business operations data.

Visual histories can support a range of interactions. First, histories provide a convenient mechanism to revisit prior analysis states and resume incomplete explorations. Adding metadata such as comments, tags, or ratings to states can facilitate later review and sharing. Interactive histories can also capture a repeatable sequence of operations that can be named and saved as a reusable macro. This feature enables analysts who are dealing with many similar data sets to automate their efforts. Histories might spur sharing: analysts can export selected analysis trails, ranging from screen shots to interactive presentations, to external media. Finally, histories also provide a means to study analysts and model analytic processes.

Annotate. Interactive visualizations often serve not only as data-exploration tools, but also as a means for recording, organizing, and communicating insights gained during exploration. One option is to allow textual annotation of states within a visual history. More expressive annotations are possible through direct interaction with the view, using the selection techniques discussed earlier. Analysts may wish to "point" to specific items or regions within a visualization and associate these annotations with explanatory text or links to other views.

Freeform *graphical annotations* provide one expressive form of pointing.[10] Drawing a circle around a cluster of items or pointing an arrow at a peak in a graph can direct the attention of viewers. The angle or color of the arrow or shape of the hand-drawn circle may communicate emotional cues or add emphasis. Although such drawings allow a high degree of expression, they lack an explicit tie to the underlying data. Free-form annotations implemented as vector graphics can persist over geometric transformations such as panning and zooming, but if they are not "*data-aware*," then they may become meaningless in the face of operations such as filtering or aggregation.

Annotations can be made data-aware when realized as selections. These selections can be represented as a set of selected items, a declarative query, or both. Data-aware annotations allow a pointing intention to be reapplied to different views of the same data, enabling reuse of refer-

ences across different choices of visual encodings. As data-aware annotations are machine readable, they might also be used to enable search or identify data subsets of high interest.

Share. Researchers in visual analytics often focus on the perceptual and cognitive processes of a single analyst. In practice, real-world analysis is also a social process that may involve multiple interpretations, discussion, and dissemination of results.[10,22] To support the analysis life cycle, visual analytics tools should support social interaction. At minimum, tools must be able to export views or data subsets for sharing and revisitation. Figure 11 shows sense.us,[10] one example of a collaborative visual analysis tool incorporating view sharing, annotation, and discussion. The system consists of (a) an interactive visualization, (b) a set of graphical annotation tools, (c) bookmark trails for saved views, (d) text-entry field for adding comments (bookmarks can be dragged onto the text field to link views to a comment), (e) textual comments attached to the current view, and (f) a shareable URL that is updated automatically as the visualization state changes.

A simple but effective aid to collaboration is view sharing via *application bookmarking*: a visual analytics system should be able to model and export its internal state. Unlike a static screen shot, bookmarking enables analysts to take up an exploration where their collaborators left off. View sharing often takes the form of a URL or similar identifier that allows a collaborator to navigate quickly to a view of interest. Seeing an identical view provides collaborators with a common ground for discussion. Annotation methods can be applied within such views to further collaboration. One challenge for effective view sharing concerns dynamic data: should a bookmarked view maintain a snapshot to historical data, provide access to the most current data, or both?

Another method of sharing and dissemination is to *publish* a visualization. Commercial tools such as Spotfire and Tableau can publish visualization dashboards as interactive Web pages with support for selection, search, and drill-down to enable some amount of follow-up analysis. Services such as IBM's Many Eyes[22] can be used to embed visualization applets in external Web sites. While publishing is a necessary condition for broad sharing, it may not be sufficient by itself for engaging viewers. Visualizations embedded within a blog or discussion forum can reach an established audience and may foster discussion more effectively than a centralized site.

Other collaborative concerns depend on the context of use. Are collaborators working *synchronously* (same time) or *asynchronously* (different time)? Are they *co-located* (same place) or *distributed* (different place)? Each of these configurations may require specialized strategies for access control, presence indicators, and activity awareness.[10,14]

Guide. The exploration process is well understood for some traditional domains. For example, a very simple workflow might remove incomplete data items, sort, select high-value items, and report on these selections. Analysts, however, may need to develop new strategies that are formalized to guide newcomers and provide progress indicators to experts. Visual analysis systems can incorporate *guided analytics* to lead analysts through workflows for common tasks.

Some processes are clearly linear, but many visual analytics tasks require richer *systematic yet flexible* processes that allow analysts to take excursions while keeping track of what they have done. For example, SocialAction[15] organizes social-network analysis into a sequence of activities (for example, rank nodes, plot nodes, find communities); the system allows analysts to skip steps selectively and keeps a record of which steps have been completed. In Figure 12, the panel on the left suggests common steps to structure social network analysis and provides progress indicators. In a related vein, experts often develop visualizations that are used by less knowledgeable team members, in much the same way that spreadsheet macros enable specialists to encode accounting or business practices for others. More research is needed to identify effective visual analytics processes and enable expert analysts to create reusable workflows.

In recent years, journalists have been experimenting with different forms of *narrative visualization*[16] by structuring interactive graphics to tell stories with data. Visualizations from *The New York Times*, *Washington Post*, *The Guardian*, and other news sources often lead the viewer through a linear narrative, guided by supporting text and annotations. In Figure 13, for example, an interactive graphic guides the reader through decades of budget predictions. At a story's conclusion, such visualizations provide interactive controls for further exploration. These narrative structures both communicate key observations from the data and cleverly provide a *tacit tutorial* of the available interactions by animating each component along with the story. By the time the presentation opens up for freeform exploration,

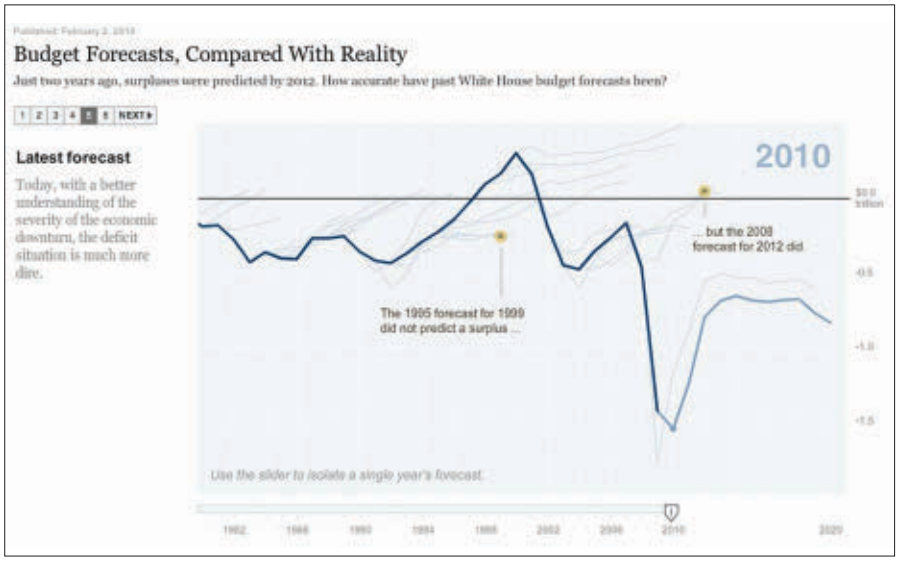


Figure 13. Data storytelling by *The New York Times*.

the viewers have already seen demonstrations of the interactive controls. These forms of narrative visualization demonstrate how guided analytics can help disseminate data-driven stories to a general audience.

Conclusion

We hope this taxonomy and discussion will help advance visual analytics on multiple fronts. For students and newcomers to the field, the taxonomy provides an orientation to the interactive concerns at the heart of visual analysis. We encourage interested readers to consult the systems, books, and papers referenced in this article to develop a deeper understanding. For developers, the taxonomy provides a checklist of items to consider when creating new analysis tools. For researchers, the taxonomy helps highlight critical areas that would benefit from further investigation, including new methods for interactive view specification, a closer integration of visualization and statistical algorithms, and effective approaches to guided analytics.

Of course, by attempting to provide an abstracted picture of a domain, taxonomies may be incomplete. In some cases, we separately categorize aspects that are closely related. Dynamic query widgets enabling data specification often serve as a means of view navigation. Selection techniques are also central to effective annotation schemes.

In other instances, we selectively omit material. For example, we do not go into great depth regarding implementation details. Especially for large datasets, supporting real-time interactivity requires careful attention to system design and poses important research challenges ranging from low-latency architectures to intelligent sampling and aggregation methods. How to best incorporate statistical methods into a visualization environment remains a central challenge; our discussion of derived data only scratches the surface. Other concerns include formatting, cleaning, and integrating data. Incorrect or improperly structured data diverts the energy of trained analysts and presents a significant barrier to newcomers.

These concerns represent active areas of research, and we expect our characterization of the field to evolve in the years to come. We invite the insights and commentary of the visualization, statistics, database, and HCI communities, and eagerly anticipate the continued flowering of improved tools for making sense of the wealth of data that surrounds us.

Acknowledgments

We thank our colleagues and students for providing valuable comments on drafts: Maneesh Agrawala, Jason Chuang, Cody Dunne, John Guerra-Gomez, Pat Hanrahan, Sean Kandel, Diana MacLean, and Kostas Pantazos.

This work was partially supported by National Science Foundation grants IIS-0968521, IIS-1017745, CCF-0964173 and SBE-0915645, NIH-National Cancer Institute grant RC1-CA147489, and ONC-SHARP grant on Cognitive Information Design and Visualization.

Related articles on queue.acm.org

A Conversation with Jeff Heer, Martin Wattenberg and Fernanda Viégas
http://queue.acm.org/detail.cfm?id=1744741

A Tour through the Visualization Zoo
Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky
http://queue.acm.org/detail.cfm?id=1805128

A Conversation with Ed Catmull
http://queue.acm.org/detail.cfm?id=1883592

References

For a complete set of references, see http://queue.acm.org/detail.cfm?id=2146416/

1. Abram, G. and Treinish, L. An extended data-flow architecture for data analysis and visualization. In *Proceedings of the IEEE Conference on Visualization* (1995), 263–270.
2. Becker, R.A. and Cleveland, W. S. Brushing scatterplots. *Technometrics 29*, 2 (1987), 127–142.
3. Bederson, B.B. and Hollan, J.D. Pad++: a zooming graphical interface for exploring alternate interface physics. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (1994), 17–26; http://doi.acm.org/10.1145/192426.192435.
4. Card, S.K., Mackinlay, J. and Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.
5. Cleveland, W.S. *The Elements of Graphing Data*. Hobart Press, Lafayette, IN, 1994.
6. Derthick, M. and Roth, S.F. Enhancing data exploration with a branching history of user operations. *Knowledge Based Systems 14*, 1-2 (2001): 65–74.
7. Grossman, T. and Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2005), 281-290; http://doi.acm.org/10.1145/1054972.1055012.
8. Heer, J. and Card, S.K. DOITrees revisited: scalable, space-constrained visualization of hierarchical data. *Proceedings of Advanced Visual Interfaces:* (2004), 421–424; http://doi.acm.org/10.1145/989863.989941.
9. Heer, J., Mackinlay, J., Stolte, C. and Agrawala, M. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008): 1189–1196; http://portal.acm.org/citation.cfm?id=1477066.1477414.
10. Heer, J., Viégas, F. B. and Wattenberg, M. Voyager and voyeurs: supporting asynchronous collaborative information visualization. *Commun. ACM 52*, 1 (Jan. 2009): 87–97; http://doi.acm.org/10.1145/1435417.1435439.
11. Hill, W.C. and Hollan, J.D. Deixis and the future of visualization excellence. In *Proceedings of the IEEE Conference on Visualization*: (1991), 314–320; http://portal.acm.org/citation.cfm?id=949607.949659.
12. Hochheiser, H. and Shneiderman, B. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Info. Visualization 3*, 1 (2004), 1–18.
13. Inselberg, A. and Dimsdale, B. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the IEEE Conference on Visualization*, (1990), 361–378.
14. Isenberg, P., Tang, A. and Carpendale, S. An exploratory study of visual information analysis. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, (2008), 1217–1226; http://doi.acm.org/10.1145/1357054.1357245.
15. Perer, A. and Shneiderman, B. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. *Proceedings of Intelligent User Interfaces*, (2008), 109–118; http://doi.acm.org/10.1145/1378773.1378788.
16. Segel, E. and Heer, J. Narrative visualization: telling stories with data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1139–1148.
17. Shneiderman, B. Dynamic queries for visual information seeking. *IEEE Software 11*, 6 (1994), 70–77.
18. Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings of the IEEE Symposium on Visual Languages*, 1996; http://portal.acm.org/citation.cfm?id=832277.834354.
19. Stolte, C., Tang, D. and Hanrahan, P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics 8* (2002), 52–65.
20. Tufte, E. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.
21. van Ham, F. and Perer, A. Search, show context, expand on demand: supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 953–960; http://dx.doi.org/10.1109/TVCG.2009.108.
22. Viégas, F.B., Wattenberg, M., van Ham, F., Kriss, J. and McKeon, M. Many Eyes: a site for visualization at Internet scale. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1121–1128.
23. Weaver, C. E. Building highly-coordinated visualizations in Improvise. In *Proceedings of the IEEE Information Visualization Conference*, (2004), 159–166.
24. Wilkinson, L. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, Secaucus, NJ, 2005.

Jeffrey Heer (jheer@cs.stanford.edu) is an assistant professor of computer science at Stanford University, where he works on human-computer interaction, visualization, and social computing. In 2009, he was named to *MIT Technology Review*'s TR35 (35 innovators under the age of 35).

Ben Shneiderman (ben@cs.umd.edu) is a professor in the department of computer science, founding director of the Human-Computer Interaction Laboratory, and a member of the Institute for Advanced Computer Studies at the University of Maryland, College Park.

DOI:10.1145/2133806.2133822

Article development led by acmqueue
queue.acm.org

With this open database, you can mine microprocessor trends over the past 40 years.

BY ANDREW DANOWITZ, KYLE KELLEY, JAMES MAO, JOHN P. STEVENSON, AND MARK HOROWITZ

CPU DB: Recording Microprocessor History

IN NOVEMBER 1971, Intel introduced the world's first single-chip microprocessor, the Intel 4004. It had 2,300 transistors, ran at a clock speed of up to 740KHz, and delivered *60,000* instructions per second while dissipating 0.5 watts. The following four decades witnessed exponential growth in compute power,

a trend that has enabled applications as diverse as climate modeling, protein folding, and computing real-time ballistic trajectories of angry birds. Today's microprocessor chips employ billions of transistors, include multiple processor cores on a single silicon die, run at clock speeds measured in gigahertz, and deliver more than four million times the performance of the original 4004.

Where did these incredible gains come from? This article sheds some light on this question by introducing CPU DB (cpudb.stanford.edu), an open and extensible database collected by Stanford's VLSI Research Group over several generations of processors (and students). We gathered information on commercial processors from 17 manufacturers and placed it in CPU DB, which now contains data on 790 processors spanning the past 40 years.

In addition, we provide a methodology to separate the effect of technology scaling from improvements on other frontiers (for example, architecture and software), allowing the comparison of machines built in different technologies. To demonstrate the utility of this data and analysis, we use it to decompose processor improvements into contributions from the physical scaling of devices, and from improvements in microarchitecture, compiler, and software technologies.

Figure 1. The diamonds indicate how processor performance actually scaled with time, while the squares denote how much speedup came from improving the manufacturing process.

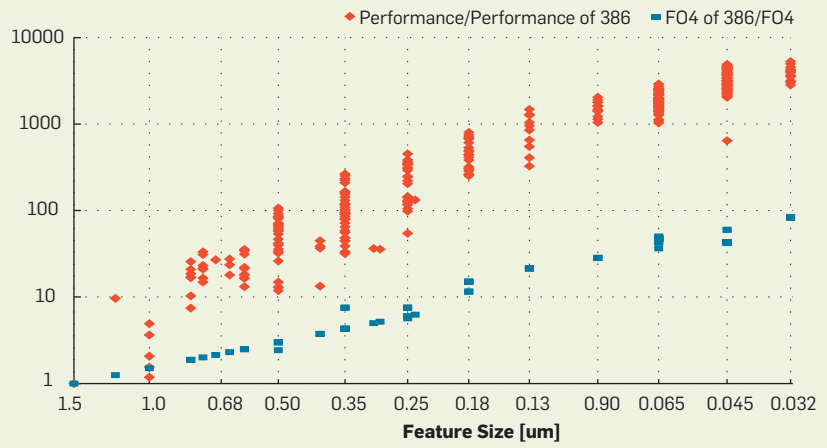


Figure 2. Pollack's rule using CPU DB: performance vs. transistor count. The regression yields $Perf_{norm} = n_{trans}^{0.37}$.

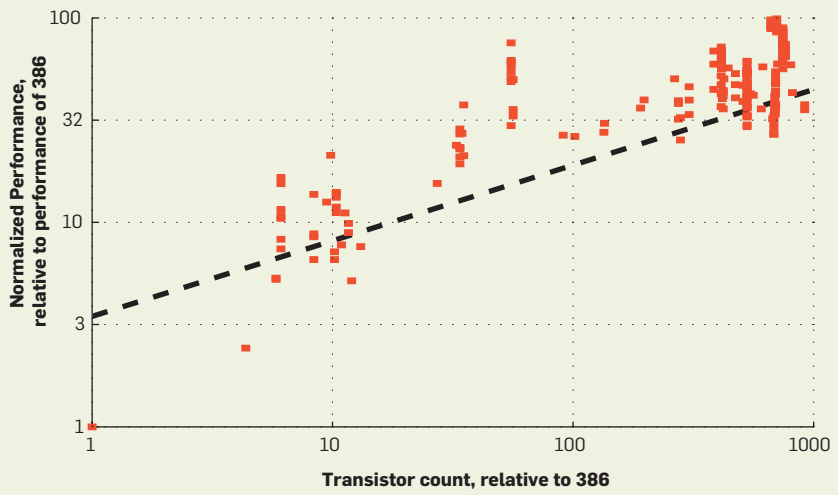


Table 1. Categories used to organize per-processor specifications in CPU DB.

Category			
Processor architecture and microarchitecture	Memory system	Physical characteristics	Technology
Summary Parameter			
Architecture family	Last level cache	Vdd nominal Clock frequency TDP	Process size
Parameters			
Manufacturer	L1 data size	Vdd high	Process name
Family name	L1 instruction size	Vdd low	Process type
Code name	L2 size	Nominal frequency	Feature size
Model name	L3 size	Turbo frequency	Effective channel length
Date released	Memory bandwidth	Low power frequency	Number of metal layers
Number of cores	FSB pins	TDP	Metal type FO4 delay
Threads per core	Memory pins	Die size	
Word size	Power and ground pins I/O pins	Number of transistors	

While information about current processors is easy to find, it is rarely arranged in a manner that is useful to the research community. For example, the data sheet may contain the processor's power, voltage, frequency, and cache size, but not the pipeline depth or the technology minimum feature size. Even then, these specifications often fail to tell the full story: a laptop processor operates over a range of frequencies and voltages, not just the 2GHz shown on the box label.

Not surprisingly, specification data gets more difficult to find the older the processor becomes, especially for those that are no longer made, or worse, whose manufacturers no longer exist. We have been collecting this type of data for three decades and are now releasing it in the form of an open repository of processor specifications. The goal of CPU DB is to aggregate detailed processor specifications into a convenient form and to encourage community participation, both to leverage this information and to keep it accurate and current. CPU DB is populated with desktop, laptop, and server processors, for which we use SPEC[13] as our performance-measuring tool. In addition, the database contains limited data on embedded cores, for which we are using the CoreMark benchmark for performance.[5] With time and help from the community, we hope to extend the coverage of embedded processors in the database.

For users to analyze different processor features, CPU DB contains many data entries for each CPU, ranging from physical parameters such as number of metal layers, to overall performance metrics such as SPEC scores. To make viewing relevant data easier, the database includes summary fields, such as nominal clock frequency, that try to represent more detailed scaling data. Table 1 shows the current list of CPU DB parameters. Table 2 summarizes the "microarchitecture" specifications.

All high-performance processors today tell the system what supply voltage they need within a range of allowable values. This makes it difficult to track how power-supply voltage has scaled over time. Instead of relying on the specified worst-case behavior, researchers are free to analyze the power, frequency, and voltage that a

processor actually uses while running an application, and then add it to the CPU DB repository. Table 3 is a summary of the measured parameters tracked in CPU DB.

While CPU DB includes a large set of processor data fields, certain members of the architecture community will likely want to explore data fields that we did not think to include. To handle such situations, users are encouraged to suggest new data columns. These suggestions will be reviewed and then entered in the database.

A similar system helps keep CPU DB accurate and up to date. Users can submit data for new processors and architectures, and suggest corrections to data entries. We understand that users may not have data for all of the specifications, and we encourage users to submit any subsets of the data fields. New data and corrections will be reviewed before being applied to the database.

With these mechanisms for adding and vetting data, CPU DB will be a powerful tool for architects who wish to incorporate processor data into their studies. Because many database users will probably want to perform analyses on the raw CPU DB data, the full database is downloadable in comma-separated value format.

Technology Normalization Methodology

CPU DB allows side-by-side access to performance data for relatively simple in-order processors (up to the mid-1990s) and modern out-of-order processors. One could ask if, at the cost of lower performance, the simplicity of the older designs conferred an efficiency advantage. Unfortunately, direct comparisons using the raw data are difficult because, over the years, manufacturing technologies have improved significantly. A fair comparison would be possible if both processors were manufactured using the same process; but since porting all of these older processors to modern technologies is not feasible, we need another approach. To enable such comparisons, we instead estimate how processor performance and power would scale with technology.

Our main performance metric is based on industry-standard SPEC CPU2006 scores.[13] Unfortunately, most older processors did not run SPEC 2006 and instead measured performance in MIPS (million instructions per second) and, later, in terms of SPEC 1989, SPEC 1992, SPEC 1995, and SPEC 2000. In those cases we estimate SPEC 2006 numbers by converting old scores into a *SPEC 2006 equivalent* score using a conversion factor. The conversion values are determined by examining systems that have scores for two versions of SPEC and then taking the geometric mean of the set of ratios between overlapping scores. This method was used to create the summary performance scores in the database. We also provide the raw scores so that users can develop better conversion methods over time.

To estimate the performance of a processor if it were manufactured using a newer process, we calculate the

Table 2. Microarchitectural parameters contained in CPU DB.

Manufacturer	Microarchitecture	Revision	ISA
ISA version	ISA extensions	Floating point pipe stages	Integer pipe stages
Max uOps issued per cycle	Integer functional units	Load store functional units	Floating point functional units
Total functional units	Max instructions decoded per cycle	Reorder buffer	Instruction window size
Instruction fetch queue size	Branch history table	Branch target buffer	Branch predictor accuracy
Integer registers	Floating point registers	Total registers	Floating point coproc.
TLB entries	Out of order	Integrated mem. controller	

Table 3. Measured parameters in CPU DB. Note that spec benchmarks also include comprehensive fields for performance on individual SPEC subtests.

Power	Voltage	Performance
Power for specified load	Vdd for specified load	SPECRATE 2006
Idle power	Vdd idle	SPEC 2006
Max operating power	Vdd at max power	SPEC 2000
		SPEC 1995
		SPEC 1992
		MIPs

Figure 3. Pollack's rule using CPU DB: performance vs. normalized area. The regression yields $Perf_{norm} = n_{trans}^{0.46}$.

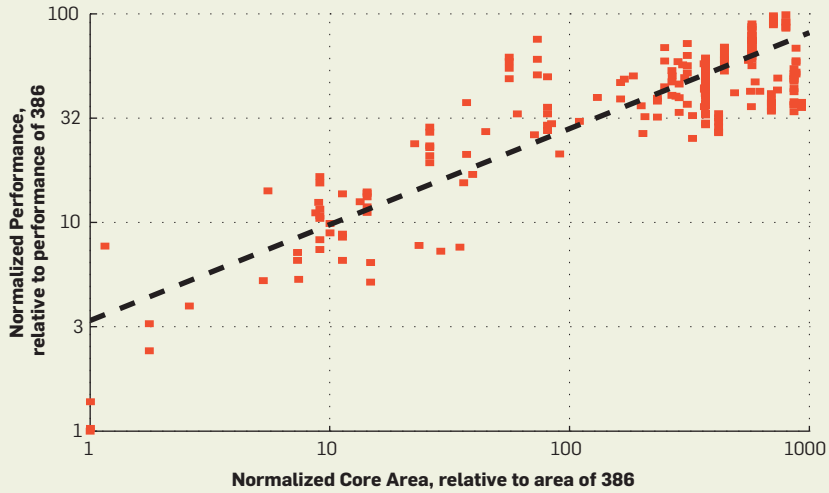


Figure 4. Scaling of transistor feature sizes over time. Up to the 130nm node, feature size scaled every two to three years. Since the 90nm generation, feature size scaling has accelerated to every two years.

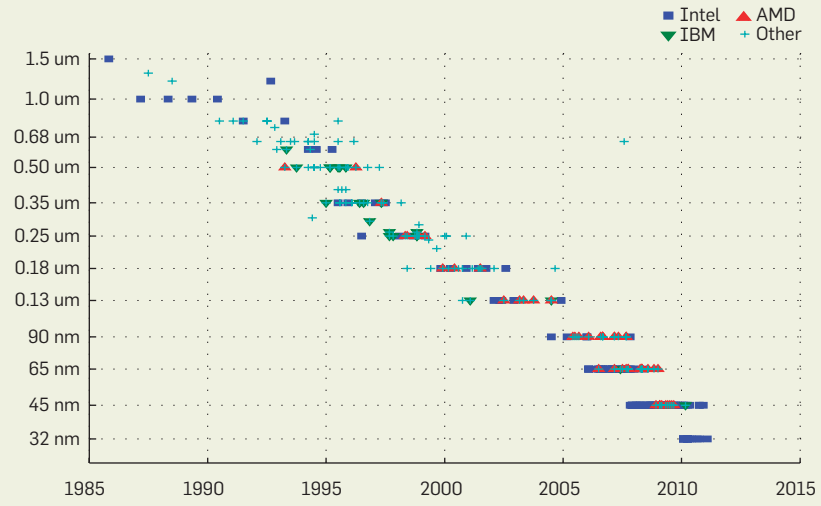
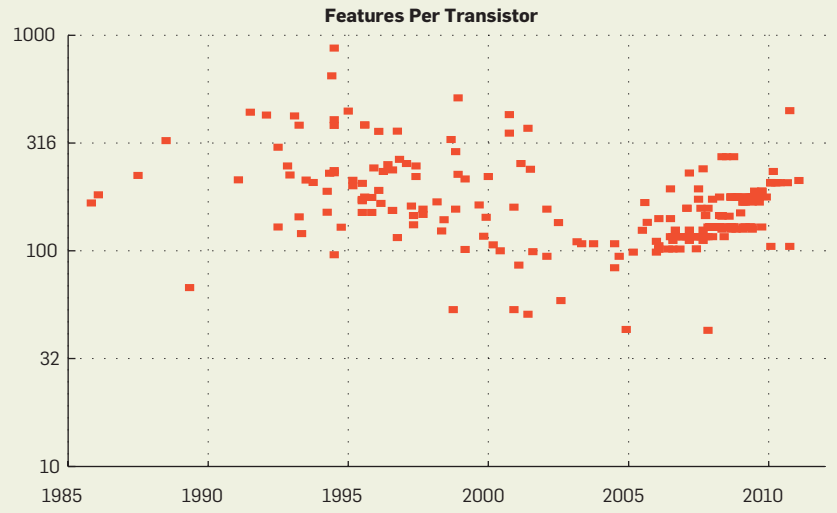


Figure 5. In modern chips, the number of features per transistor has started to grow.



clock frequency in that technology using gate-delay data. While the speed of the cache memory on the processor scales with technology, the delay going to main memory has scaled only slowly with time. As a result, doubling the clock frequency generally does not double the processor's performance. We finesse this issue the same way the microprocessor industry does: by scaling the on-chip cache so the percentage memory stall time remains constant. Using the empirical rule that miss rates are proportional to the square root of the cache size,[9,14] we expand the last-level cache by four times for each doubling of clock frequency. Thus, we assume that the processor performance scales with clock frequency, but we penalize the energy and area of the processor by growing its cache.

For the clock-cycle time estimate, we need to know how the delays of the gates and wires will scale. Fortunately, the delay scaling of different logic gates is similar, so it is sufficient to measure how the delay of a single gate scales. Our analysis uses the delay of an inverter driving four equivalent inverters (a fanout of four, or FO4) as the gate-speed metric. Inverters are the most common gate type, and their delay is often published in technology papers. For wire delay it is important to remember that a design's area will shrink with scaling, so its wire delay will, in general, reduce slowly or, at worst, stay constant. Its effect on cycle time depends on the internal circuit design. Designers generally pipeline long wires, so they tend not to limit the critical path. Thus, we ignore wire delay and make the slightly optimistic assumption that a processor's frequency in the new technology will be greater by the ratio of FO4s from old to new:

$$f_2 = f_1 \frac{FO4_1}{FO4_2}$$

Using FO4 as a basic metric has an additional advantage: it cleanly covers the performance/energy variation that comes from changing the supply voltage. Two processors, even built in the same technology, might be operated at different supply voltages. The energy difference between the two can be calculated directly from the supply voltage, but the voltage's effect on performance is harder to estimate. Using FO4 data for these designs at two different voltages provides all the information that is needed.

Having accounted for the effect of the scaled memory systems, we find that estimating the power of a processor with scaled technology is fairly straightforward. Processor power has two components: dynamic and leakage. In an optimized design, the leakage power is around 30% of the dynamic power, and the leakage power will scale as the dynamic power scales.[16]

Dynamic power is given by the product of the processor's average activity factor, α (the probability that a node will switch each cycle), the processor frequency, and the energy to switch the transistors:

$$Energy = C\{V_{dd}\}^2$$

The processor's average activity factor depends on the logic and not the technology, so it is constant with scaling. Since capacitance per unit length is roughly constant with scaling, C should be proportional to the feature size λ. We have already estimated how

the frequency will scale, so the estimated power and performance scaling for technology is:

$$P_2 = P_1 \frac{\lambda_2 V_{dd2}^2 FO4_1}{\lambda_1 V_{dd1}^2 FO4_2} + P_{cache}$$

$$Perf_2 = Perf_1 \frac{FO4_1}{FO4_2}$$

For analyzing processor efficiency, it is often better to look at energy per operation rather than power. Energy/op factors out the linear relationship that both performance and power have with frequency (FO4). Lowering the frequency changes the power but does not change the energy/op. Since energy/op is proportional to the ratio of power to performance, we derive equation 3 by dividing equations 1 and 2:

$$\frac{energy}{op} \propto \frac{P_1}{Perf_1} \frac{\lambda_2 V_{dd2}^2}{\lambda_1 V_{dd1}^2} + \frac{P_{cache}}{Perf_1} \frac{FO4_2}{FO4_1}$$

With these expressions, it is possible to normalize CPU DB processors' performance and energy into a single process technology. While Intel's Shekhar Borkar et al. gave a rough sketch of how technology scaling and architectural improvement contributed to processor performance over the years,[2] our data and normalization method can be used to generate an actual scatter plot showing the breakdown between the two factors: faster transistors (resulting from technology scaling) and architectural improvement. As seen in Figure 1, process scaling and microarchitectural scaling each contribute nearly the same amount to processor performance gains.

As a quick sanity check for our normalization results, we plot normalized performance versus transistor count and normalized area in figures 2 and 3. These plots look at Pollack's rule, which states that performance scales as the square root of design complexity.[1] Pollack's rule has been used in numerous published studies to compare performance against processor die resource usage.[2,4,10,15] Figures 2 and 3 show that our normalized data is in close agreement with Pollack's rule, suggesting our normalization method accurately represents design performance.

Physical Scaling

One of the nice side benefits of collecting this database is that it allows one to see how chip complexity, voltage, and power have scaled over time, and how well scaling predictions compare with reality. The rate of feature scaling has accelerated in recent years (Figure 4). Up through the 130nm (nanometer) process generation, feature size scaled down by a factor of

$$\alpha = \frac{1}{\sqrt{2}}$$

approximately every two to three years. Since the 90nm generation, however, a new process has been introduced approximately every two years. Intel appears to be driving this intense schedule and has been one of the first to market for each process since the 180nm generation.

As a result of this exponential scaling, in the 25 years since the release of the Intel 80386, transistor area has shrunk by a factor of almost 4,000. If feature size scaling were all that were driving processor density, then transistor counts would have scaled by the same rate. An analysis of commercial microprocessors, however, shows that transistor count has actually grown by a factor of 16,000.

One simple reason why transistor growth has outpaced feature size is that processor dies have grown. While the 80386 microprocessor had a die size of 103 mm², modern Intel Core i7 dies have an area of up to 296 mm². This is not the whole story behind transistor scaling, however. Figure 5 shows technology-independent transistor density by plotting how many square minimum features an average processor transistor occupies. We generated this data by taking the die area, divid-

Figure 6. Voltage vs. feature size. It is clear that voltage scaling did not follow one simple rule. First, by convention, it was maintained at 5 volts. Once voltage reductions were required, a new convention was established at 3.3 volts. Then voltage was reduced in proportion to feature size until the 130nm node. Log-space regression reveals that voltage scaled roughly as the square root of feature size between the 0.6um and 130nm nodes.

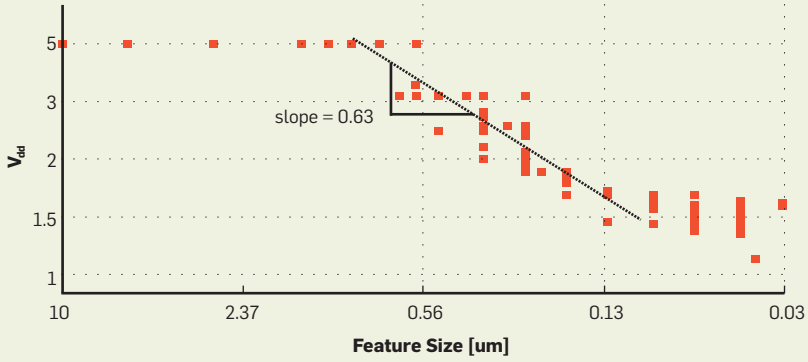
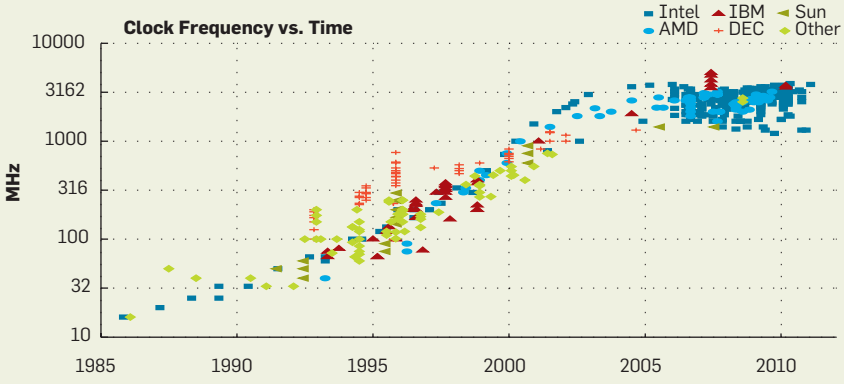


Figure 7. Processor frequency scaling with time. As illustrated, processor frequency has largely leveled off since 2005.



ing by the feature size squared, and then dividing by the number of transistors. From 1985 to 2005 increasing metal layers and larger cache structures (with their high transistor densities) had decreased the average size of a transistor by four times. Interestingly, since 2005, transistor density actually dropped by roughly a factor of two. While our data does not indicate a reason for this change, we suspect it results from a combination of stricter design rules for sub-wavelength lithography, using more robust logic styles in the processor, and a shrinking percentage of the processor area used for cache in chip multiprocessors.

Our data also provides some interesting insight into how supply voltages have scaled over time. Most people know voltage scales with technology feature size, so many assume that this scaling is proportional to feature size as originally proposed in Robert Dennard's 1974 article.[6] As he and others have noted, however, and as shown in Figure 6, voltage has not scaled at the same pace as feature size.[3,12] Until roughly the 0.6 µm node, processors maintained an operating voltage of 5 volts, since that was the common supply voltage for popular logic families of the day, and processor power dissipation was not an issue. It was not until manufacturers went to 3.3 volts in the 0.6 µm generation that voltage began to scale with feature size. Fitting a curve on the voltage data from the half-micron to the 0.13 µm process generations, our data indicates that, even when voltage scaled, it did so with roughly the square root of feature size. This slower scaling has been attributed to reaping a dual benefit of faster gates and better immunity to noise and process variations at the cost of higher chip-power density.

From the 0.13 µm generation on, voltage scaling seems to have slowed. At the same time, however, trends in voltage have become much more difficult to estimate from our data. As mentioned earlier, today almost all processors define their own operating voltage. The data sheets have only the operating range. Figure 6 plots the maximum specified voltage. More user data should provide insight on how supply voltages are really scaling.

Figure 8. Power density over time. From 1985 through 2005, power density grew by roughly a factor of 32. Since 2005, power density has largely started to decrease.

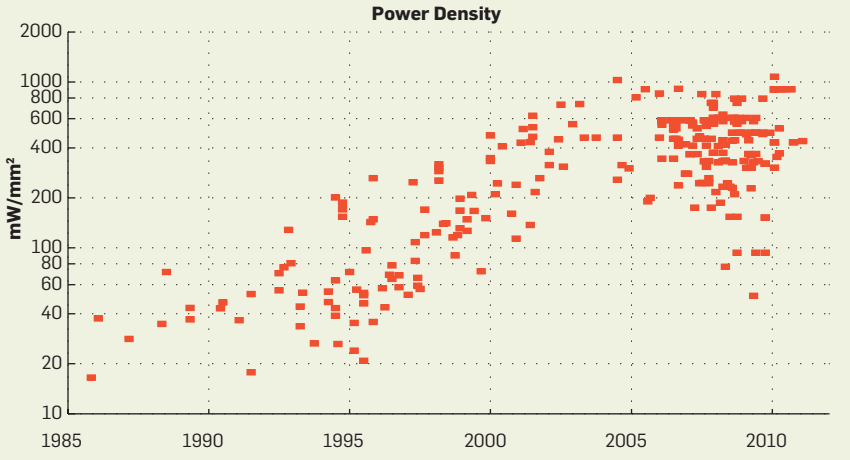
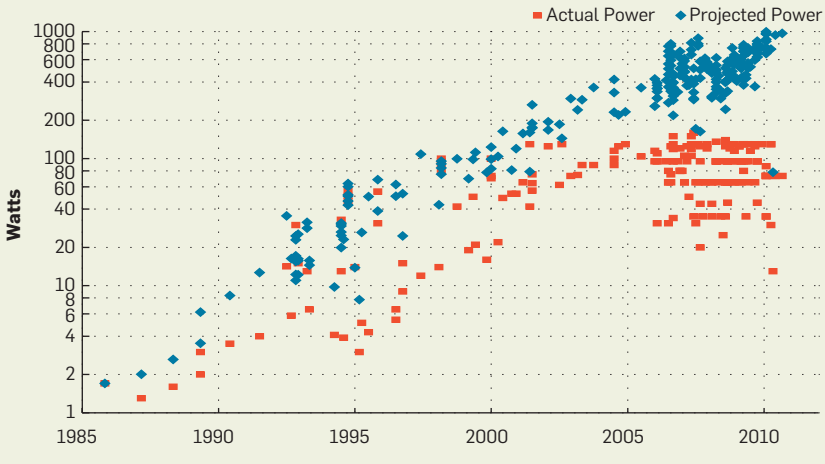


Figure 9. How power should have scaled, given how voltage, number of transistors, and performance actually scaled.



Circuits and Pipelining

Circuit designers and microarchitects were not content to scale frequency with gate speed—if they had been, then microprocessors would be running at only around 500MHz today. As Figure 7 shows, frequencies scaled much faster than simple gate speed. The reason for this discrepancy is largely because of architectural decisions that decreased the logic depth in each processor pipeline stage and increased the number of stages. From 1985 to around 2000, the frequency rapidly increased as a result of faster, more parallel circuit implementations of adders, branch units, and caches, and the use of aggressive pipelining. These trends are evident in the contrast between the two-stage fetch/execute pipeline of the Intel 80386, and the 30-plus pipeline stages in the Prescott Pentium IV.

Since 2000, processor frequencies have stagnated, but this is not the whole story. Our data confirms that gate speeds have continued to improve with technology. What is different now, though, is that the industry has moved away from deeply pipelined machines and is now designing machines that do more work per pipeline stage. The reason for this change is simple: power. While short-tick machines are possible and might be optimal from a performance perspective,[7,11,14] they are not energy efficient.[8]

In light of slower voltage scaling and faster frequency scaling, it comes as no surprise that processor power has increased over time. As illustrated in

Figure 8, processor power density has increased by more than a factor of 32 from the release of the 80386 through 2005, although it has recently started to decrease as energy-efficient computing has grown in importance.

Interestingly, scaling rules say power should be much worse. From the Intel 80386 to a Pentium 4, feature size scaled by 16 times, supply voltage scaled by around four times, and frequency scaled by 200 times. This means that the power density should have increased by a factor of $16 \cdot 200/4^2 = 200$, which is much larger than the power density increase of 32 times shown in Figure 8. Figure 9 compares observed power with how power should have scaled if we just scaled up an Intel 386 architecture to match the performance of new processors. The eightfold savings represents circuit and microarchitectural optimizations—such as clock gating—that have been done during this period to keep power under control. The energy savings of these techniques had initially been growing, but, unfortunately, recently seems to have stabilized at around the eightfold mark. This is not a good sign if we hope to continue to scale performance, since technology scaling of energy is slowing down.

Microarchitecture and Software

While process technologists were finding ways to scale transistors, processor architects were working equally hard in advancing and innovating at the microarchitecture level. Indeed, this effect can be seen in CPU DB where, after normalizing for technology, we observe a hundredfold improvement in microarchitecture/software performance since the Intel 80386 days. Historically, as the number of transistors per chip increased with technology scaling, architects found ways to use those transistors to create faster, more advanced uniprocessors. In addition to aggressive clock scaling, architects implemented features such as speculative execution, parallel instruction issue, out-of-order processing, and larger caches—all of which contributed to improved single-threaded performance.

By approximately 2005, increasingly complex processors, along with slowed voltage scaling, caused processors to hit a new constraint: the power wall. This resulted in a significant shift in the industry. Moore's Law meant that processor designers could still expect an ever-increasing number of transistors, but they had to use these transistors in energy-efficient ways; increasing performance now meant decreasing energy/instruction to keep power constant. As a response to this challenge, the industry transitioned toward CMP (chip multiprocessor) designs that use many simple processors to increase the aggregate performance of the chip.

Figure 10. Energy/op vs. performance. Note these energy/ops do not reflect any scaling of the on-chip memory system.

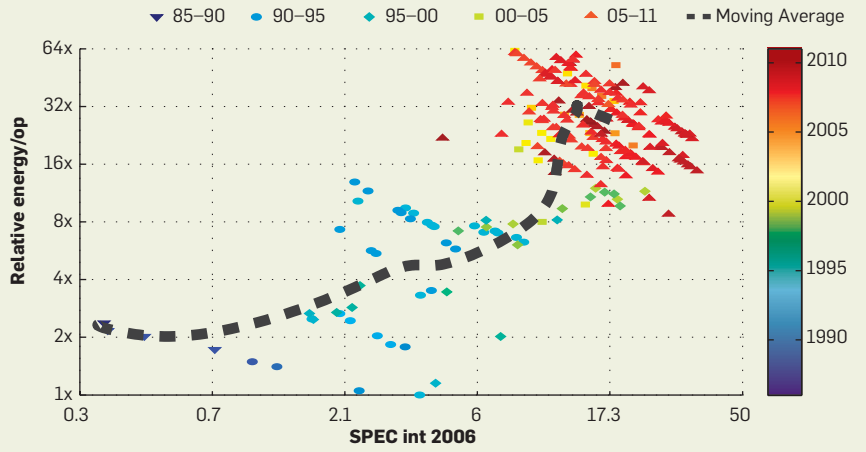


Figure 11. Energy/op vs. performance, modified to scale up the memory system of older cores.

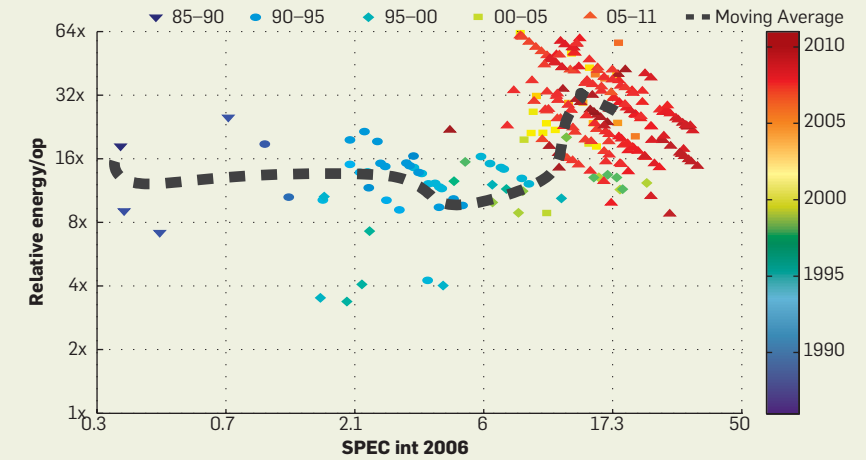


Figure 10 plots the technology-normalized energy/op versus the normalized performance. For this plot, we assume the power needed to scale up the cache size is small compared with the processor power, providing an optimistic assumption of the efficiency of these early machines. This plot indicates that, for early processor designs, energy/op remains relatively constant while performance scales up.

We noticed from this plot, however, that some of the early processors (for example, the Pentium) appear far more energy efficient than modern processor designs. To estimate the scaled energy of these processors more fairly, we scale the caches by the square of the improvement in frequency to keep the memory stall percentage constant, and we estimate the power of a 45nm low-power SRAM at around

Figure 12. Performance vs. year since 2005.

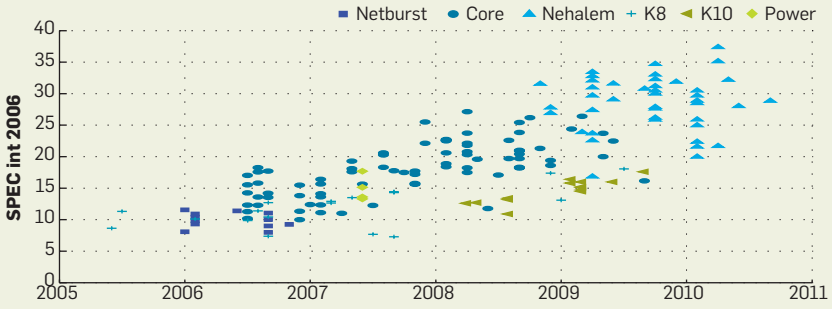


Figure 13. Performance vs. clock rate and cache size since 2005 (LLC is the last level cache size).

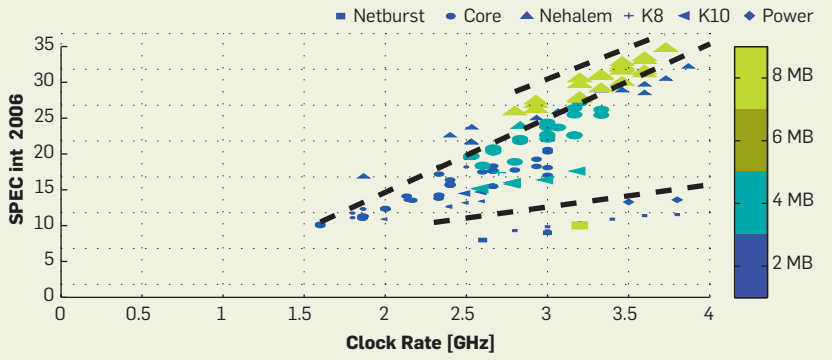


Figure 14. FO4 delays per cycle for processor designs. FO4 delay per cycle is roughly proportional to the amount of computation completed per cycle.

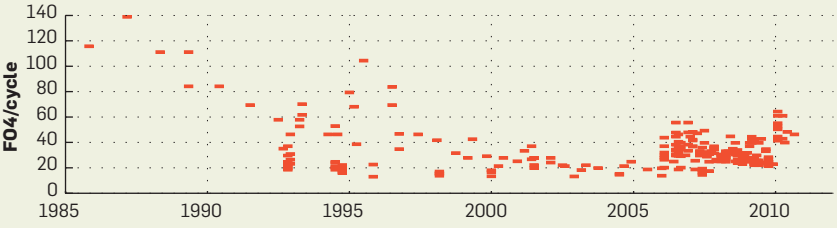
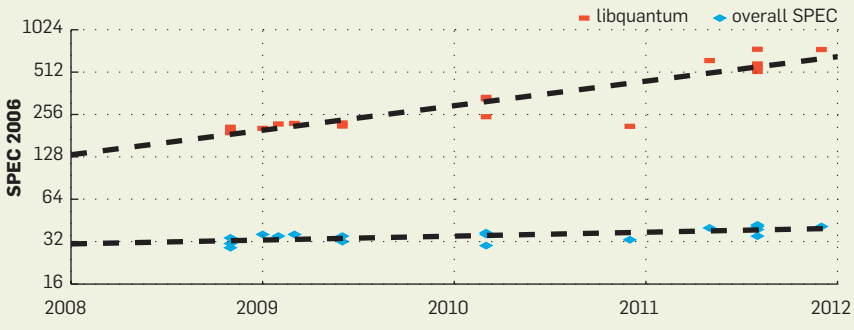


Figure 15. Libquantum score vs. SPEC score. This figure shows how compiler optimizations have led to performance boosts in Libquantum.



0.5W/MB. Including this cache energy-correction factor yields the results in Figure 11. Comparing these two plots demonstrates how critical the memory system is for low-energy processors. The leakage power of our estimated large on-chip cache increases the energy cost of an instruction by four to eight times for simple processors.

Surprisingly, however, the original Pentium designs are still substantially more energy efficient than other designs in the plot. Clearly, more analysis is warranted to understand whether this apparent efficiency can be leveraged in future machines.

In recent years, desktop processors have shifted toward high-throughput parallel machines. With this shift, it was unclear whether processor designers would be able to scale single-core performance. A brief analysis of the data in Figure 12 shows that single-core performance continues to scale with each new architecture. Within an architecture, performance depends largely on the part's frequency and cache size. Figure 13 illustrates this point by plotting the performance versus frequency and cache size for several modern processor designs. Frequency scaling with each new architecture is slower than before, and peak frequencies are now often used only when the other processor cores are idle. Figure 14 plots cycle time measured in gate delays and shows why processor clock frequency seems to have stalled: processors moved to shorter pipelines, and the resulting slower frequency has taken some time to catch up to the older hyperpipelined rates.

More interesting is that even when controlling for the effects of frequency and cache size, single-core microarchitectural performance is still being improved with each generation of chips (Figure 13). Improvements such as on-chip memory controllers and extra execution units all play a role in determining overall system efficiency, and architects are still finding improvements to make.

Our results, however, come with the caveat that some portion of the performance improvement in modern single-core performance comes from compiler optimizations. Figure 15 shows how performance of the SPEC

2006 benchmark Libquantum scales over time on the Intel Bloomfield architecture. Libquantum concentrates a large amount of computation in an inner `for` loop that can be optimized. As a result, Libquantum scores have risen 18 times without any improvement to the underlying hardware. Also, many SPEC scores for modern processors are measured with the Auto Parallel flag turned on, indicating that the measured "single-core" performance might still be benefiting from multicore computing.

Conclusion

Over the past 40 years, VLSI designers have used an incredible amount of engineering expertise to create and improve these amazing devices we call microprocessors. As a result, performance has improved and the energy/op has decreased by many orders of magnitude, making these devices the engines that power our information technology infrastructure. CPU DB is designed to help explore this area. Using the data in CPU DB and some simple scaling rules, we have conducted some preliminary studies to show the kinds of analyses that are possible. We encourage readers to explore and contribute to the processor data in CPU DB, and we look forward to learning more about processors from the insights they develop.

Acknowledgments

Contributing authors to this article are: Omid Azizi, Hicamp Systems, Inc.; John S. Brunhaver II, Stanford University; Ron Ho, Oracle; Stephen Richardson, Stanford University; Ofer Shacham, Stanford University; and Alex Solomatnikov, Hicamp Systems, Inc.

For more information about the online CPU database and how to contribute data, please visit cpudb.stanford.edu.

Even when controlling for the effects of frequency and cache size, single-core microarchitectural performance is still being improved with each generation of chips.

Related articles on queue.acm.org

A Conversation with Steve Furber
http://queue.acm.org/detail.cfm?id=1716385

Real-World Concurrency
Bryan Cantrill, Jeff Bonwick
http://queue.acm.org/detail.cfm?id=1454462

The Price of Performance
Luiz André Barroso
http://queue.acm.org/detail.cfm?id=1095420

References

1. Borkar, S. Thousand core chips: A technology perspective. In *Proceedings of the 44th annual Design Automation Conference* (2007), 746–749; http://doi.acm.org/10.1145/1278480.1278667.
2. Borkar, S. and Chien, A.A. The future of microprocessors. *Commun. ACM 54*, 5 (May 2011), 67–77; http://doi.acm.org/10.1145/1941487.1941507.
3. Chang, L., Frank, D., Montoye, R., Koester, S., Ji, B., Coteus, P., Dennard, R. and Haensch, W. Practical strategies for power-efficient computing technologies. In *Proceedings of the IEEE 98*, 2 (2010); 215–236.
4. Chung, E.S., Milder, P.A., Hoe, J.C. and Mai, K. Single-chip heterogeneous computing: Does the future include custom logic, FPGAs, and GPGPUs? In *Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture* (2010), 225–236; http://dx.doi.org/10.1109/MICRO.2010.36.
5. CoreMark, an EEMBC Benchmark. CoreMark scores for embedded and desktop CPUs; http://www.coremark.org/home.php.
6. Dennard, R., Gaensslen, F., Yu, H., Rideout, V., Bassous, E. and LeBlanc, A. Design of ion-implanted MOSFETs with very small physical dimensions. In *Proceedings of the IEEE 87*, 4 (1999), 668–678 (reprinted from IEEE Journal of Solid-State Circuits, 1974).
7. Hartstein, A. and Puzak, T. The optimum pipeline depth for a microprocessor. In *Proceedings of the 29th Annual International Symposium on Computer Architecture* (2002), 7–13.
8. Hartstein, A. and Puzak, T. Optimum power/performance pipeline depth. In *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture* (Dec. 2003), 117–125.
9. Hartstein, A., Srinivasan, V., Puzak, T.R. and Emma, P.G. Cache miss behavior: Is it $\sqrt{2}$? In *Proceedings of the 3rd Conference on Computing Frontiers* (2006), 313–320; http://doi.acm.org/10.1145/1128022.1128064.
10. Hill, M. and Marty, M. Amdahl's Law in the multicore era. *Computer 41*, 7 (2008), 33–38.
11. Hrishikesh, M., Jouppi, N., Farkas, K., Burger, D., Keckler, S. and Shivakumar, P. The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays. In *Proceedings of the 29th Annual International Symposium on Computer Architecture* (2002), 14–24.
12. Nowak, E.J. Maintaining the benefits of CMOS scaling when scaling bogs down. *IBM Journal of Research and Development 46*, 2.3 (2002), 169–180.
13. Standard Performance Evaluation Corporation (SPEC). SPEC CPU2006 results; http://www.spec.org/cpu2006/results/.
14. Sprangle, E., Carmean, D. Increasing processor performance by implementing deeper pipelines. In *Proceedings of the 29th Annual International Symposium on Computer Architecture* (2002), 25–34.
15. Woo, D.H. and Lee, H.-H. Extending Amdahl's law for energy-efficient computing in the many-core era. *Computer 41*, 12 (2008), 24–31.
16. Zhang, X. High-performance low-leakage design using power compiler and multi-Vt libraries. Synopsys Users Group (SNUG) Europe, 2003.

Andrew Danowitz is currently a Ph.D. candidate in electrical engineering at Stanford University. His research focuses on reconfigurable hardware design methodologies and energy efficient architectures.

Kyle Kelley is a Ph.D candidate in electrical engineering at Stanford University. His research interests include energy-efficient VLSI design and parallel computing.

James Mao is a Ph.D. candidate in the VLSI Research Group at Stanford University working on verification tools for mixed-signal designs.

John P. Stevenson is a graduate of the U.S. Naval Academy and served as an officer onboard the USS *Los Angeles* (SSN-688) prior to entering the Ph.D. program at Stanford University. His interests include computer architecture and digital circuit design. He is investigating novel memory-system architectures.

Mark Horowitz is chair of the electrical engineering department and the Yahoo! Founders Professor at Stanford University, and a founder of Rambus, Inc. His research interests span using EE and CS analysis methods to problems in molecular biology to creating new design methodologies for analog and digital VLSI circuits.

DOI:10.1145/2133806.2133824

Magic numbers are strictly hocus-pocus, so usability studies must test many more subjects than is usually assumed.

BY MARTIN SCHMETTOW

Sample Size in Usability Studies

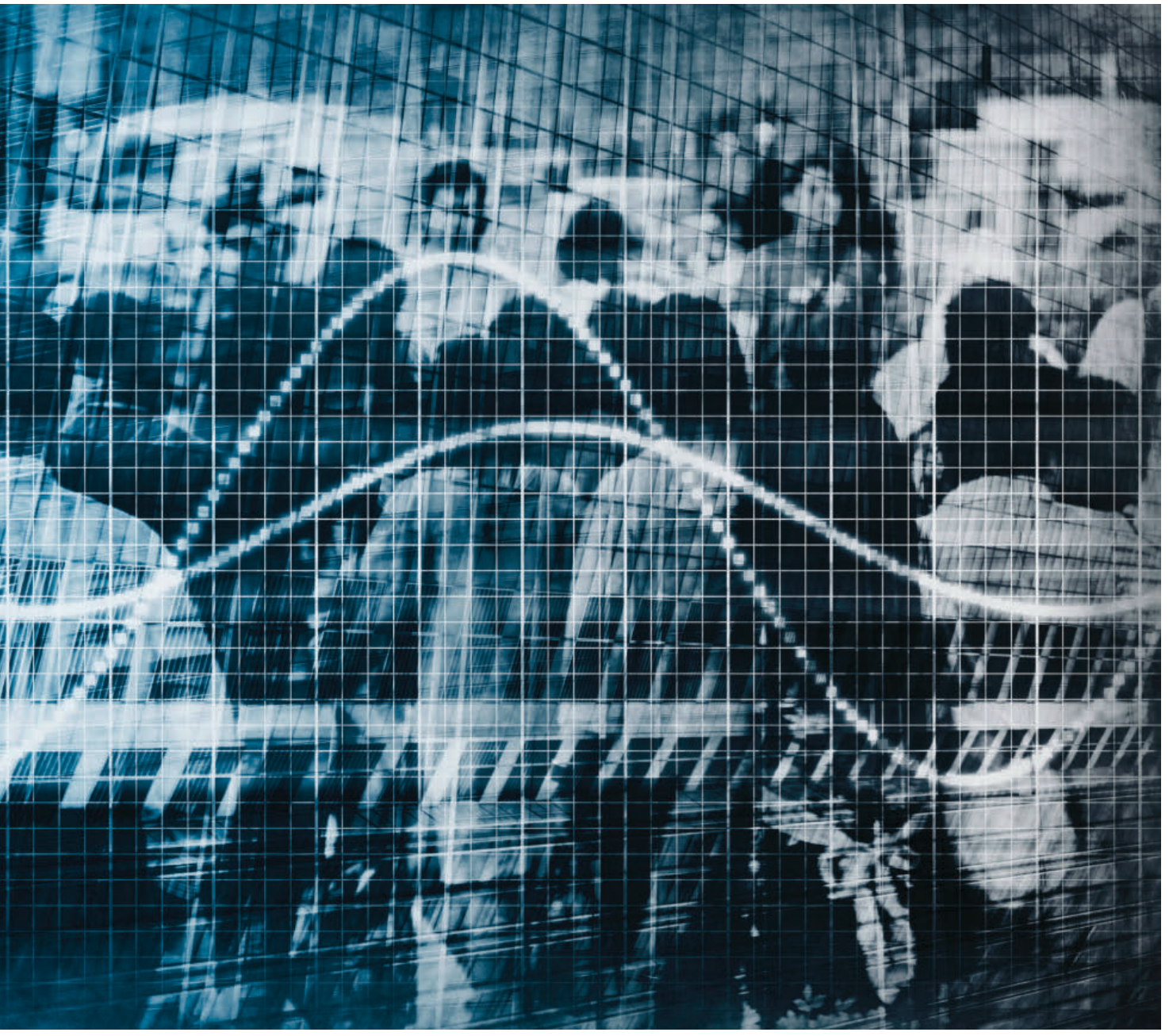
PHOTOGRAPH BY KENTOH / SHUTTERSTOCK.COM

» key insights

- Usability testing is recommended for improving interactive design, but discovery of usability problems depends on the number of users tested.
- For estimating required sample size, usability researchers often resort to either magic numbers or the geometric series formula; inaccurate for making predictions, both underestimate required sample size.
- When usability is critical, an extended statistical model would help estimate the number of undiscovered problems; researchers incrementally add participants to the study until it (almost) discovers all problems.

USABILITY STUDIES ARE a cornerstone activity for developing usable products. Their effectiveness depends on sample size, and determining sample size has been a research issue in usability engineering for the past 30 years.[10] In 2010, Hwang and Salvendy[6] reported a meta study on the effectiveness of usability evaluation, concluding that a sample size of 10 ± 2 is sufficient for discovering 80% of usability problems (not five, as suggested earlier by Nielsen[13] in 2000). Here, I show the Hwang and Salvendy study ignored fundamental mathematical properties of the problem, severely limiting the validity of the 10 ± 2 rule, then look to reframe the issue of effectiveness and sample-size estimation to the practices and requirements commonly encountered in industrial-scale usability studies.

Usability studies are important for developing usable, enjoyable products, identifying design flaws (usability problems) likely to compromise the user experience. Usability problems take many forms,



possibly slowing users doing tasks, increasing the probability of errors, and making it difficult for users to learn a particular product. Usability studies take two general forms: In empirical usability testing, representative users are observed performing typical tasks with the system under consideration, and in usability inspections, experts examine the system, trying to predict where and how a user might experience problems. Many variants of usability testing and expert inspection have been proposed, but how effective are they at actually discovering usability problems? Moreover, how can HCI researchers increase the effectiveness of usability studies? The answer is simple: Increasing the sample size (number of tested participants or number of experts) means more problems will be found. But how many evaluation sessions should researchers conduct? What is a sufficient sample size to discover a certain proportion of problems, if one wants to find, say, at least 80% of all those that are indeed there to be found?

Attempts to estimate the sample size date to 1982[10]; a distinct line of research emerged about 10 years later when Virzi[20] suggested a mathematical model for the progress of usability studies. The proportion of successfully discovered usability problems D was assumed to depend on the average probability p of finding a problem in a single session and number of independent sessions n (the sample or process size). The progress of discovery D was assumed to follow a geometric series $D=1-(1-p)n$.

In 1993, Nielsen and Landauer[14] reported that the average probability p varies widely among studies.

Based on the average *p=0.31* over several studies, Nielsen later concluded that 15 users is typically enough to find virtually all problems,[13] recommending three smaller studies of five participants each (finding 85% of problems in each group) for driving iterative design cycles. Unfortunately, researchers, students, and usability professionals alike misconstrued Nielsen's recommendations and began to believe a simplified version of the rule: Finding 85% of the problems is enough, and five users usually suffice to reach that target.

This conclusion initiated the "five users is (not) enough" debate, involving proponents and skeptics from research and industry.[a] Spool and Schroeder[18] reviewed an industrial dataset, concluding that complex modern applications require a much larger sample size to reach a target of 80% discovery. In 2001, Caulton[3] said the probability of discovering a particular problem likely differs among subgroups within a user population. Likewise, Woolrych and Cockton[22] presumed that heterogeneity in the sample of either participants or experts could render Virzi's formula biased.

The debate has continued to ponder the mathematical foundation of the geometric series model. In fact, the formula is grounded in another well-known model—binomial distribution—addressing the question of how often an individual problem is discovered through a fixed number of trials (sample size or process size *n*). The binomial model is based on three fundamental assumptions that likewise are relevant for the geometric series model:

Independence. Discovery trials are stochastically independent;

Completeness. Observations are complete, such that the total number of problems is known, including those not yet discovered; and

Homogeneity. The parameter *p* does not vary, such that all problems are equally likely to be discovered within a study; I call the opposite of this assumption "visibility variance."

Observing that the average probability *p* varies across studies[14] is a strong argument against generalized assertions like "*X* test participants suffice to find *y*% of problems." A mathematical solution for dealing with uncertainty regarding *p* devised by Lewis[9] suggested that estimating the mean probability of discovery *p* from the first few sessions of a study is helpful in predicting required sample size. Lewis also realized it is not enough to take only the average rate of successful discovery events as an estimator for *p*. The true total number of existing problems is typically unknown a priori, thus violating the completeness assumption. In incomplete studies, not-yet-discovered problems decrease estimated probability. Ignoring incompleteness results in an optimistic bias for the mean probability *p*. For a small sample size, Lewis suggested a correction term for the number of undiscovered problems, or the Good-Turing (GT) adjustment.

However, when evaluating the prediction from small-size subsamples via Monte-Carlo sampling, Lewis treated the original studies as if they were complete. Hence, he did not adjust the baseline of total problem counts for potentially undiscovered problems, which is critical at small process size or low effectiveness. For example, in Lewis's MacErr dataset, a usability testing study with 15 participants, about 50% of problems (76 of 145) were discovered only once. This ratio indicates a large number of problems with low visibility, so it is unlikely that all of them would be discovered with a sample of only 15 users. Hence, the dataset may be incomplete.

Moreover, Lewis's approach was still based on Virzi's original formula, including its homogeneity assumption. In 2008, I showed that homoge-

a For a comprehensive view of the debate see Jeff Sauro's Web site http://www.measuringusability.com/blog/five-history.php

Figure 1. Binomial model fit of the Law and Hvannberg study[8] 169×169mm (72×72DPI).

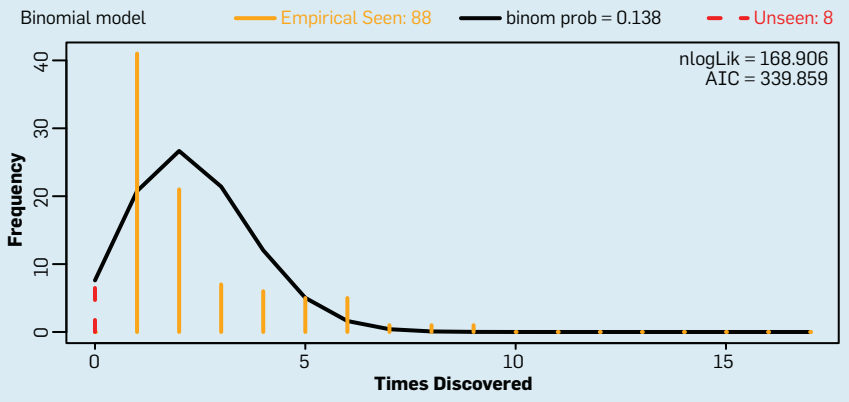
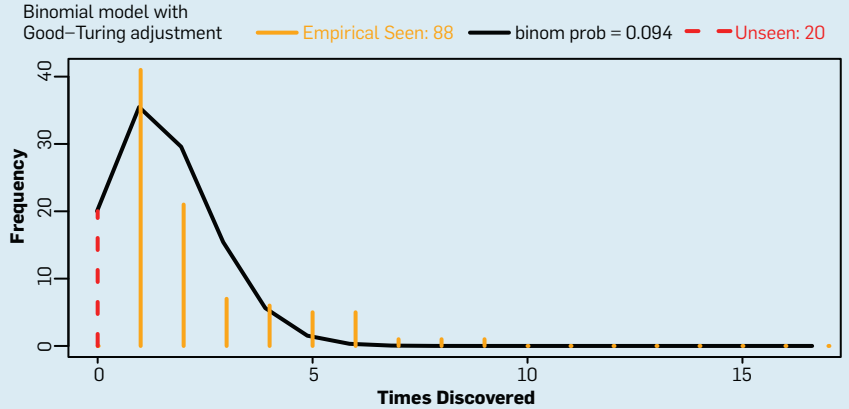


Figure 2. Binomial model fit with Good-Turing adjustment of the Law and Hvannberg study[8] 169×169mm (72×72DPI).



neity cannot be taken for granted.[17] Instead, visibility variance turned out to be the regular case, producing a remarkable effect; progress no longer follows the geometric series, moving instead much more slowly over the long term. The consequence of ignoring visibility variance and not accounting for incompleteness is the same; the progress of a study is overestimated, so the required sample size is underestimated.

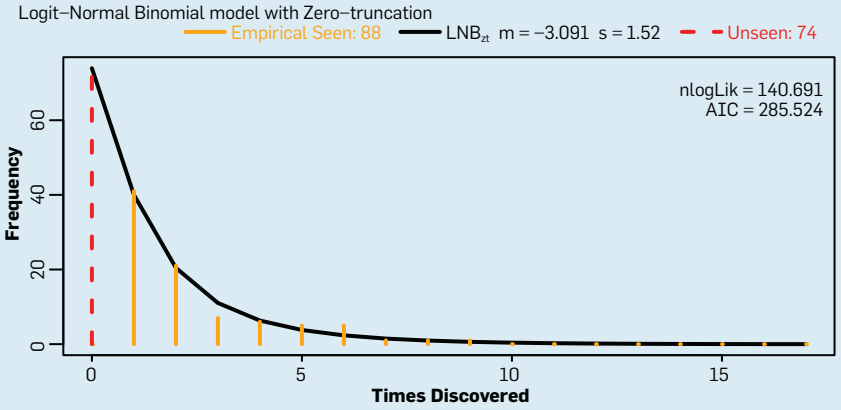
In their 2010 meta study, Hwang and Salvendy[6] analyzed the results of many research papers published since 1990 in order to define a general rule for sample size (replacing Nielsen's magic number five). Hwang's and Salvendy's minimum criterion for inclusion in their study was that a study reported average discovery rates, or number of successful problem discoveries divided by total number of trials (number of problems multiplied by number of sessions). However, this statistic may be inappropriate, as it neither accounts for incompleteness nor for visibility variance. Taking one reference dataset from the meta study as an example, I now aim to show how the 10±2 rule is biased. It turns out that the sample size required for an 80% target is much greater than previously assumed.

Seen and Unseen

In a 2004 study conducted by Law and Hvannberg,[8] 17 independent usability inspection sessions found 88 unique usability problems, reporting on the frequency distribution of the discovery of each problem. A first glance at frequency distribution reveals that nearly half the problems were discovered only once (see Figure 1). This result raises suspicion that the study did not uncover all existing problems, meaning the dataset is most likely incomplete.

In the study, a total of 207 events represented successful discovery of problems. Assuming completeness, the binomial probability is estimated as *p=207/(17*88)=0.138*. Using Virzi's formula, Hwang and Salvendy estimated the 80% target being met through 11 sessions, supporting their 10±2 rule. However, Figure 1 shows the theoretical binomial distribution is far from matching the observed distribution, reflecting three discrepancies:

Figure 3. Fit of the LNB_{zt} model on the Law and Hvannberg study[8] 169×169mm (72×72DPI).



Never-observed problems. The theoretical distribution predicts a considerable number of never-observed problems;

Singletons. More problems are observed in exactly one session than is predicted by the theoretical distribution; and

Frequent occurrences. The number of frequently observed problems (in more than five sessions) is undercounted by the theoretical distribution.

The first discrepancy indicates the study was incomplete, as the binomial model would predict eight unseen problems. The GT estimator Lewis proposed is an adjustment researchers can make for such incomplete datasets, smoothing the data by setting the number of unseen events to the number of singletons, here 41.[b] With the GT adjustment the binomial model obtains an estimate of *p=0.094* (see Figure 2). The GT adjustment lets the binomial model predict the sample size for an 80% discovery target at 16, which is considerably beyond the 10±2 rule.

Variance Matters

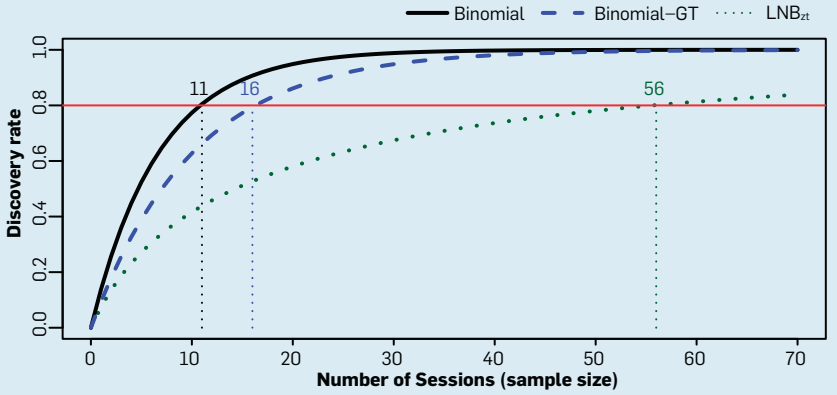
The way many researchers understand variance is likely shaped by the common analysis of variance (ANOVA) and underlying Gaussian distribution. Strong variance in a dataset is interpreted as noise, possibly forcing researchers to increase the sample size; variance is therefore often called a nuisance parameter. Conveniently, the Gaussian distribution has a separate parameter for variance, uncoupling it from the parameter of interest, the mean. That is, more variance makes the estimation less accurate but usually does not introduce bias. Here, I address why variance is not harmless for statistical models rooted in the binomial realm, as when trying to predict the sample size of a usability study.

Binomial distribution has a remarkable property: Its variance is tied to the binomial parameters, the sample size n and the probability p, as in $Var = np(1-p)$. If the observed variance exceeds $np(1-p)$ it is called overdispersion, and the data can no longer be taken as binomially distributed. Overdispersion has an interesting interpretation: The probability parameter p varies, meaning, in this case, problems vary in terms of visibility. Indeed, Figures 1 and 2 shows the observed distribution of problem discovery has much fatter left and right tails than the plain binomial and GT-adjusted models; more variance is apparently observed than can be handled by the binomial model.

Regarding sample-size estimation in usability studies, the 2006 edition of the *International Encyclopedia of Ergonomics and Human Factors* says, "There is no compelling evidence that a probability density function would lead to an advantage over a single value for p."[19] However, my own 2008–2009 results call this assertion into question. The regular case seems to be that p varies, strongly affecting the

b Lewis favors an equally weighted combination of normalization procedure and GT adjustment, but its theoretical justification is tenuous, ultimately making only a small difference to prediction (p=0.085).

Figure 4. Comparing process predictors on the Law and Hvannberg study[8] 169×169mm (72×72DPI).



progress of usability studies.[16,17] When problem visibility varies, progress toward finding new problems would be somewhat quicker in early sessions but decelerate compared to the geometric model as sample size increases. The reason is that easy-to-discover problems show up early in the study. When discovered, they are then frequently rediscovered, taking the form of the fat right tail of the frequency distribution. These reoccurrences increase the estimated average probability p but do not contribute to the study, as progress is measured only in terms of finding new problems. Moreover, with increased variance comes more intractable problems (the fat left tail), and revealing them requires much more effort than the geometric series model might predict.[c]

Improved Prediction

Looking to account for variance of problem visibility, as well as unseen events, I proposed, in 2009, a mathematical model I call the "zero-truncated logit-normal binomial distribution," or LNB_{zt}.[16] It views problem visibility as a normally distributed latent property with unknown mean and variance, so the binomial parameter p can vary by a probability density function—exactly what the encyclopedia article by Turner et al.[19] neglected. Moreover, zero-truncation accounts for the unknown number of never-discovered problems.

Figure 3 outlines the LNB_{zt} model fitted to the Law and Hvannberg dataset. Compared to the binomial model, this distribution is more dispersed, smoothly resembling the shape of the observed data across the entire range. It also estimates the number of not-yet-discovered problems at 74, compared to eight with the binomial model and 20 with GT adjustment, suggesting the study is only half complete.

The improved model fit can also be shown with more rigor than through visual inspection alone. Researchers can use a simple Monte-Carlo procedure to test for overdispersion.[d,17] A more sophisticated analysis is based on the method of maximum likelihood (ML) estimation. Several ways are available for comparing models fitted by the ML method; one is the Akaike Information Criterion (AIC).[2] The lower value for the LNB^{zt} model (AIC=286, see Figure 3) compared to the binomial model (AIC=340, see Figure 1) confirms that LNB^{zt} is a better fit with the observed data.[e]

The LNB_{zt} model also helps usability researchers predict the progress of the evaluation process through the derived logit-normal geometric formula.[16] For the Law and Hvannberg study[8] a sample size of n=56 participants is predicted for the 80% discovery target (see Figure 4), taking HCI researchers way beyond the 10±2 rule or any other magic number suggested in the literature.

Not So Magical

Using the LNBzt model since 2008 to examine many usability studies, I can affirm that visibility variance is a fact and that strong incompleteness usually occurs for datasets smaller than n=30 participants. Indeed, most studies I am aware of are much smaller, with only a few after 2001 adjusting for unseen events and not one accounting for visibility variance. The meta study by Hwang and Salvendy[6] carries both biasing factors—incompleteness and visibility variance—thus most likely greatly understating required sample size.

Having seen data from usability studies take a variety of shapes, I hesitate to say the LNB_{zt} model is the last word in sample-size estimation. My concern is that the LNB_{zt} model still makes assumptions, and it is unclear how they are satisfied for typical datasets "in the wild." Proposing a single number as the one-and-only solution is even less justified, whether five, 10, or 56.

Problem Population

Besides accounting for variance, the LNB_{zt} approach has one remarkable advantage over Lewis's predictor for required sample size: It allows for estimating the number of not-yet-discovered problems. The difference between the two approaches—LNB_{zt} vs. Lewis's adjustment—is that whereas Lewis's GT estimation first smooths the data by adding virtual data points for undiscovered problems, then estimates p, the LNB_{zt} method first estimates the parameters on the unmodified data, then determines the most likely number of unobserved problems.[16]

Recasting the goal from predicting sample size to estimating the number of remaining problems is not a wholly new idea. In software inspection, the so-called capture-recapture (CR) mod-

c Rephrasing this in terms of reliability engineering, the geometric series model becomes the discrete version of the exponential probability function, resulting in a stable hazard function for a problem's likelihood of being discovered. With visibility variance, the hazard function decreases over an increasing number of sessions.

d For a program and tutorial on the Monte-Carlo test for overdispersion see http://schmettow.info/Heterogeneity/

e The GT adjustment adds virtual data points so cannot be compared through AIC.

els have been investigated for managing defect-discovery processes; see, for example, Walia and Carver.[21] CR models are derived from biology, serving to estimate the size of animal populations, as in, for example, Dorazio and Royle.[4] Field researchers capture and mark animals on several occasions, recording each animal's capture history and using it to estimate the total number of animals in the area. Several CR models notably allow for the heterogeneous catchability of animals, usually referred to as Mh models. In inspection research, Mh models allow for visibility variance of defects, frequently helping predict the number of remaining defects better than models with a homogeneity assumption; see, for example, Briand et al.[1]

Also worth noting is that most studies in inspection research focus on a single main question: Have all or most defects been discovered or are additional inspections required? Sample-size prediction is rarely considered. In addition, the number of inspectors is often below the magic numbers of usability-evaluation research. One may speculate that software defects are easier to discover and possibly vary less in terms of visibility compared to usability problems. A detailed comparison of sample size issues in usability studies and management of software inspections has not yet been attempted.

The Timing of Control

The LNB_{zt} model promises to bridge these parallel lines of research, as it supports both goals: predicting sample size and controlling the process. Generally, three strategies are available for managing sample size:

Magic number control. Claims existence of a universally valid number for required sample size;

Early control. Denotes estimating sample size from the first few sessions, as introduced by Lewis[9]; and

Late control. Abstains from presetting the sample size, deciding instead on study continuation or termination by estimating the number of remaining problems; a decision to terminate is made when the estimate reaches a preset target, when, say, less than 20% of problems are still undiscovered.

An approach based on a magic number is inappropriate for prediction because usability studies differ so much in terms of effectiveness. Early control might seem compelling, because it helps make a prediction at an early stage of a particular study when exact planning of project resources is still beneficial; for example, a usability professional may run a small pilot study before negotiating the required resources with the customer. Unlike the late-control strategy, early control is conducted on rather small sample sizes. Hence, the crucial question for planning usability studies is: Do early sample-size predictors have sufficient predictive power?

Confidence of Prediction

The predictive power of any statistical estimator depends on its reliability, typically expressed as an interval of confidence. For the LNB_{zt} model the confidence intervals tend to be large, even at moderate sample size, and are too large to be useful for the early planning of resources; for example, the 90% confidence interval in the full Law and Hvannberg[8] dataset ranges from 37 to 165, for an 80% target. This low reliability renders the early-control strategy problematic, as it promises to deliver an estimate after as few as two to four sessions.[9]

Worth noting is that confidence intervals for the binomial model are typically much tighter.[16] However, tight confidence intervals are never an advantage if the estimator p is biased. There can be no confidence without validity. Fortunately, confidence intervals get tighter when the process approaches completeness and can serve as, say, a late-control strategy.

More Research Needed

The late-control strategy continuously monitors whether a study has met a certain target. Continuous monitoring may eventually enable usability practitioners to offer highly reliable usability studies to their paying customers. However, to serve projects with such strict requirements means any estimation procedure needs further evidence to produce accurate estimates under realistic conditions. The gold standard for assessing the accuracy of estimators is Monte-Carlo sampling, as it makes no assumptions about the shape of the probability distribution. Unfortunately, Monte-Carlo sampling requires complete datasets, implying huge sample sizes. Moreover, such studies must also cover a range of conditions. It cannot be expected that a study involving a complex commercial Web site has the same properties as a study testing, say, a medical infusion pump.

Several studies involving software inspection have validated CR models by purposely seeding defects in the artifacts being considered. This is another way to establish completeness, as the total number of seeded defects is known in advance. However, I doubt it is viable for usability studies. Usability problems are likely too complex and manifold, and designing user interfaces with seeded usability problems requires a substantial development effort and financial budget.

A conclusive approach, despite being lightweight, is to compare goodness-of-fit among various models, as I have tried to show here. A model that better fits the data is probably also superior at predicting a study's future progress. As another advantage, researchers may approach the task of picking a solid predictive model by reexamining existing datasets. However, such an examination requires access to the frequency distribution of problem discovery. Few studies report on that distribution, so, another meta study would require the cooperation of the original authors.

Industrial Applications?

To my knowledge, adoption of quantitative management is marginal in industrial usability studies. Objections seem to reflect two general themes: supporting different goals in the development process and interpreting raw observational data from the studies.

Reacting to Hwang and Salvendy,[6] Molich[11] said that rigid quality assurance is rarely the sole purpose of a usability study; such studies are often done as a kind of screening test to justify another redesign cycle. Accordingly, Nørgaard and Hornbæk found that industrial usability studies are often used to confirm problems that are already known.[15]

Molich[11] also advocated for a series of smaller studies driving an iterative design cycle, reflecting a

broad consensus among usability engineers. However, this approach barely benefits from quantitative control, as such small-scale studies do not strive for completeness. This view is also indirectly supported by John and Marks[7] showing that fixing usability problems is often ineffective and might even introduce new problems. Iterative design mitigates this issue by putting each redesign back into the loop. In the literature, the same study is often cited when the so-called downstream utility of usability evaluation is addressed. Downstream utility carries the effectiveness of usability studies beyond basic discovery of problems by focusing on effective communication and proper redesign guidance. However, such issues are admittedly of higher priority compared to the quantitative control of usability studies.

While the importance of sample size management depends on the context of the study, data quality is a precondition for a prediction to be of value. The models for estimating evaluation processes are based primarily on observing the reoccurrence of problems. Hence, for any observation to be counted it must first be clear to the researchers whether it is novel or a reoccurrence of a previously discovered problem. Several studies have shown only weak consensus on what constitutes a usability problem. Molich's comparative usability evaluation (CUE) series of studies (1998–2011) repeatedly found that any two professional teams running a usability study typically report results that differ in many respects; see, for example, Molich and Dumas.[12] Furthermore, the pattern of reoccurrence depends on the exact procedure to map raw observations onto defined usability problems.[5] All this means that estimations of sample size or remaining problems may lack objectivity because they depend on the often idiosyncratic procedures of data preparation.

Conclusion

Predicting the progress of a usability study is less straightforward than has been assumed in the HCI literature. Incompleteness and visibility variance mean the geometric series formula grossly understates required sample size. Most reports in the literature on usability evaluation effectiveness reflect this optimistic bias, as does the 10±2 rule of Hwang and Salvendy.[6] Consequently, I doubt that 80% of problems can be discovered with only 10 users or even with 10 experts. This limitation should also concern usability practitioners who test only a few participants in iterative design cycles. Most problems are likely to remain undiscovered through such studies.

As much as usability professionals and HCI researchers want a magic number, the very idea of identifying it is doomed to failure, as usability studies differ so much at identifying usability problems. Estimating a particular study's effectiveness from only a few early sessions is possible in theory, but such predictions are too unreliable to be practical. The late-control approach reflects potential for application domains where safety, economic, or political expectations make usability critical. Expensive, quantitatively managed studies can help develop high-quality interactive systems, reflecting that quality assurance was adequate. Most usability practitioners will likely continue to use strategies of iterative low-budget evaluation where quantitative statements are unreliable but also unnecessary. C

References

1. Briand, L.C., El Emam, K., Freimut, B.G., and Laitenberger, O. A comprehensive evaluation of capture-recapture models for estimating software defect content. *IEEE Transactions on Software Engineering 26*, 6 (June 2000), 518–540.
2. Burnham, K.P. and Anderson, D.R. Multimodel Inference. Understanding AIC and BIC in model selection. *Sociological Methods & Research 33*, 2 (Nov. 2004), 261–304.
3. Caulton, D.A. Relaxing the homogeneity assumption in usability testing. *Behaviour & Information Technology 20*, 1 (2001), 1–7.
4. Dorazio, R.M. and Royle, J.A. Mixture models for estimating the size of a closed population when capture rates vary among individuals. *Biometrics 59*, 2 (June 2003), 351–64.
5. Hornbæk, K. and Frøkjær, E. Comparison of techniques for matching of usability problem descriptions. *Interacting with Computers 20*, 6 (Dec. 2008), 505–514.
6. Hwang, W. and Salvendy, G. Number of people required for usability evaluation: The 10±2 rule. *Commun. ACM 53*, 5 (May 2010), 130–133.
7. John, B. and Marks, S. Tracking the effectiveness of usability evaluation methods. *Behaviour & Information Technology 16*, 4 (1997), 188–202.
8. Law, E.L.-C. and Hvannberg, E.T. Analysis of combinatorial user effect in international usability tests. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria, Apr. 24–29). ACM Press, New York, 2004, 9–16.
9. Lewis, J.R. Evaluation of procedures for adjusting problem-discovery rates estimated from small samples. *International Journal of Human-Computer Interaction 13*, 4 (2001), 445–479.
10. Lewis, J.R. Testing small system customer set-up. In *Proceedings of the Annual Meeting of the Human Factors and Ergonomics Society*. Human Factors Society, Santa Monica, CA, 1982, 718–720.
11. Molich, R. How many participants needed to test usability? *Commun. ACM 53*, 8 (Aug. 2010), 7.
12. Molich, R. and Dumas, J. Comparative usability evaluation (CUE-4). *Behaviour & Information Technology 27*, 3 (2008), 263–281.
13. Nielsen, J. *Why You Only Need to Test with 5 Users.* Jakob Nielsen's Alertbox (Mar. 19, 2000); http://www.useit.com/alertbox/20000319.html
14. Nielsen, J. and Landauer, T.K. A mathematical model of the finding of usability problems. In *Proceedings of INTERCHI 1993* (Amsterdam, the Netherlands, Apr. 24–29). ACM Press, New York, 1993, 206–213.
15. Nørgaard, M. and Hornbæk, K. What do usability evaluators do in practice? An explorative study of think-aloud testing. In *Proceedings of the Sixth Conference on Designing Interactive Systems* (University Park, PA, June 26–28). ACM Press, New York, 2006, 209–218.
16. Schmettow, M. Controlling the usability evaluation process under varying defect visibility. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (Cambridge, U.K., Sept. 1–5). British Computer Society, Swinton, U.K., 2009, 188–197.
17. Schmettow, M. Heterogeneity in the usability evaluation process. In *Proceedings of the 22nd British HCI Group Annual Conference on Human-Computer Interaction* (Liverpool, U.K., Sept. 1–5). British Computer Society, Swinton, U.K., 2008, 89–98.
18. Spool, J. and Schroeder, W. Testing Web sites: Five users is nowhere near enough. *CHI Extended Abstracts on Human Factors in Computing Systems* (Seattle, Mar. 31–Apr. 5), ACM Press, New York, 2001, 285–286.
19. Turner, C.W., Lewis, J.R., and Nielsen, J. Determining usability test sample size. In *International Encyclopedia of Ergonomics and Human Factors*, W. Karwowski, Ed. CRC Press, Boca Raton, FL, 2006, 3084–3088.
20. Virzi, R.A. Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors: The Journal of the Human Factors and Ergonomics Society 34*, 4 (1992), 457–468.
21. Walia, G.S. and Carver, J.C. Evaluation of capture-recapture models for estimating the abundance of naturally occurring defects. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (Kaiserslautern, Germany, Oct. 9–10). ACM Press, New York, 2008, 158–167.
22. Woolrych, A. and Cockton, G. Why and when five test users aren't enough. In *Proceedings of the IHM-HCI Conference*, J. Vanderdonckt, A. Blandford, and A. Derycke, Eds. (Lille, France, Sept. 10–14). Cépadèus Éditions, Toulouse, France, 2001, 105–108.

Martin Schmettow (m.schmettow@utwente.nl) is an assistant professor in the Department Cognitive Psychology and Ergonomics of the University of Twente, Enschede, The Netherlands.

DOI:10.1145/2133806.2133823

Even after almost a dozen years, they still deliver solid guidance for software development teams and their projects.

BY LAURIE WILLIAMS

What Agile Teams Think of Agile Principles

IN THE MID-1990s, the prescribed means of keeping software development projects out of trouble and on schedule was to follow a heavyweight software development methodology consisting of a complete requirements document, including architecture and design, followed by coding and testing based on a thorough test plan. The philosophy was often summarized as "Do it right the first time." Common belief among software engineers at the time was that projects run into trouble when they do not strictly adhere to a methodology, and, if only they did, all would be well. In reality, all was rarely well.

At the same time, a simmering undercurrent that had begun to undercut this doctrine was to follow an exceedingly iterative, lightweight software development methodology. Purportedly, a number of independent "rogue" consultants were rescuing projects in trouble through variations of these methodologies. The first to stand up and say, "Look at me," and attract wide attention, was Extreme Programming[1] in about 1999. The creators of other methodologies, including Adaptive Software Development, or ASD,[6] Crysta,[4] Dynamic Systems Development Method, or DSDM,[11] Feature Driven Development, or FDD,[8] and Scrum,[10] followed suit with "Hey, I'm doing something like that, too!"

Then, in February 2001, something remarkable happened: Rather than focus on their differences and the "competitive advantage" of their own methodologies, 17 creators and supporters[a] of the lightweight methodologies gathered in Snowbird, UT, to discuss their

a Software engineers in attendance in Snowbird included Kent Beck, Mike Beedle, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andy Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas, and Arie van Bennekum.

common interests and philosophies, coining the term "agile software development" to describe their methodologies. This unity rocked the software industry. In Snowbird, the Manifesto for Agile Software Development[b] and Principles Behind the Agile Manifesto[c] were born and endorsed by all 17 attendees, spelling out their values like this:

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- ***Individuals and interactions*** *over processes and tools;*
- ***Working software*** *over comprehensive documentation;*
- ***Customer collaboration*** *over contract negotiation; and*
- ***Responding to change*** *over following a plan.*

That is, while there is value in the items below (not bold), we value the items above (bold) more.

The Agile Manifesto and the agile principles thus began to serve as a rallying cry for some and the bull's-eye in the dartboard for others. "Religious" methodology wars ensued between the agilists and those supporting what came to be known as "plan driven,"[2] methodologies, the term that came to be used for "not agile" methodologies.

These wars have since subsided. Observations at international agile conferences indicate that companies in all industrial domains have generally come to coexist peacefully with agile methodologies. Many have embraced them, while some use many agile practices and others just a few. Meanwhile, agile practices have evolved, with new ones emerging and others fading away.

So how well do the Agile Manifesto and its 12 principles still capture what is valued by practicing software engineers in industry and by teams that have adopted agile methodologies as their own practices have matured and evolved? How do agile teams regard the principles today? Here, "agile teams" refers to teams claiming to use an agile software development methodology.

b http://agilemanifesto.org/
c http://agilemanifesto.org/principles.html

» key insights

- **The 12 original agile principles created by 17 software engineers in 2001 defined the agile trend that continues to transform the entire software industry.**
- **Rather than view one another solely as competition, these same engineers also wrote the Agile Manifesto, cooperatively focusing on their common interest in agile development and greatly magnifying any of their potential individual contributions.**
- **Supported by this foundation, agile practices used by software development teams today continue to evolve to address ever-changing user expectations and development team challenges.**

Surveys

I conducted two surveys in 2010 at North Carolina State University to weigh the community's view of the principles and use of associated practices. I administered them through surveymonkey.com, advertising the first survey on a number of agile-related user groups (such as those on Yahoo! and LinkedIn). Additionally, I emailed approximately 100 personal contacts, inviting them to participate and forward the survey to their colleagues. Respondents from the first survey could optionally provide their email address if they wanted me to send aggregated results of the survey. When respondents received these results, I further invited them to participate in a follow-on survey.

The first survey focused on the original principles and commonly used software development practices, as of 2010, beginning the first set of questions with the following instruction, followed by a list of the principles in random order:

How important is this principle that comes from the original agile principles authored in 2001 for agile teams in 2010? (1=not very important; 5=essential, the team is not agile if it doesn't follow this principle)

I began the second set of questions with the following instruction, followed by a list of 45 software development practices typically associated with agile:

What practices are essential for a team to be considered agile? (1=not important; 5=essential, a team is not agile unless it does this practice)

With each set of questions, I offered respondents space to provide textual commentary to augment their quantitative responses.

The first survey was completed by 326 respondents with extensive experience in agile software development (see the figure here). Those indicating they had been using an agile methodology for 10 years or more were using what came to be called an "agile methodology" post-Manifesto. Respondents were primarily from North America (59%) and Europe (29%). Of the 326, 18 (55%) indicated they worked on teams with 30 or more members; 313 (96%) worked in a distributed fashion, with 110 (34%) having teams all in the same country, 42 (13%) all in the same continent, and 160 (49%) spread across different continents; and 52 (16%) indicated they worked on safety-critical projects.

I based the follow-on survey on the optional textual commentary provided by respondents of the first survey (see Table 1), distilling the most common comments from the first survey in a revised agile principle and asking their opinion of the revised principle. The motivation behind creating the suggested revision was to highlight emerging industry trends and possible missing subtleties and/or evolution of the original principles; for example, I changed the original principle "Working software is the primary measure of progress" to "Valuable, high-quality software is the primary measure of progress at the end of each short, timeboxed iteration." The follow-on survey sought feedback on the revised principle, though my intent was not to replace the original principle. Respondents in the follow-on survey reflected roughly the same characteristics as participants in the first survey in terms of professional experience and geographic location, with 93 of the original 326 respondents providing feedback on each of the revised principles.

The second survey also asked how valuable respondents considered the principles, as well as "Why are the agile principles valuable?," letting respondents pick as many responses as they felt were applicable, along with the opportunity to provide additional comments.

The personal comments in the survey's "Other" category are best represented by this one: "The purpose of any principle is to provide a simple, clear source of guidance and inspiration. The agile principles are important because they distill the values of 'agile' into as little text as possible. By review-

ing them as we consider implementation specifics, we can make sure our day-to-day processes are serving the purposes that presumably we decided we wanted to meet."

Original Principles

My discussion here highlights the most noteworthy of the 93 textual comments from both surveys and the supporting data analysis rather than looking to explain each principle. Note that 11 of the 12 had a mean score of 4.1 out of 5 or higher, indicating a high level of support for principles that had been spelled out 10 years earlier.

Tier One (mean 4.6)

Principle 1 (standard deviation 0.8). Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Principle 3 (standard deviation 0.7). Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.

Respondents' commentary emphasized delivery of a solution with "high business value" to a customer early and often, along with willingness to respond to feedback. One respondent suggested that principles 1 and 3 were probably redundant, a view supported by statistics based on overall survey responses. The Pearson's *r* values for these two principles was 0.31, among the highest correlations between any two principles.

Tier Two (mean 4.5)

Principle 5 (standard deviation 0.9). Build projects around motivated individuals. Give them an environment and support they need, and trust them to get the job done.

Principle 7 (standard deviation 0.8). Working software is the primary measure of progress.

Principle 12 (standard deviation 0.8). The team regularly reflects on how to be more effective, tuning and adjusting its behavior accordingly.

Respondents' comments concerning principle 5 emphasized the need to empower and respect motivated individuals while making them "feel they can make a difference and [are] part of building something out of the ordinary." Some respondents said providing the "support they need" included removing obstacles so the team could operate efficiently.

Principle 7 attracted the most comments, with respondents saying that it, in particular, along with the full set of principles, in general, did not adequately emphasize the need to produce high-quality software and test and elicit nonfunctional requirements. The short-term, functional focus of iterations can lead to trouble. "Flaccid Scrum"[d] is the term coined by Martin Fowler, a noted author and speaker on software development, to refer to teams using only Scrum's project-management practices without also following sound engineering practices. Progress eventually slows for all Flaccid Scrum teams, according to Fowler, because they have not paid enough attention to the quality of the code. In some cases, only the easiest scenario of a feature (often called the "happy path") is demonstrated at the end of an iteration. The feature can then be considered "done," with project focus then turning to implementing a new set of features. Teams today more often define and adhere to sound "done criteria," stipulating the quality and testing steps a team must take before a feature is considered done (see Table 2).

Reacting to principle 12, respondents showed strong enthusiasm for holding retrospectives at least every iteration if not more often "for feedback and creating a culture of continuous improvement and building respect."

Tier Three (mean 4.4)

Principle 9 (standard deviation 0.8). Continuous attention to technical excellence and good design enhances agility.

Respondents gave strong support for this principle but provided no further commentary or clarification of their views.

Tier Four (mean 4.3)

Principle 2 (standard deviation 0.8). Welcome changing requirements even late in development; agile processes harness change for the customer's competitive advantage.

Principle 10 (standard deviation 1.0).

Survey respondents' experience with agile software development.

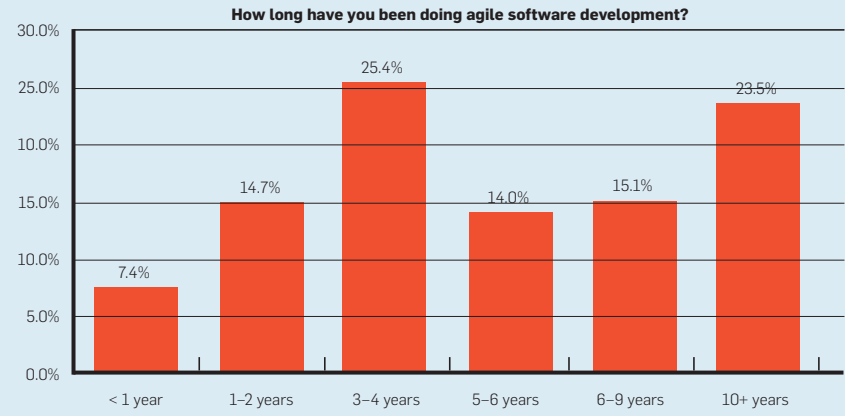


Table 1. Value of agile principles.

Why are the agile principles valuable?	Response Percent
They aren't really; the important thing is to use agile practices.	9.8%
They aren't really; the important thing is to look at the Agile Manifesto.	0%
Because they guide teams new to agile.	48.8%
Because all agile teams choose among software development practices, but, if they want to be agile, they should choose practices that are in line with the principles.	64.6%
They aren't really; no one looks at them anyway.	0%
Other [please comment]	20.7%

d http://www.martinfowler.com/bliki/FlaccidScrum.html

Simplicity, the art of maximizing the amount of work not done, is essential.

Some commenters on principle 2 suggested a project's requirements should change only at the beginning of each iteration. Agile methodologies aim to reduce waste associated with "thrashing," or progress implementing a feature, stopping and starting and maybe never being complete due to constantly changing priorities. This wasted effort can be reduced through a rule stating that once a feature is started, it must be completed.

One commenter described principle 10 as "Build great software...that addresses users' needs without unnecessary features."

Table 2. Agile principles.

	Mean	Standard Deviation
Continuous integration	4.5	0.8
Short iterations (30 days or less)	4.5	0.8
"Done" criteria	4.5	0.8
Automated tests run with each build	4.4	0.9
Automated unit testing	4.4	0.9
Iteration reviews/demos	4.3	0.8
"Potentially shippable" features at the end of each iteration	4.3	0.9
"Whole" multidisciplinary team with one goal	4.3	0.8
Synchronous communication	4.4	0.9
Embracing changing requirements	4.3	0.8
Features in iteration are customer-visible/customer-valued	4.3	0.8
Prioritized product backlog	4.4	0.9
Retrospective	4.2	1.0
Collective ownership of code	4.2	0.9
Sustainable pace	4.2	0.8
Refactoring	4.2	1.0
"Complete" feature testing done during iteration	4.1	0.9
Negotiated scope	4.1	0.9
Stand up/Scrum meeting	4.1	1.1
Timeboxing	4.1	1.1
Test-driven development unit testing	4.0	1.0
Just-in-time requirements elaboration	4.0	1.0
Small teams (12 people or less)	4.0	1.1
Emergent design	4.0	1.0
Configuration management	4.0	1.2
Daily customer/product manager involvement	3.9	1.0
Release planning	3.9	1.1
Test-driven development acceptance testing	3.8	1.0
Team documentation focuses on decisions rather than planning	3.8	1.2
Informal design; no big design up front	3.7	1.0
Co-located team	3.6	1.1
Team velocity	3.6	1.1
Requirements written as informal stories	3.6	1.1
10-minute build	3.6	1.3
Task planning	3.5	1.2
Coding standard	3.5	1.2
Kanban	3.4	1.6
Acceptance tests written by product manager	3.4	1.2
Pair programming	3.3	1.2
Burndown charts	3.3	1.3
Code inspections	3.2	1.3
Design inspections	3.3	1.3
Planning Poker	3.1	1.4
Stabilization iterations	3.0	1.5

Tier Five (mean 4.1)

Principle 4 (standard deviation 1.0). Businesspeople and developers must work together daily throughout the project.

Principle 6 (standard deviation 1.0). The most effective method of conveying information to and within a development team is face-to-face conversation.

Principle 8 (standard deviation 0.9). Agile processes promote sustainable development; sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Concerning principle 4 several respondents said that developers (often seen as those writing the code) should not be the only ones to work with businesspeople (a.k.a product owners). Rather, the whole team, including user-interface analysts, testers, project managers, developers, and businesspeople should collaborate. Others commenters said, "Every day often isn't realistic, nor is it necessarily needed."

Principle 6 was generally supported, though some commenters said the "requirement for face-to-face conversation is a severely limiting factor for distributed teams, and it seems to be a generational issue as well." In today's connected world, synchronous communication through instant messaging, Voice over IP, and WebEx may effectively stand in for face-to-face communication.

Several representative comments on principle 8 indicating the commenters' negative experience with relatively intense iterations ad infinitum:

"Agile does not promote sustainable development but increases the kind of focus that leads to burnout";

"Sustainable pace is extremely important, but we also sometimes have to slow down and think about things a little";

"Emphasize scheduled downtime as part of sustainable pace"; and

"The team should have dedicated exploratory study time that contributes to its ability to produce innovation."

Tier Six (mean 3.8)

Principle 11 (standard deviation 1.0). The best architectures, requirements, and designs emerge from self-organizing teams.

Concerning principle 11, several commenters suggested the need for a release vision and that teams

should understand how the product "contribute[s] to the larger goals of the [user] organization." The principles do not explicitly state that a release plan must be developed, with agile teams often beginning their iterations without such a vision communicated to the whole team. However, agile methodologies have always advocated producing a feasible, prioritized release backlog that also serves as the release vision. One commenter said, "You really need to do some systems engineering when building large systems," while some agilists may consider such systems engineering the equivalent of "big design up front."

Respondents also commented that the lean software development[9] concept of minimizing work-in-process was missing from the principles but still important for agile teams. An emerging lean trend in agile software development that does not appear in the original principles is the use of kanban, or signboard or billboard in Japanese, as a possible replacement for iterations and, in general, a focus on limiting work-in-process. However, this trend is not inconsistent with the original principles. Finally, commenters also said the principles did not cover planning, learning, and collaboration, and communication was not emphasized enough.

Overall, the results of both surveys suggested overwhelming support for the original principles, even after more than 10 years of use. However, survey commenters also said three concepts were missing from the principles but had still been part of the agile software development methodologies from the beginning. First, two principles included the term "developer" in places where the intended connotation was more likely the "whole team." Second, the principles did not explicitly say that a release vision would be created prior to starting incremental development. Finally, the principles did not insist that the working software produced should be valuable and of high quality, though this notion has been part of agile since it was first laid out.

Principle 6 (standard deviation 1.0). The most effective method of conveying information to and within a development team is face-to-face conversation.

Revised Principles

I assimilated the most common comments from the first survey to revise the original principles through the follow-on survey seeking feedback on the revisions, including:

Principle 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. [no change]

Principle 2. Welcome changing requirements at the start of each iteration, even late in development; agile processes harness change for the customer's competitive advantage.

Principle 3. [delete; redundant with Principle 1]

Principle 4. The whole team, from businesspeople through testers, must communicate and collaboratively work together throughout the project.

Principle 5. Build projects around empowered, motivated individuals with a shared vision of success; give them the environment and support they need, clear their external obstacles, and trust them to get the job done.

Principle 6. The most efficient, effective method for conveying information to and within a development team is through synchronous communication; important decisions are documented so are not forgotten.

Principle 7. Valuable, high-quality software is the primary measure of progress at the end of each short time-boxed iteration.

Principle 8. Agile processes promote sustainable development. The whole team should be able to maintain a reasonable work pace that includes dedicated time for exploration, visioning, refactoring, and obtaining and responding to feedback.

Principle 9. Continuous attention to technical excellence and good design enhances agility. [no change]

Principle 10. Simplicity—the art of maximizing the amount of work not done—is essential. [no change]

Principle 11. The best architectures, requirements, and designs emerge from self-organizing teams guided by a vision for product release.

Principle 12. With each iteration, the team candidly reflects on the success of the project, feedback, and how to be more effective, then tunes and adjusts its plans and behavior accordingly.

The 93 respondents to the second survey also provided 164 textual comments on the principles. About 11% of the comments (18 of 164) indicated the respondents preferred fewer words, as in "keep it simple." The revised principles are often longer than the original.

However, the follow-on survey indicated general agreement with the revisions, with two exceptions:

First, the most prominent reaction among survey commenters was when the word "iterations" was added to a principle (such as in revised principles 2, 7, and 12). The recent introduction of the lean software development kanban[7] practice removed the notion of iterations for many teams. With kanban, a feature can begin at any time if the "pull system" indicates the team has the capacity to start new work. As a result, kanban teams often lack defined iterations.

Second, many commenters also reacted negatively to the switch from "face-to-face communication" to "synchronous communication." Despite the fact (discussed earlier) that 96% of survey respondents worked on distributed teams and the assertion by one commenter that the change was "a nice update for the digital world," survey respondents generally emphasized that nothing beats face-to-face for verbal and non-verbal communication alike, and wanted principle 6 to represent the ideal practice.

Agile Practices

Table 2 lists the results of the first survey, which asked whether agile practices are essential for a team to be considered agile; each agile practice is followed by the mean response and the standard deviation of the responses, with 1 indicating the practice is not very important and 5 that the practice is essential for agile teams. Note that the practices at the top of the list generally have a lower standard deviation (connoting greater consistency among survey respondents) than those at the bottom of the list.

Many original agile practices (such as continuous integration and short iterations) are often at the top of such a list, while the more recent, emergent practices (such as Planning Poker, kanban, and stabilization iterations) are at the bottom. Planning Poker[5] is a Wideband Delphi[3] practice for estimating team-based effort. Stabilization iterations (generally two weeks) can occur at the end of all feature-producing iterations or periodically throughout a longer release cycle. During stabilization iterations, testers can perform additional integration, regression, and performance testing; the defect backlog can be reduced; the product backlog can be more intensively groomed; and some preliminary architecture, design, and dependency analysis for the next group of iterations can take place. Some teams find that stabilization iterations mid-release reduce burnout and provide time for exploration and learning.

An agile practice that survey respondents said was left off the list was the "spike"; that is, teams do spikes when they do not know enough about a feature to effectively estimate the resources needed for its implementation. A spike is a timeboxed experiment that allows developers to learn just enough about something unknown about a feature implementation (such as a new technology) to be able to estimate the effort required to deliver the feature.

Conclusion

The authors of the Agile Manifesto and the original 12 principles spelled out the essence of the agile trend that has transformed the software industry over more than a dozen years. That is, they nailed it.

Acknowledgements

Funding for my two-part study was provided by the Scrum Alliance (http://www.scrumalliance.org/). Many thanks to the North Carolina State University Realsearch group (www.realsearchgroup.org) and to the survey respondents. C

References

1. Beck, K. *Extreme Programming Explained: Embrace Change, Second Edition.* Addison-Wesley, Reading, MA, 2005.
2. Boehm, B. and Turner, R. Using risk to balance agile and plan-driven methods. *IEEE Computer 36*, 6 (June 2003), 57–66.
3. Boehm, B.W. *Software Engineering Economics.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.
4. Cockburn, A. *Agile Software Development.* Addison Wesley Longman, 2001.
5. Grenning, J. *Planning Poker or How to Avoid Analysis Paralysis while Release Planning,* 2002; http://renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf
6. Highsmith, J. *Adaptive Software Development.* Dorset House, New York, 1999.
7. Kniberg, H. and Skarin, M. *Kanban and Scrum: Making the Most of Both.* C4Media, Lexington, KY, 2010.
8. Palmer, S.R. and Felsing, J.M. *A Practical Guide to Feature-Driven Development.* Prentice Hall PTR, Upper Saddle River, NJ, 2002.
9. Poppendieck, M. and Poppendieck, T. *Lean Software Development.* Addison Wesley, Boston, 2003.
10. Schwaber, K. and Beedle, M. *Agile Software Development with SCRUM.* Prentice-Hall, Upper Saddle River, NJ, 2002.
11. Stapleton, J. *DSDM: The Method in Practice, Second Edition.* Addison Wesley Longman, 2003.

Laurie Williams (williams@csc.ncsu.edu) is a professor of computer science in the Department of Computer Science at North Carolina State University, Raleigh, NC, and an agile trainer and coach.

DOI:10.1145/2133806.2133826

Surveying a suite of algorithms that offer a solution to managing large document archives.

BY DAVID M. BLEI

Probabilistic Topic Models

AS OUR COLLECTIVE knowledge continues to be digitized and stored—in the form of news, blogs, Web pages, scientific articles, books, images, sound, video, and social networks—it becomes more difficult to find and discover what we are looking for. We need new computational tools to help organize, search, and understand these vast amounts of information.

Right now, we work with online information using two main tools—search and links. We type keywords into a search engine and find a set of documents related to them. We look at the documents in that set, possibly navigating to other linked documents. This is a powerful way of interacting with our online archive, but something is missing.

Imagine searching and exploring documents based on the themes that run through them. We might "zoom in" and "zoom out" to find specific or broader themes; we might look at how those themes changed through time or how they are connected to each other. Rather than finding documents through keyword search alone, we might first find the theme that we are interested in, and then examine the documents related to that theme.

For example, consider using themes to explore the complete history of the New York Times. At a broad level, some of the themes might correspond to the sections of the newspaper—foreign policy, national affairs, sports. We could zoom in on a theme of interest, such as foreign policy, to reveal various aspects of it—Chinese foreign policy, the conflict in the Middle East, the U.S.'s relationship with Russia. We could then navigate through time to reveal how these specific themes have changed, tracking, for example, the changes in the conflict in the Middle East over the last 50 years. And, in all of this exploration, we would be pointed to the original articles relevant to the themes. The thematic structure would be a new kind of window through which to explore and digest the collection.

But we do not interact with electronic archives in this way. While more and more texts are available online, we simply do not have the human power to read and study them to provide the kind of browsing experience described above. To this end, machine learning researchers have developed *probabilistic topic modeling*, a suite of algorithms that aim to discover and annotate large archives of documents with thematic information. Topic modeling algorithms are statistical methods that analyze the words of the original texts to discover the themes that run through them, how those themes are connected to each other, and how they change over

» key insights

- **Topic models are algorithms for discovering the main themes that pervade a large and otherwise unstructured collection of documents. Topic models can organize the collection according to the discovered themes.**
- **Topic modeling algorithms can be applied to massive collections of documents. Recent advances in this field allow us to analyze streaming collections, like you might find from a Web API.**
- **Topic modeling algorithms can be adapted to many kinds of data. Among other applications, they have been used to find patterns in genetic data, images, and social networks.**

time. (See, for example, Figure 3 for topics found by analyzing the *Yale Law Journal*.) Topic modeling algorithms do not require any prior annotations or labeling of the documents—the topics emerge from the analysis of the original texts. Topic modeling enables us to organize and summarize electronic archives at a scale that would be impossible by human annotation.

Latent Dirichlet Allocation

We first describe the basic ideas behind *latent Dirichlet allocation* (LDA), which is the simplest topic model.[8] The intuition behind LDA is that documents exhibit multiple topics. For example, consider the article in Figure 1. This article, entitled "Seeking Life's Bare (Genetic) Necessities," is about using data analysis to determine the number of genes an organism needs to survive (in an evolutionary sense).

By hand, we have highlighted different words that are used in the article. Words about *data analysis*, such as "computer" and "prediction," are highlighted in blue; words about *evolutionary biology*, such as "life" and "organism," are highlighted in pink; words about *genetics*, such as "sequenced" and "genes," are highlighted in yellow. If we took the time to highlight every word in the article, you would see that this article blends genetics, data analysis, and evolutionary biology in different proportions. (We exclude words, such as "and" "but" or "if," which contain little topical content.) Furthermore, knowing that this article blends those topics would help you situate it in a collection of scientific articles.

LDA is a statistical model of document collections that tries to capture this intuition. It is most easily described by its generative process, the imaginary random process by which the model assumes the documents arose. (The interpretation of LDA as a probabilistic model is fleshed out later.)

We formally define a *topic* to be a distribution over a fixed vocabulary. For example, the *genetics* topic has words about genetics with high probability and the *evolutionary biology* topic has words about evolutionary biology with high probability. We assume that these topics are specified before any data has been generated.[a] Now for each document in the collection, we generate the words in a two-stage process.

- Randomly choose a distribution over topics.
- For each word in the document
  a. Randomly choose a topic from the distribution over topics in step #1.
  b. Randomly choose a word from the corresponding distribution over the vocabulary.

This statistical model reflects the intuition that documents exhibit multiple topics. Each document exhibits the topics in different proportion (step #1); each word in each document is drawn from one of the topics (step #2b), where the selected topic is chosen from the per-document distribution over topics (step #2a).[b]

In the example article, the distribution over topics would place probability on *genetics*, *data analysis*, and

a Technically, the model assumes that the topics are generated first, before the documents.

b We should explain the mysterious name, "latent Dirichlet allocation." The distribution that is used to draw the per-document topic distributions in step #1 (the cartoon histogram in Figure 1) is called a *Dirichlet distribution*. In the generative process for LDA, the result of the Dirichlet is used to *allocate* the words of the document to different topics. Why *latent*? Keep reading.

Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.

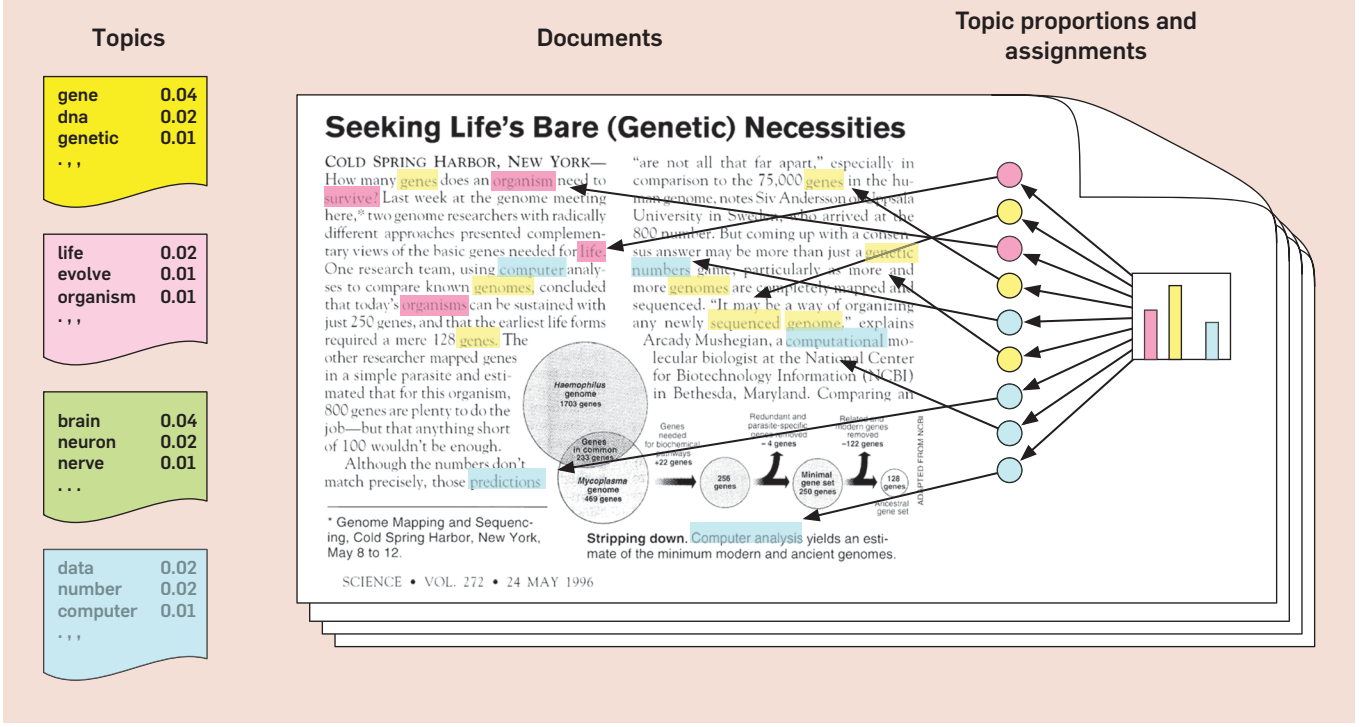
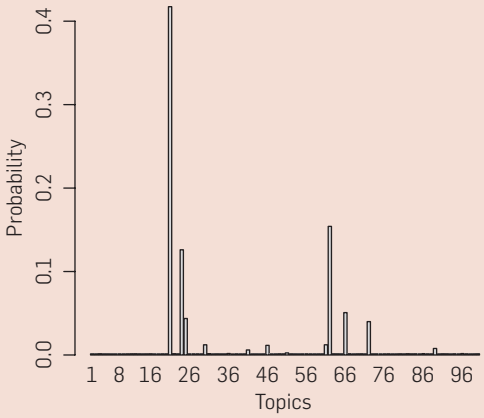


Figure 2. Real inference with LDA. We fit a 100-topic LDA model to 17,000 articles from the journal *Science*. At left are the inferred topic proportions for the example article in Figure 1. At right are the top 15 most frequent words from the most frequent topics found in this article.



"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

evolutionary biology, and each word is drawn from one of those three topics. Notice that the next article in the collection might be about *data analysis* and *neuroscience*; its distribution over topics would place probability on those two topics. This is the distinguishing characteristic of latent Dirichlet allocation—all the documents in the collection share the same set of topics, but each document exhibits those topics in different proportion.

As we described in the introduction, the goal of topic modeling is to automatically discover the topics from a collection of documents. The documents themselves are observed, while the topic structure—the topics, per-document topic distributions, and the per-document per-word topic assignments—is *hidden structure*. The central computational problem for topic modeling is to use the observed documents to infer the hidden topic structure. This can be thought of as "reversing" the generative process—what is the hidden structure that likely generated the observed collection?

Figure 2 illustrates example inference using the same example document from Figure 1. Here, we took 17,000 articles from *Science* magazine and used a topic modeling algorithm to infer the hidden topic structure. (The algorithm assumed that there were 100 topics.) We then computed the inferred topic distribution for the example article (Figure 2, left), the distribution over topics that best describes its particular collection of words. Notice that this topic distribution, though it can use any of the topics, has only "activated" a handful of them. Further, we can examine the most probable terms from each of the most probable topics (Figure 2, right). On examination, we see that these terms are recognizable as terms about genetics, survival, and data analysis, the topics that are combined in the example article.

We emphasize that the algorithms have no information about these subjects and the articles are not labeled with topics or keywords. The interpretable topic distributions arise by computing the hidden structure that likely generated the observed collection of documents.[c] For example, Figure 3 illustrates topics discovered from *Yale Law Journal*. (Here the number of topics was set to be 20.) Topics about subjects like genetics and data analysis are replaced by topics about discrimination and contract law.

The utility of topic models stems from the property that the inferred hidden structure resembles the thematic structure of the collection. This interpretable hidden structure annotates each document in the collection—a task that is painstaking to perform by hand—and these annotations can be used to aid tasks like information retrieval, classification, and corpus exploration.[d] In this way, topic modeling provides an algorithmic solution to managing, organizing, and annotating large archives of texts.

LDA and probabilistic models. LDA and other topic models are part of the larger field of *probabilistic modeling*. In generative probabilistic modeling, we treat our data as arising from a generative process that includes *hidden variables*. This generative process defines a *joint probability distribution* over both the observed and hidden random variables. We perform data analysis by using that joint distribution to compute the *conditional distribution* of the hidden variables given the

c Indeed calling these models "topic models" is retrospective—the topics that emerge from the inference algorithm are interpretable for almost any collection that is analyzed. The fact that these look like topics has to do with the statistical structure of observed language and how it interacts with the specific probabilistic assumptions of LDA.

d See, for example, the browser of *Wikipedia* built with a topic model at http://www.sccs.swarthmore.edu/users/08/ajb/tmve/wiki100k/browse/topic-list.html.

Figure 3. A topic model fit to the *Yale Law Journal*. Here, there are 20 topics (the top eight are plotted). Each topic is illustrated with its top-most frequent words. Each word's position along the *x*-axis denotes its specificity to the documents. For example "estate" in the first topic is more specific than "tax."

observed variables. This conditional distribution is also called the *posterior distribution*.

LDA falls precisely into this framework. The observed variables are the words of the documents; the hidden variables are the topic structure; and the generative process is as described here. The computational problem of inferring the hidden topic structure from the documents is the problem of computing the posterior distribution, the conditional distribution of the hidden variables given the documents.

We can describe LDA more formally with the following notation. The topics are $\beta_{1:K}$, where each β_k is a distribution over the vocabulary (the distributions over words at left in Figure 1). The topic proportions for the dth document are θ_d, where $\theta_{d,k}$ is the topic proportion for topic k in document d (the cartoon histogram in Figure 1). The topic assignments for the dth document are z_d, where $z_{d,n}$ is the topic assignment for the nth word in document d (the colored coin in Figure 1). Finally, the observed words for document d are w_d, where $w_{d,n}$ is the nth word in document d, which is an element from the fixed vocabulary.

With this notation, the generative process for LDA corresponds to the following joint distribution of the hidden and observed variables,

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right). \quad (1)$$

Notice that this distribution specifies a number of dependencies. For example, the topic assignment $z_{d,n}$ depends on the per-document topic proportions θ_d. As another example, the observed word $w_{d,n}$ depends on the topic assignment $z_{d,n}$ and *all* of the topics $\beta_{1:K}$. (Operationally, that term is defined by looking up as to which topic $z_{d,n}$ refers to and looking up the probability of the word $w_{d,n}$ within that topic.)

These dependencies define LDA. They are encoded in the statistical assumptions behind the generative process, in the particular mathematical form of the joint distribution, and—in a third way—in the *probabilistic graphical model* for LDA. Probabilistic graphical models provide a graphical language for describing families of probability distributions.[e] The graphical model for LDA is in Figure 4. These three representations are equivalent ways of describing the probabilistic assumptions behind LDA.

In the next section, we describe the inference algorithms for LDA. However, we first pause to describe the short history of these ideas. LDA was developed to fix an issue with a previously developed probabilistic model *probabilistic latent semantic analysis* (pLSI).[21] That model was itself a probabilistic version of the seminal work on *latent semantic analysis*,[14] which revealed the utility of the singular value decomposition of the document-term matrix. From this matrix factorization perspective, LDA can also be seen as a type of principal component analysis for discrete data.[11,12]

Posterior computation for LDA. We now turn to the computational

e The field of graphical models is actually more than a language for describing families of distributions. It is a field that illuminates the deep mathematical links between probabilistic independence, graph theory, and algorithms for computing with probability distributions.[35]

problem, computing the conditional distribution of the topic structure given the observed documents. (As we mentioned, this is called the *posterior*.) Using our notation, the posterior is

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D} \mid w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})}. \quad (2)$$

The numerator is the joint distribution of all the random variables, which can be easily computed for any setting of the hidden variables. The denominator is the *marginal probability* of the observations, which is the probability of seeing the observed corpus under any topic model. In theory, it can be computed by summing the joint distribution over every possible instantiation of the hidden topic structure.

That number of possible topic structures, however, is exponentially large; this sum is intractable to compute.[f] As for many modern probabilistic models of interest—and for much of modern Bayesian statistics—we cannot compute the posterior because of the denominator, which is known as the *evidence*. A central research goal of modern probabilistic modeling is to develop efficient methods for approximating it. Topic modeling algorithms—like the algorithms used to create Figures 1 and 3—are often adaptations of general-purpose methods for approximating the posterior distribution.

Topic modeling algorithms form an approximation of Equation 2 by adapting an alternative distribution over the latent topic structure to be close to the true posterior. Topic modeling algorithms generally fall into two categories—sampling-based algorithms and variational algorithms.

Sampling-based algorithms attempt to collect samples from the posterior to approximate it with an empirical distribution. The most commonly used sampling algorithm for topic modeling is *Gibbs sampling*, where we construct a *Markov chain*—a sequence of random variables, each dependent on the previous—whose limiting distribution is the posterior. The Markov chain is defined on the hidden topic variables for a particular corpus, and the algorithm is to run the chain for a long time, collect samples from the limiting distribution, and then approximate the distribution with the collected samples. (Often, just one sample is collected as an approximation of the topic structure with

f More technically, the sum is over all possible ways of assigning each observed word of the collection to one of the topics. Document collections usually contain observed words at least on the order of millions.

Figure 4. The graphical model for latent Dirichlet allocation. Each node is a random variable and is labeled according to its role in the generative process (see Figure 1). The hidden nodes—the topic proportions, assignments, and topics—are unshaded. The observed nodes—the words of the documents—are shaded. The rectangles are "plate" notation, which denotes replication. The *N* plate denotes the collection words within documents; the *D* plate denotes the collection of documents within the collection.

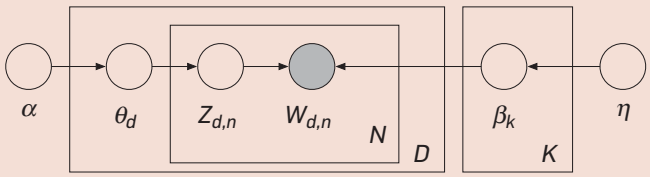
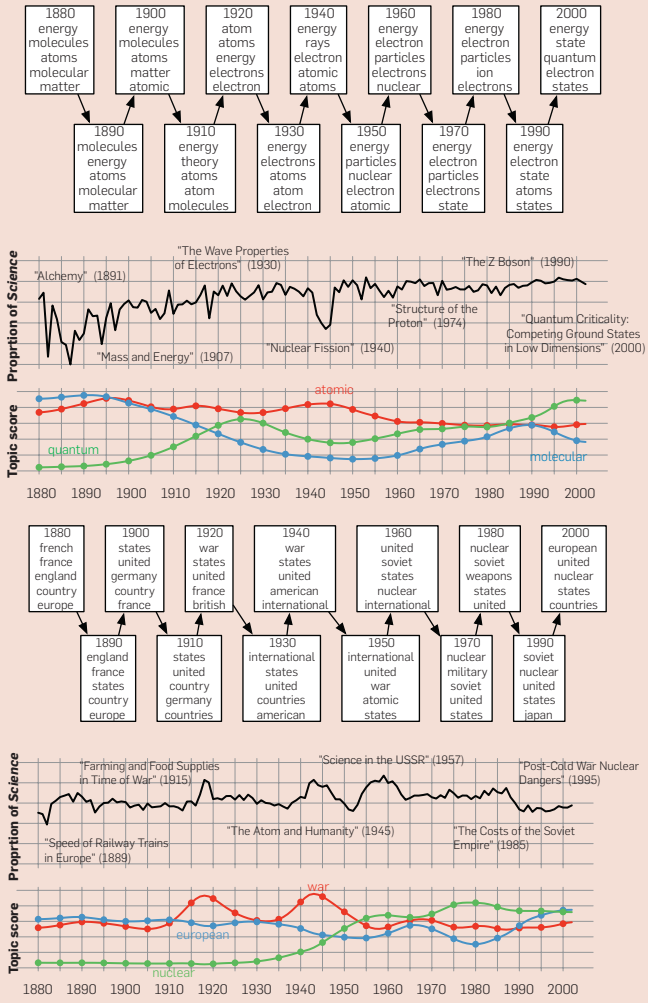


Figure 5. Two topics from a dynamic topic model. This model was fit to *Science* from 1880 to 2002. We have illustrated the top words at each decade.



maximal probability.) See Steyvers and Griffiths[33] for a good description of Gibbs sampling for LDA, and see http://CRAN.R-project.org/package=lda for a fast open-source implementation.

Variational methods are a deterministic alternative to sampling-based algorithms.[22,35] Rather than approximating the posterior with samples, variational methods posit a parameterized family of distributions over the hidden structure and then find the member of that family that is closest to the posterior.[g] Thus, the inference problem is transformed to an optimization problem. Variational methods open the door for innovations in optimization to have practical impact in probabilistic modeling. See Blei et al.[8] for a coordinate ascent variational inference algorithm for LDA; see Hoffman et al.[20] for a much faster online algorithm (and open-source software) that easily handles millions of documents and can accommodate streaming collections of text.

Loosely speaking, both types of algorithms perform a search over the topic structure. A collection of documents (the observed random variables in the model) are held fixed and serve as a guide toward where to search. Which approach is better depends on the particular topic model being used—we have so far focused on LDA, but see below for other topic models—and is a source of academic debate. For a good discussion of the merits and drawbacks of both, see Asuncion et al.[1]

Research in Topic Modeling

The simple LDA model provides a powerful tool for discovering and exploiting the hidden thematic structure in large archives of text. However, one of the main advantages of formulating LDA as a probabilistic model is that it can easily be used as a module in more complicated models for more complicated goals. Since its introduction, LDA has been extended and adapted in many ways.

Relaxing the assumptions of LDA. LDA is defined by the statistical assumptions it makes about the corpus. One active area of topic modeling research is how to relax and extend these assumptions to uncover more sophisticated structure in the texts.

One direction for topic modeling is to develop evaluation methods that match how the algorithms are used. How can we compare topic models based on how interpretable they are?

One assumption that LDA makes is the "bag of words" assumption, that the order of the words in the document does not matter. (To see this, note that the joint distribution of Equation 1 remains invariant to permutation of the words of the documents.) While this assumption is unrealistic, it is reasonable if our only goal is to uncover the course semantic structure of the texts.[h] For more sophisticated goals—such as language generation—it is patently not appropriate. There have been a number of extensions to LDA that model words nonexchangeably. For example, Wallach[36] developed a topic model that relaxes the bag of words assumption by assuming that the topics generate words conditional on the previous word; Griffiths et al.[18] developed a topic model that switches between LDA and a standard HMM. These models expand the parameter space significantly but show improved language modeling performance.

Another assumption is that the order of documents does not matter. Again, this can be seen by noticing that Equation 1 remains invariant to permutations of the ordering of documents in the collection. This assumption may be unrealistic when analyzing long-running collections that span years or centuries. In such collections, we may want to assume that the *topics* change over time. One approach to this problem is the dynamic topic model[5]—a model that respects the ordering of the documents and gives a richer posterior topical structure than LDA. Figure 5 shows a topic that results from analyzing all of *Science* magazine under the dynamic topic model. Rather than a single distribution over words, a topic is now a sequence of distributions over words. We can find an underlying theme of the collection and track how it has changed over time.

A third assumption about LDA is that the number of topics is assumed

g Closeness is measured with *Kullback–Leibler divergence*, an information theoretic measurement of the distance between two probability distributions.

h As a thought experiment, imagine shuffling the words of the article in Figure 1. Even when shuffled, you would be able to glean that the article has something to do with genetics.

known and fixed. The Bayesian nonparametric topic model[34] provides an elegant solution: the number of topics is determined by the collection during posterior inference, and furthermore, new documents can exhibit previously unseen topics. Bayesian nonparametric topic models have been extended to hierarchies of topics, which find a tree of topics, moving from more general to more concrete, whose particular structure is inferred from the data.[3]

There are still other extensions of LDA that relax various assumptions made by the model. The correlated topic model[6] and pachinko allocation machine[24] allow the occurrence of topics to exhibit correlation (for example, a document about *geology* is more likely to also be about *chemistry* than it is to be about *sports*); the spherical topic model[28] allows words to be *unlikely* in a topic (for example, "wrench" will be particularly unlikely in a topic about *cats*); sparse topic models enforce further structure in the topic distributions;[37] and "bursty" topic models provide a more realistic model of word counts.[15]

Incorporating metadata. In many text analysis settings, the documents contain additional information—such as author, title, geographic location, links, and others—that we might want to account for when fitting a topic model. There has been a flurry of research on adapting topic models to include metadata.

The author-topic model[29] is an early success story for this kind of research. The topic proportions are attached to authors; papers with multiple authors are assumed to attach each word to an author, drawn from a topic drawn from his or her topic proportions. The author-topic model allows for inferences about authors as well as documents. Rosen-Zvi et al. show examples of author similarity based on their topic proportions—such computations are not possible with LDA.

Many document collections are linked—for example, scientific papers are linked by citation or Web pages are linked by hyperlink—and several topic models have been developed to account for those links when estimating the topics. The *relational topic model* of Chang and Blei[13] assumes that each document is modeled as in LDA and that the links between documents depend on the distance between their topic proportions. This is both a new topic model and a new network model. Unlike traditional statistical models of networks, the relational topic model takes into account node attributes (here, the words of the documents) in modeling the links.

Other work that incorporates metadata into topic models includes models of linguistic structure,[10] models that account for distances between corpora,[38] and models of named entities.[26] General -purpose methods for incorporating metadata into topic models include Dirichlet-multinomial regression models[25] and supervised topic models.[7]

Other kinds of data. In LDA, the topics are distributions over words and this discrete distribution generates observations (words in documents). One advantage of LDA is that these choices for the topic parameter and data-generating distribution can be adapted to other kinds of observations with only small changes to the corresponding inference algorithms. As a class of models, LDA can be thought of as a *mixed-membership model* of grouped data—rather than associating each group of observations (document) with one component (topic), each group exhibits multiple components in different proportions. LDA-like models have been adapted to many kinds of data, including survey data, user preferences, audio and music, computer code, network logs, and social networks. We describe two areas where mixed-membership models have been particularly successful.

In population genetics, the same probabilistic model was independently invented to find ancestral populations (for example, originating from Africa, Europe, the Middle East, among others) in the genetic ancestry of a sample of individuals.[27] The idea is that each individual's genotype descends from one or more of the ancestral populations. Using a model much like LDA, biologists can both characterize the genetic patterns in those populations (the "topics") and identify how each individual expresses them (the "topic proportions"). This model is powerful because the genetic patterns in ancestral populations can be hypothesized, even when "pure" samples from them are not available.

LDA has been widely used and adapted in computer vision, where the inference algorithms are applied to natural images in the service of image retrieval, classification, and organization. Computer vision researchers have made a direct analogy from images to documents. In document analysis, we assume that documents exhibit multiple topics and the collection of documents exhibits the same set of topics. In image analysis, we assume that each image exhibits a combination of visual patterns and that the same visual patterns recur throughout a collection of images. (In a preprocessing step, the images are analyzed to form collections of "visual words.") Topic modeling for computer vision has been used to classify images,[16] connect images and captions,[4] build image hierarchies,[2,23,31] and other applications.

Future Directions

Topic modeling is an emerging field in machine learning, and there are many exciting new directions for research.

Evaluation and model checking. There is a disconnect between how topic models are evaluated and why we expect topic models to be useful. Typically, topic models are evaluated in the following way. First, hold out a subset of your corpus as the test set. Then, fit a variety of topic models to the rest of the corpus and approximate a measure of model fit (for example, probability) for each trained model on the test set. Finally, choose the model that achieves the best held-out performance.

But topic models are often used to organize, summarize, and help users explore large corpora, and there is no technical reason to suppose that held-out accuracy corresponds to better organization or easier interpretation. One open direction for topic modeling is to develop evaluation methods that match how the algorithms are used. How can we compare topic models based on how interpretable they are?

This is the *model checking* problem. When confronted with a new corpus and a new task, which topic model should I use? How can I decide which of the many modeling assumptions are important for my goals? How should I move between the many kinds of topic models that have been developed? These questions have been given some attention by statisticians,[9,30] but they have been scrutinized less for the scale

of problems that machine learning tackles. New computational answers to these questions would be a significant contribution to topic modeling.

Visualization and user interfaces. Another promising future direction for topic modeling is to develop new methods of interacting with and visualizing topics and corpora. Topic models provide new exploratory structure in large collections—how can we best exploit that structure to aid in discovery and exploration?

One problem is how to display the topics. Typically, we display topics by listing the most frequent words of each (see Figure 2), but new ways of labeling the topics—by either choosing different words or displaying the chosen words differently—may be more effective. A further problem is how to best display a document with a topic model. At the document level, topic models provide potentially useful information about the structure of the document. Combined with effective topic labels, this structure could help readers identify the most interesting parts of the document. Moreover, the hidden topic proportions implicitly connect each document to the other documents (by considering a distance measure between topic proportions). How can we best display these connections? What is an effective interface to the whole corpus and its inferred topic structure?

These are user interface questions, and they are essential to topic modeling. Topic modeling algorithms show much promise for uncovering meaningful thematic structure in large collections of documents. But making this structure useful requires careful attention to information visualization and the corresponding user interfaces.

Topic models for data discovery. Topic models have been developed with information engineering applications in mind. As a statistical model, however, topic models should be able to tell us something, or help us form a hypothesis, about the data. What can we *learn* about the language (and other data) based on the topic model posterior? Some work in this area has appeared in political science,[19] bibliometrics,[17] and psychology.[32] This kind of research adapts topic models to measure an external variable of interest, a difficult task for unsupervised learning that must be carefully validated.

In general, this problem is best addressed by teaming computer scientists with other scholars to use topic models to help explore, visualize, and draw hypotheses from their data. In addition to scientific applications, such as genetics and neuroscience, one can imagine topic models coming to the service of history, sociology, linguistics, political science, legal studies, comparative literature, and other fields, where texts are a primary object of study. By working with scholars in diverse fields, we can begin to develop a new interdisciplinary computational methodology for working with and drawing conclusions from archives of texts.

Summary

We have surveyed *probabilistic topic models*, a suite of algorithms that provide a statistical solution to the problem of managing large archives of documents. With recent scientific advances in support of unsupervised machine learning—flexible components for modeling, scalable algorithms for posterior inference, and increased access to massive datasets—topic models promise to be an important component for summarizing and understanding our growing digitized archive of information. C

References

1. Asuncion, A., Welling, M., Smyth, P., Teh, Y. On smoothing and inference for topic models. In *Uncertainty in Artificial Intelligence* (2009).
2. Bart, E., Welling, M., Perona, P. Unsupervised organization of image collections: Taxonomies and beyond. *Trans. Pattern Recognit. Mach. Intell. 33,* 11 (2010) (2301–2315).
3. Blei, D., Griffiths, T., Jordan, M. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J. ACM 57,* 2 (2010), 1–30.
4. Blei, D., Jordan, M. Modeling annotated data. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2003), ACM Press, 127–134.
5. Blei, D., Lafferty, J. Dynamic topic models. In *International Conference on Machine Learning* (2006), ACM, New York, NY, USA, 113–120.
6. Blei, D., Lafferty, J. A correlated topic model of Science. *Ann. Appl. Stat., 1,* 1 (2007), 17–35.
7. Blei, D., McAuliffe, J. Supervised topic models. In *Neural Information Processing Systems* (2007).
8. Blei, D., Ng, A., Jordan, M. Latent Dirichlet allocation. *J. Mach. Learn. Res. 3* (January 2003), 993–1022.
9. Box, G. Sampling and Bayes' inference in scientific modeling and robustness. *J. Roy. Stat. Soc. 143*, 4 (1980), 383–430.
10. Boyd-Graber, J., Blei, D. Syntactic topic models. In *Neural Information Processing Systems* (2009).
11. Buntine, W. Variational extensions to EM and multinomial PCA. In *European Conference on Machine Learning* (2002).
12. Buntine, W., Jakulin, A. Discrete component analysis. *Subspace, Latent Structure and Feature Selection.* C. Saunders, M. Grobelink, S. Gunn, and J. Shawe-Taylor, Eds. Springer, 2006.
13. Chang, J., Blei, D. Hierarchical relational models for document networks. *Ann. Appl. Stat. 4*, 1 (2010).
14. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R. Indexing by latent semantic analysis. *J. Am. Soc. Inform. Sci. 41*, 6 (1990), 391–407.
15. Doyle, G., Elkan, C., Accounting for burstiness in topic models. In *International Conference on Machine Learning* (2009), ACM, 281–288..
16. Fei-Fei, L., Perona, P. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Vision and Pattern Recognition* (2005), 524–531.
17. Gerrish, S., Blei, D. A language-based approach to measuring scholarly impact. In *International Conference on Machine Learning* (2010).
18. Griffiths, T., Steyvers, M., Blei, D., Tenenbaum, J. Integrating topics and syntax. *Advances in Neural Information Processing Systems 17.* L. K. Saul, Y. Weiss, and L. Bottou, eds. MIT Press, Cambridge, MA, 2005, 537–544.
19. Grimmer, J. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Polit. Anal. 18*, 1 (2010), 1.
20. Hoffman, M., Blei, D., Bach, F. On-line learning for latent Dirichlet allocation. In *Neural Information Processing Systems* (2010).
21. Hofmann, T. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence (UAI)* (1999).
22. Jordan, M., Ghahramani, Z., Jaakkola, T., Saul, L. Introduction to variational methods for graphical models. *Mach. Learn. 37* (1999), 183–233.
23. Li, J., Wang, C., Lim, Y., Blei, D., Fei-Fei, L., Building and using a semantivisual image hierarchy. In *Computer Vision and Pattern Recognition* (2010).
24. Li, W., McCallum, A. Pachinko allocation: DAG-structured mixture models of topic correlations. In *International Conference on Machine Learning* (2006), 577–584.
25. Mimno, D., McCallum, A. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Uncertainty in Artificial Intelligence* (2008).
26. Newman, D., Chemudugunta, C., Smyth, P. Statistical entity-topic models. In *Knowledge Discovery and Data Mining* (2006).
27. Pritchard, J., Stephens, M., Donnelly, P. Inference of population structure using multilocus genotype data. *Genetics 155* (June 2000), 945–959.
28. Reisinger, J., Waters, A., Silverthorn, B., Mooney, R. Spherical topic models. In *International Conference on Machine Learning* (2010).
29. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smith, P., The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence* (2004), AUAI Press, 487–494.
30. Rubin, D. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Ann. Stat. 12*, 4 (1984), 1151–1172.
31. Sivic, J., Russell, B., Zisserman, A., Freeman, W., Efros, A., Unsupervised discovery of visual object class hierarchies. In *Conference on Computer Vision and Pattern Recognition* (2008).
32. Socher, R., Gershman, S., Perotte, A., Sederberg, P., Blei, D., Norman, K. A Bayesian analysis of dynamics in free recall. In *Advances in Neural Information Processing Systems 22.* Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds, 2009.
33. Steyvers, M., Griffiths, T. Probabilistic topic models. *Latent Semantic Analysis: A Road to Meaning.* T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, eds. Lawrence Erlbaum, 2006.
34. Teh, Y., Jordan, M., Beal, M., Blei, D. Hierarchical Dirichlet processes. *J. Am. Stat. Assoc. 101*, 476 (2006), 1566–1581.
35. Wainwright, M., Jordan, M. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn. 1*(1–2) (2008), 1–305.
36. Wallach, H. Topic modeling: Beyond bag of words. In *Proceedings of the 23rd International Conference on Machine Learning* (2006).
37. Wang, C., Blei, D. Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. *Advances in Neural Information Processing Systems 22.* Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds. 2009, 1982–1989.
38. Wang, C., Thiesson, B., Meek, C., Blei, D. Markov topic models. In *Artificial Intelligence and Statistics* (2009).

David M. Blei (blei@cs.princeton.edu) is an associate professor in the computer science department of Princeton University, Princeton, N.J.

ICMI 2012

Oct 22-26, Santa Monica, CA

ICMI is the premier international forum for multidisciplinary research on multimodal human-human and human-computer interaction, interfaces, and system development.

Important Dates:

Workshop proposals:	March 1, 2012
Special session proposals:	March 7, 2012
Paper & demo submission:	May 4, 2012
Author notification:	July 23, 2012
Camera-ready:	August 20, 2012
Main Conference:	Oct 23-25, 2012
Workshops:	Oct 22, 26, 2012

Call for Papers: ICMI 2012 will feature a single-track main conference including: keynote speakers, technical full and short papers (including oral and poster presentations), special sessions, demonstrations, exhibits and doctoral spotlight papers. Topics of interest include, but are not limited to:

Multimodal interaction processing Machine learning, pattern recognition, and signal processing approaches for the analysis and modeling of multimodal interaction between people and among the different modalities within people; adaptation and multimodal input fusion and output generation, addressing any combination of: vision, gaze, audio, speech, smell/olfaction, taste, gestures, pen, haptic and tangible, bio-signals such as brain activity and skin conductivity.

Interactive systems and applications Mobile and ubiquitous systems, automotive and navigation systems, human-robot and human-virtual agent interaction, virtual and augmented reality, education, authoring, entertainment, gaming, telepresence, assistive and prosthetic systems, brain-computer interfaces, universal access, healthcare, biometry, intelligent environments, meeting analysis and meeting spaces, indexing, retrieval and summarization, etc.

Data, evaluation, and standards for multimodal interactive systems Design issues, principles, and best practices and authoring techniques for human-machine interfaces using any combinations of input and/or output multiple modalities. Architectures; assessment techniques and methodologies; corpora; annotation and browsing of multimodal interactive data; W3C and other standards for multimodal interaction and interfaces; evaluation techniques for multimodal systems.

Modeling human communication patterns The modalities and applications named above drive a need for multimodal models of human-human and human-machine communication, including verbal and non-verbal interaction, affordances of different modalities, multimodal discourse and dialogue modeling, modeling of culture as it pertains to multimodality, long-term multimodal interaction, multimodality in social and affective interaction, multimodal social signal processing.

Photo Credit:
Patrick Smith Photography

http://www.acm.org/icmi/2012

DOI:10.1145/2133806.2133825

A research agenda for making the smart grid a reality.

BY SARVAPALI D. RAMCHURN, PERUKRISHNEN VYTELINGUM, ALEX ROGERS, AND NICHOLAS R. JENNINGS

Putting the 'Smarts' into the Smart Grid: A Grand Challenge for Artificial Intelligence

THE PHENOMENAL GROWTH in material wealth experienced in developed countries throughout the 20th century has largely been driven by the availability of cheap energy derived from fossil fuels (originally coal, then oil, and most recently natural gas). However, the continued availability of this cheap energy cannot be taken for granted given the growing concern that increasing demand for these fuels (and particularly, demand for oil) will outstrip our ability to produce them (so-called 'peak oil').[9] Many mature oil and gas fields around the world have already peaked and their annual production is now steadily declining. Predictions of when world oil production will peak vary between 0–20 years into the future, but even the most conservative estimates provide little scope for complacency given the significant price increases that peak oil is likely to precipitate.[1] Furthermore, many of the oil and gas reserves that do remain are in environmentally or politically sensitive regions of the world where threats to supply create increased price volatility (as evidenced by the 2010 Deepwater Horizon disaster and ongoing unrest in the Middle East). Finally, the growing consensus on the long-term impact of carbon emissions from burning fossil fuels suggests that even if peak oil is avoided, and energy security assured, a future based on fossil fuel use will expose regions of the world to damaging climate change that will make the lives of many of the world's poorest people even harder.[15]

Against this background, many governments around the world have begun taking action to transition to a low carbon economy. For example, the U.K. has legislated to reduce CO_2 emissions by 80% by 2050 (compared to 1990 levels).[8] Achieving this aim requires that the direct use of fossil fuels that we are familiar with today is almost entirely eliminated. Thus, the use of electric vehicles (EVs) and high-speed electric trains will have to become widespread in order to reduce

» key insights

- **To safeguard the quality of life of future generations, it is absolutely essential to build cleaner and more efficient electrical grids.**
- **The smart grid represents a vision of a future electricity grid, radically different to those currently deployed, where the bidirectional flow of both electricity and information allows demand to be actively managed in real time, such that electricity can be generated at scale from intermittent renewable sources.**
- **Delivering this decentralized, autonomous, and intelligent system represents a Grand Challenge for computer science and artificial intelligence research, and while its impact lies in the future, we need to start work today.**

ILLUSTRATION BY PAUL PRICE


AN91060Ø

our reliance on oil for transportation.[a] Likewise, our homes and offices will have to be heated by efficient ground and air source heat pumps powered by electricity rather than existing natural gas and oil fired boilers.[22] As a result (and given the general growth of the world economy), electricity demand across the world is predicted to increase by 76%, or 4,800 gigawatts (GW), by 2030 (compared to 2007 levels).[20] Crucially, much of the electricity needed to meet this demand will have to be generated from renewable wind, solar, and tidal sources rather than the coal and natural gas power plants that we use today.

It is this increased demand for electricity, and the requirements for its generation, that present perhaps the greatest challenge. In most countries, the electricity grid has changed very little since it was first installed, and all existing grids are predicated on the central idea that electricity is produced by a relatively small number of large fossil fuel burning power stations and is delivered to a much larger number of customers, often some distance from these generators, on-demand. The grid itself relies on ageing infrastructure (for example, 40-year-old transmission lines and transformers, and 20-year-old power stations), is plagued by poor information flow (for example, most domestic electricity meters are read at intervals of several months), and has significant inefficiencies arising from losses within the transmission (on a national level) and distribution (on a local level) networks.[12]

The vision of an electricity grid that makes extensive use of renewable generation challenges this current situation. Renewable generation is both intermittent and distributed, with the output of such generators being determined by local environmental conditions (such as wind speeds and cloud cover in the case of wind turbines and photovoltaic (PV) solar panels, respectively) that can vary significantly over minutes and hours. Thus, it will no longer be possible for supply to continuously follow the vagaries of consumer demand, but rather, the demand-side will have to be managed to ensure that demand for electricity is matched against the available supply. EVs will play a part in this, since not only do they represent a significant extra load that must be satisfied, but more positively, they also provide a distributed form of energy storage,[b] which may allow the grid to smooth out this variable supply.

Furthermore, meeting the increased demand for renewable generation may require hundreds of thousands, or even millions of such generators, distributed across both the transmission and distribution networks. These generators may need to act together, effectively working as virtual power plants (VPPs), or may be located on every building across the grid, resulting in a distributed network of prosumers[c] who both produce and consume electricity depending on their local requirements. Thus, unlike existing grids where electricity generally flows one-way from generators to consumers, this will result in flows of electricity that vary in magnitude and direction continuously. To guarantee the security of the network (such as, the maintenance of stable voltages and frequencies, and the reliability of supply) and to avoid the cascading failures that plague today's grid,[d] new control procedures must be devised. Indeed, the number and variability of generators will require that the grid is able to act autonomously, under human supervision but not necessarily under human control, to diagnose potential problems and self-heal.

Thus, there is a growing consensus that existing grids cannot simply be extended to address these challenges, but rather, a fundamental reengineering of the grid is required; one that envisages the creation of a 'smart grid', described by the U.S. Department of Energy[12] as: *A fully automated power delivery network that monitors and controls every customer and node, ensuring a two-way flow of electricity and information between the power plant and the appliance, and all points in between. Its distributed intelligence, coupled with broadband communications and automated control systems, enables real-time market transactions and seamless interfaces among people, buildings, industrial plants, generation facilities, and the electric network.*

What is perhaps most striking about this vision is that not only does it present many challenges in terms of power systems engineering, telecommunications, and cybersecurity, but at its core are concepts, such as distributed intelligence, automation, and information exchange, that have long been the focus of research within the computer science and the artificial intelligence (AI) communities. In this article, we argue that the smart grid provides significant new challenges for research in AI since smart grid technologies will require algorithms and mechanisms that can solve problems involving a large number of highly heterogeneous actors (for example, consumers with different demand profiles or generators with different volatilities), each with their own aims and objectives, having to operate within significant levels of uncertainty (such as, where the network conditions and the outcome of actions taken by individual entities on the grid will be more unpredictable or uncontrollable) and dynamism (where demand and supply at different points in the network will be in a significant state of flux). Hence, we illustrate how such issues arise within the key components of the smart grid—demand-side management, EVs, VPPs, the emergence of prosumers, and self-healing networks—and by showing which components and which interactions need to be smart, we provide a research agenda for this community for making the smart grid a reality.

a Electric motors are inherently more efficient than internal combustion engines, and are "future proof" in that their carbon emissions reduce as the electricity used to supply them becomes cleaner.

b Energy storage in existing grids is typically limited to a small number of pumped storage generators that pump water from a low reservoir to a high one when electricity is plentiful, and recover this potential energy by letting the water flow back through a turbine, when electricity is in short supply.

c The term "prosumer" was coined in 1970 by futurologist Alvin Toffler in his book *Future Shock* in order to describe the actors in the marketplace who would not just consume but also actively participate in the production of customized goods.

d The Northeast blackout of 2003 that forced the shutdown of over 100 power plants and affected 55 million people—the largest blackout in U.S. history—was precipitated by a single overloaded transmission line, in Ohio, sagging and touching overgrown vegetation.

Demand-Side Management

A key requirement for a safe and efficient electricity grid is that supply and demand are always in perfect balance. Now, in the day-to-day running of today's electricity grid, this is achieved by varying the supply side in real time to match demand (increasing and decreasing the output of generators such that voltage and frequency are maintained across the grid). Hence, the idea that electricity should be available at all times at the flick of a switch has permeated most, if not all, of our daily activities in the modern world.

However, as far back as the 1980s, Schweppe and colleagues highlighted numerous reasons why demand for electricity should be made more adaptive to supply conditions.[34] They noted that doing so would allow peaks in demand to be "flattened," thus allowing generation assets to be reduced; particularly, expensive (and carbon-intensive) peaking plants that might only be used for several hours or less each day. This flattening would result in longer term and cheaper production contracts, producing a more efficient grid with lower prices for consumers. Furthermore, it would also provide significant benefits for grid operators. For example, if generation capacity was temporarily restricted due to some unforeseen event (either due to faults or if renewable energy sources are unavailable), then controlling demand would ensure that those generators available were not overloaded. In addition, after a power failure has occurred, the ability to synchronize demand with supply as connections are recovered and generators are brought up to speed would significantly accelerate recovery from such failures (a point we discuss later).

The need for demand-side management is even more apparent within a grid that makes extensive use of intermittent renewable generation. In this case, there is a high likelihood that there will be periods when there is insufficient generation capacity to meet demand. It is thus imperative that demand can be reduced at these times. Conversely, there may also be times when renewable energy is plentiful, and demand should increase to make the best use of this energy.

Unlike existing grids where electricity generally flows one-way from generators to consumers, [the smart grid] will result in flows of electricity that vary in magnitude and direction continuously.

To date, approaches to reduce demand have been limited to either directly controlling the devices used by the consumers (for example, automatically switching off high-load devices such as air conditioners at peak times), or to providing customers with tariffs that deter peak time use of electricity. The advent of the smart grid with two-way information flows, and smart meters making real-time measurements of consumption, would allow demand-side management to be deployed at scale across the entire grid, providing every home and every commercial and industrial consumer with the ability to automatically reduce load in response to signals from the grid.

However, doing so may be ineffective, or at worst, detrimental, since such initiatives tend to reduce the natural diversity of consumers' peak demands and shift all of these peaks to specific periods.[36] For example, static time-of-use (TOU) pricing where the price of electricity at night is cheaper than during the day, has been observed to create significant additional peaks in demand as soon as the off-peak period is reached.[30,36] Similarly, critical peak pricing (CPP), which is often applied on the West Coast of the U.S to control air conditioners at peak times, can often create additional peaks as devices turn back on as soon as the critical period is over. Given this, a number of researchers have suggested that more sophisticated tariffs, such as real-time pricing (RTP) or spot pricing (where the price per kWh of electricity consumed is different for each half-hour and is provided to the consumer a day, or a few hours, ahead of time), in conjunction with more sophisticated 'agents' that can autonomously respond to these price signals, would avoid this.[34] However, even RTP can create unexpected peaks in demand, when all individuals respond to a signal in the same way, and inadvertently synchronize with others.[30]

Thus, it appears that demand-side management technologies that simply rely on reacting to control or price signals will not be enough. Rather, what is necessary are more sophisticated approaches that are truly adaptive to the state of the grid, that are able to learn the correct response given any particu-

lar situation, and that can look ahead and predict both supply and demand trends in the near future, in order to prepare for future reductions in available supply, or to make the most effective use of supply when it is available.

The design of such intelligent systems is challenged by the complexity of the domains in which they are deployed. For example, within a home, demand reduction may involve shifting the time of use of a number of electrical appliances, each with their own individual constraints (for example, lighting cannot be shifted, a washing machine can be shifted by a day or two, while a dishwasher may be shiftable by a few hours[24]). Similarly, both heating (given that this will likely be electrified through the use of efficient heat pumps) and cooling loads can be shifted as long as the comfort and temperature preferences of the householders are met. To be effective in this, it may also be necessary for such systems to learn the thermal properties of the home in which they are deployed, as well as the local weather conditions, and the way in which these local conditions impact on the heat loss, or gain, of the home. Crucially, these approaches will have to take into account the fact that each individual householder will have his or her own preferences, and these preferences must either be explicitly elicited, or learned. Since these preferences are likely to exhibit change over time, and depend on the current activities of the householder and local weather conditions, in computational terms this translates into an online learning and scheduling problem under uncertainty.

Similarly, commercial and industrial consumers will be constrained by existing contracts and commercial considerations (for example, a factory may have to deliver products within certain deadlines, while a data center has to be available to its customers 24 hours a day), and must balance demand reduction against these additional factors. Large industrial consumers of electricity with significant heating, cooling, or pumping loads may have considerable flexibility regarding when they actually consume electricity as long as some overarching constraints are satisfied.[e]

It will be important to design simulation systems that can accurately represent both the grid and the reaction of consumers, in order to predict the emergent properties of the system under a range of different conditions and worst-case scenarios.

However, to do so in a responsive way requires that the usage optimization algorithm that is deployed is able to model and predict both the prices within the grid, and also the industrial processes themselves (similar to the home heating scenario where a thermal model of the home must be learned). Furthermore, in both settings, it will be essential that the householders and business owners are able to understand the consequences of the automated actions that are taken, and are happy to delegate control to an intelligent device or software agent. In this respect, it will be important to define the adjustable autonomy of such systems; to what extent should the agent automatically decide to shift devices to run at certain times, and when should it ask for confirmation from the user.[33]

The development of these autonomous technologies raises the prospect that such systems will be widely deployed in possibly millions of homes, each individually reacting to prices and to the preferences of householders. Defining the convergence properties (that is, how the aggregate demand profile will respond to price signals) of such a complex system will be central to the definition of what constitutes safe and efficient behaviors for the grid. In particular, it will be necessary to ensure that neither significant inefficiencies, nor excessive volatility ensue from these autonomous systems converging to poor equilibria (or not converging at all). Hence, it will be important to design simulation systems that can accurately represent both the grid and the reaction of consumers, in order to predict the emergent properties of the system under a range of different conditions (for example, weather patterns or social activities) and worst-case scenarios (some generators fail or lines trip).

Against this background, recent work has begun to research the use of autonomous agents, representing individual consumers, that interact through markets,[10,40] and individually learn to optimize their use of electrical loads or storage devices in a number

e During the 2000 California electricity crisis, which saw extremely high spot prices, several bauxite smelters realized there was greater profit to be had in reselling electricity they had bought in long-term forward contracts, than in using it themselves to produce aluminium.[3]

of simplified settings.[28,30] Simulations of such systems point to the effectiveness of adaptive behaviors (that learn to react to prices) on the grid. In addition, human-computer interaction technologies have also been proposed to improve the reaction of users to the information from smart meters.[16,37] While promising, we believe this work represents only the beginnings of the research needed in this area.

Thus, in summary, we believe the key AI challenges in demand-side management are:

- Designing automation technologies for heterogeneous devices that learn to adapt their energy consumption against real-time price signals when faced with uncertainty in predictions of future demand and supply, the individual users' preferences, and the constraints of the overarching system (domestic, commercial, or industrial) within which it is deployed;
- Developing the means by which the automated decisions of these systems can be effectively communicated to, and controlled by, their human owners, while allowing a varying range of autonomous behaviors; and
- Developing simulation and prediction tools to allow the systemwide consequences of deploying pricing mechanisms and energy management agents to be assessed by grid operators and suppliers.

Electric Vehicles

With the advent of commercially viable EVs, such as the Nissan Leaf and the Chevy Volt, the coming years are likely to see the large-scale EV adoption that will shift the energy requirements of transport from fossil fuels to renewable electricity from the smart grid.[12,26] EVs are one of the key mechanisms to deliver significant reductions in carbon emissions as the transport sector is one of the largest contributors in most developed countries (approximately 20% in the U.K. and 30% in the U.S.), and the majority of these emissions are the result of private motor vehicles. As millions of EVs are deployed onto the roads, novel mechanisms, building upon the communication infrastructure and distributed intelligence in the smart grid, will be needed to ensure the batteries of these vehicles are fully charged when their owners need to use them, without overloading the network. In addition, these same batteries will form part of the decentralized demand-side management system used to reduce variations in demand and supply by charging when low-carbon renewable energy is plentiful, and discharging back into the grid when it is in short supply; so called vehicle-to-grid or V2G.

EVs place a considerable additional load on the grid due to the high charging rates that are necessary to ensure both a reasonable vehicle range of around 100 miles, and the ability to rapidly charge the battery. While a typical house may use between 20- to 50kWh of energy per day, an EV battery may be charged with 32kWh of energy in just a few hours.[18] Thus, the total energy required by these vehicles may be comparable to the total electricity consumption within the domestic sector, but all of this demand is likely to be concentrated over particular periods of the day, and over particular geographical areas; both of which are subject to shifts. For example, if all the EVs in a local neighborhood are charged at the same time (as is likely to happen as householders return home at the end of the day), the local distribution network, and in particular, the street-level transformer (which is typically undersized and allowed to cool over night), may become a significant bottleneck to supply. When the owners of these vehicles drive to work and plug in, the demand will shift in both time and geographic distribution. Similar issues occur when a large number of EVs simultaneously attend large-scale social events at sporting arenas or shopping malls.[26]

Given these continuously changing demands imposed on the local distribution network by the movement and charging of vehicles within it, and the variable supply of renewable energy, it will be necessary to devise sophisticated approaches to schedule the charging of EVs. This scheduling should make the most effective use of what renewable energy is available, while also ensuring the vehicles' batteries are fully charged when required by their owners. Furthermore, this must be done in the context of uncertainty regarding both the future availability of renewable energy, and future vehicle use. Building upon this, it will be important to design decentralized control mechanisms that can guide the charging of EVs to various points in the network, given its dynamic conditions and constraints. In particular, these mechanisms will have to take into account that consumers must be incentivized (for example, in terms of charging prices or speeds at specific points) to adapt their behavior as they may only care about their individual travel needs. The challenge is to ensure such incentives are properly designed to induce charging profiles that stabilize the grid (that is, ensure flows are secure and transformers are not overloaded) while satisfying the needs and preferences of the highly heterogeneous population of EVs each with their individual battery capacity, charging speeds, and usage pattern.

More positively, EVs will also be a key resource in the demand-side management systems discussed previously. In such systems, the ability to defer demand to times when renewable energy is more plentiful is essential, and currently, this is only possible with the subset of electrical loads that are not required to have immediate effect (for example, washing machines or dishwashers). However, the ability to store energy within large batteries allows any electrical load to be shifted, and we are likely to first see energy from EV batteries support the shifting of loads within their owners' home (vehicle-to-home or V2H), and then to providing energy back to the grid itself (V2G).[25,26,f] While the impact on the user's lifestyle of scheduling loads in the home may be minimized through the use of the EV battery, the scheduling of the battery charging and discharging cycles will need to ensure there is sufficient capacity to satisfy the loads in the home, and the travel needs of the vehicle's owner, while minimizing the cost of electricity used. Moreover, this schedule will need to be optimized for, and adapt to, the changing needs of the vehicle owner, the (real-time) price paid for feeding back to the grid, as well as the battery capacity and efficiency. Hence, such optimizations will also require learning algorithms to

f In addition to providing energy, the vehicles may also be able to provide regulation services to the grid to stabilize both the voltage and frequency of electricity.[31]

predict the pattern of use of the vehicle, and also the demand of the home.

Addressing these challenges requires intelligent systems that can fully automate the charging and discharging of these vehicles, while taking account of the current and future availability of the renewable generation, and being aware of the local constraints of the distribution network. Recent work has begun to address these challenges with online mechanism design being used to elicit users' travel requirements (that is, the amount of charge required and the time at which the EV is needed) and schedule the charging of their vehicles,[17] and suggestions to apply peak and dynamic pricing to shift demand across a city.[25] These mechanisms are likely to work and be of social value (that is, not impede the daily activities of the vehicle owners) only if they minimize waiting (charging) times for consumers and never leave consumers stranded. As such, these systems will have to draw on diverse sources of information, such as distribution network load information (for example, load on the lines, number of EVs connected at various positions and prices at different charge/discharge points), traffic information from road cameras, and geolocation services such as Google Latitude (http://latitude.google.com) or Facebook Places (http://www.facebook.com/places) that contain rich information that can be mined to predict future movements of consumers to specific locations and, hence, likely bottlenecks on specific lines and transformers in the system. Systems that can optimize the charging cycle of an EV by making sense of such a wide range of heterogeneous information sources are likely to play a key role in ensuring EVs are seamlessly integrated into the smart grid.

Thus, against this background, we identify the key AI challenges in the deployment of EVs in the smart grid as follows:

- Predicting individual users' EV charging needs based on data about their daily activities and travel needs;
- Predicting aggregate EV charging demands at different points in the network given the continuous movement of EVs, the available charge in their batteries, and the social activities their users engage in;
- Designing decentralized control mechanisms that coordinate the movement of EVs (each with different battery capacities and charging speeds) to different charge points by providing incentives to consumers to do so. The aim is to maintain secure flows on the grid and ensure transformers do not trip due to excess demand; and
- Designing algorithms to optimize the charging cycles of EVs to satisfy the predicted needs of the user (to shift loads or to travel) while maximizing the profits generated from participating in V2G sessions.

Virtual Power Plants

As larger numbers of actors (for example, EVs, homes, or renewable energy providers) in the smart grid communicate and coordinate with each other to control demand at different points in the network (for example, using demand-side management to ensure demand is able to follow the supply of renewable energy, and EVs discharging to the grid to cope with excess demand), it will be important to harness synergies that exist between them to improve the efficiency of the grid (EVs discharging to satisfy demand at times when demand-side management techniques cannot shift enough usage to later times). To this end, the concept of a VPP[2] has been proposed to capture the notion of a number of actors, coming together to sell electricity, as an aggregate.[g] However, several challenges arise in the formation and management of VPPs that coordinate a number of heterogeneous actors (EVs or renewable energy providers) to maximize the amount of energy delivered in the system while minimizing the costs and uncertainties in doing so. In particular, these individual actors must be able to come to an agreement in technical (that is, how they coordinate their consumption or production patterns) and economical (how they share the profits generated by the VPP) terms in order to maximize the value of the set of energy services (providing electricity, storing electricity, or shifting demand) they provide as a VPP.

The process of forming VPPs at a technical level means the individual actors must synchronize the largely heterogeneous services they provide within the VPP in an agile fashion to meet the requirements of the contracts they make with their customers. In particular, individual actors need to estimate the impact of their individual production (or demand reduction) on the aggregate performance of the VPP, and communicate and optimize the joint actions taken to meet the VPPs' objectives (that is, satisfy demand). These technical arrangements may need to be specified on a daily, and even on an hourly basis to maximize the profits of the individual actors. This is because if some actors can only produce energy at specific times of the day (for example, PVs generate energy during the day and tidal energy may be available at night), they will want to choose those partners they can complement better at those times (for example, a PV farm and a tidal generator may generate energy out of phase with each other and hence be highly complementary, while wind energy providers whose turbines are located in the same region will generate energy at the same time and hence be less complementary). In turn, if new actors become better partners due to changes in the environment (more wind blows at night resulting in higher predicted wind energy production than tidal or more EVs converge to a specific region due to a social event, resulting in more storage being available), then some of them might decide to leave their current VPP and form a new one (for example, PV owners may be better off storing their excess energy during the day in the EVs to be able to supply at night rather than collaborate with a tidal energy provider). Given the scale and dynamism of this optimization problem, it will be important to design decentralized coordination algorithms and strategies that allow individual VPP participants to come to the most efficient arrangements within a reasonable time. Moreover, they will need to ensure such arrangements do not overload the local distribution networks in which they are connect-

g The term "virtual power plant" is also used to describe companies that may not have any generation capacity and that simply buy generation capacity from a generator. We do not deal with such VPPs here.

ed. Given this, and the restrictions imposed by the network operator due to possible network congestion, the VPP may further have to re-optimize individual members' operations. Typically, such optimizations would have to be done while being confronted with uncertainty about the individual members' generation and consumption capacity.

The negotiation of technical arrangements must take into account that each potential member of a VPP is typically motivated to maximize its own profit, even though, as a group they compete against other actors (individuals, VPPs, or large power stations) in the system to maximize the group's profits. Therefore, it is in each actor's interest to take actions that will cost it the least while maximizing its share of the profits obtained by the VPP operations as a whole. This leaves some room for any individual resource to manipulate what it reveals as its predicted capability (such as, production, demand-response, or storage ability) as opposed to what it actually delivers on the day. For example, given their uncertainty about their production, some resources may prefer to understate their predicted production profile in case they get penalized by the group for underproducing. Alternatively, some resources may prefer to overstate their predicted production in the case that penalties for underproducing are not significant, and doing so increases their share of the profits. Such strategic considerations highlight the need to capture the provenance of decisions made by the VPP, such that it is possible to track and verify the individual actions, reports, and resulting rewards of each VPP member. The amount of provenance information this will generate will require efficient frameworks and mechanisms to represent, store, audit, and share it. Building upon provenance information it may then be possible to model the trustworthiness of individual VPP members through trust and reputation mechanisms similar to those used in online marketplaces, such as eBay or Amazon.[29] These mechanisms would, in turn, need to be designed to ensure they are robust to wrong or manipulative reports so that security measures can then be taken to ensure those actors with low trust do not cause significant disruption to the network in case they do not fulfill their part of the VPPs' operations.

It will be important to design decentralized coordination algorithms and strategies that allow individual VPP participants to come to the most efficient arrangements within a reasonable time.

Assuming trust and reputation mechanisms can render VPPs reliable, it is important to ensure the negotiations that individual energy providers engage in converge in such a way that the most efficient VPPs (those generating the maximum social welfare) are most effectively formed (that is, in minimum time and with minimum communication costs) in the system.[11] Here, convergence is achieved when all the members of the VPP are satisfied with their share of the profits generated. The strategic and computational aspects of such negotiation processes are typically studied within multi-agent systems using tools such as cooperative game theory[4] to partition the profits of groups among their members and combinational optimization algorithms to partition actors into the most efficient groupings for the system respectively.[27] However, the VPP formation process presents a number of unique challenges for AI research. In particular, given that all actors are connected in a network where flows are limited on each line, the actions (energy production or consumption) taken by each actor or VPP restricts the actions (to different degrees) of all VPPs in the system. The formation of each VPP can have significant externalities (for example, the flows created by one VPP can congest some lines, which, in turn, may prevent other VPPs from using energy sources or providing energy to consumers at the nodes connected to those lines). Moreover, the fact that each VPP compounds the uncertainty in production of each member (for example, due to uncertainty in the weather forecast or demand-side managed consumption) renders the VPP formation process highly stochastic.

All these issues will require the definition of computationally efficient search algorithms to allocate the payoffs to individual members of VPPs (as defined by game-theoretic solution concepts), while taking into account uncertainty in defining the relative contributions of each member to the aggregate performance

(that is, mainly the profits generated) of the VPP. Moreover, given that different coalitions may be formed over time, an energy provider will choose its membership of coalitions in such a way as to maximize its revenues in the long run. This makes the search for efficient payoff allocations exponentially harder since it extends the search space to include future possible coalitions (and their expected returns) as well as present ones. Initial work in applying multi-agent systems approaches to the VPP formation process include Chalkiadakis et al.[5] that provides solutions to the formation of VPPs of wind turbines with uncertain production and Dimeas and Hatziargyriou[13] that presents an agent-based framework for VPP formation. These approaches, however, are still at a preliminary stage.

To advance the state of the art in this domain, the following key AI challenges still need to be addressed:

- Designing agent-based models of different VPP actors and processes in order to capture the complexity of the technical arrangements needed to form and manage VPPs;
- Distributed combinatorial optimization of the technical arrangements of demand-side management, V2G sessions, and micro-generation, to maximize rewards;
- Designing online mechanisms to form statistically correct trust measures for energy providers and automatically capture, track, and reason about the provenance of information revealed by energy providers to form VPPs; and
- Designing search algorithms and negotiation mechanisms for individual actors to agree on which VPP to form at different points in time and how to share the profits, using computationally efficient game-theoretic solution concepts, of a VPP given uncertainty in their performance, trust in their revealed capabilities, and changing weather and demand patterns.

Energy Prosumers

Our discussion so far has highlighted the significant heterogeneity of the large numbers of renewable energy resources in the smart grid and the complexity of the interactions between them and consumers. When taken altogether, this will necessitate significant changes in the way energy is bought and sold. This is set against the current operation of the grid where, in many countries (for example, the U.S., U.K., and in many parts of the EU), the electricity market is deregulated, such that large generators (located far from the point of use) trade directly with retailers who then sell the electricity to consumers through fixed contracts and tariffs.[19,35] In these countries, electricity is traded in forward and futures markets on a long-term ahead basis (weeks, months, seasons, and even years) and on day-ahead spot markets through a range of different contracts (for example, baseload, off-peak, or half-hourly contracts). Any real-time excess or shortfall in supply and demand with respect to contracted volume is settled in the balancing market (also termed the settlement process) where the price to buy and sell electricity is typically set by the market maker rather than being based on the direct matching between bids and offers in the day-ahead market.

The widespread adoption of renewable generation at the level of individual homes and businesses will lead to the creation of markets composed of many millions of prosumers who both produce and consume energy.

In contrast, market operations in the smart grid will have to adjust to a much larger number of heterogeneous entities, distributed throughout the network (closer to the point of use of electricity), trading much smaller amounts of energy. Indeed, the widespread adoption of renewable generation at the level of individual homes and businesses will lead to the creation of markets composed of many millions of prosumers who both produce and consume energy.[14] Given this, while some prosumers may try to find an agreement with other prosumers to form VPPs (and resort to cooperative game-theoretic solutions as discussed earlier), many will directly trade in the electricity market where the game-theoretic considerations are purely noncooperative. Hence, compared to typical consumers who are mainly concerned about optimizing their electricity usage and who are typically agnostic to the real-time conditions on the electricity market, prosumers will need to optimize both their production and consumption of energy in order to make trading decisions in real time, through Internet-based interfaces to spot or forward

markets, so they maximize the profits they can make by buying (to consume or store) and selling energy (either energy that they generate, or have stored earlier). By making their own localized trading decisions, prosumers may reduce the inefficiencies (added costs for end users and lower margins for generators) resulting from retailers hedging their energy purchases to minimize their exposure to risk (in the balancing market) and selling fixed long-term contracts to their consumers at high costs.

To do so, however, means prosumers must be endowed with effective trading strategies that can cope with uncertainty in the market. To minimize this uncertainty, they will need to be informed by predictions of their own demand (that may vary according to their needs and social activities) and generation capacity (for example, using weather forecasts or their EV usage needs), as well as the future price of electricity on the market. Given these trading decisions may need to be taken in real time, these predictions must also be generated in real time, and furthermore, to ensure users understand the lifestyle or operational implications of, and agree to, autonomously chosen trading decisions, human-computer interaction mechanisms will have to be designed to ensure large numbers of users trust and participate in these markets.

Essentially, as more prosumers populate the market, electricity will become a commodity with similar properties to those traded on stock markets. Given this, prosumers will be able to speculate in markets, buying and selling not simply to consume or supply electricity, but also to profit. However, while speculation may help make the market more efficient, it may also adversely impact the operation of the grid, if the traded flows do not actually satisfy the physical constraints of the distribution network. Potential solutions point to the application of regulatory measures to reduce speculation and more importantly, to congestion pricing mechanisms[39] within the distribution network, similar to the location-based pricing used within the transmission network in many parts of the U.S.[35] In such mechanisms, prices vary geographically throughout the network to ensure the flows of electricity within it do not exceed the limits of any of the transmission lines. To ensure these mechanisms do guarantee an efficient system it will be important to study the equilibrium conditions (for example, market efficiency, loads on transmission lines) resulting from the application of these congestion prices against significantly heterogeneous populations of prosumers.

In summary, the AI challenges involved in endowing prosumers with the intelligence to trade in electricity markets while ensuring safe network flows include:

- Developing computationally efficient learning algorithms that can accurately predict both the prosumers' consumption and generation profiles (instead of only the usage profile for a consumer) as well as the price of electricity in real time in order to inform profitable trading decisions;
- Developing autonomous trading agents that can use such predictions to maximize their profit in the electricity market, and efficient algorithms to marry congestion management with market operation in distribution networks while guaranteeing good equilibrium conditions in the system; and
- Developing human-agents interaction mechanisms to allow prosumers to guide their trading decisions that take into account the prosumers' daily constraints and preferences to consume or produce energy.

Self-Healing Networks

We have discussed a number of ways in which the electricity flows are likely to become both more unpredictable and bidirectional in the smart grid. This will result in a greater need for decentralized control strategies given the sheer number of active entities embedded in the system. While this renders fault-correction mechanisms in the network even more complex, the intelligence on which these active entities rely to make their consumption or generation decisions could also be used to naturally distribute (and hence make more robust) the decision making needed to apply self-healing strategies on the network when faults occur. Generally speaking, faults may arise either because lines become overloaded or because of old infrastructure becoming more prone to failure. To prevent such faults and remedy them, network operators already rely on a number of intelligent systems at the transmission network level. Traditionally, this is achieved with the help of automatic voltage regulators and using supervisory control and data acquisition systems[6] with phasor measure units[h] for situational awareness. Using such systems, active network management[21] techniques can help to automatically reconfigure the network and send control signals to individual generators to increase generation or to precontracted loads to reduce their consumption.[7] By endowing individual components on the network with the intelligence to apply these techniques, they can automatically correct faults as and when they occur and therefore let the network self-heal.

Extending these techniques to the management of the distribution network where large numbers of prosumers will operate will require a much larger number of phasor measure units to be deployed, both because the distribution network contains many more nodes, but also because the heterogeneity of the prosumers within it means network conditions are likely to vary more rapidly, necessitating accurate and timely monitoring and control. Fully instrumenting such networks is likely to be too expensive, and thus, there is a clear need for the development of state estimation systems that do not need to have every node in the network monitored. More importantly, we will need systems that can, using information gleaned from across the grid, learn correlations between state parameters at different nodes to provide accurate and robust estimates of the system state. The vast amount of data generated from multiple actors and sensors, and the microsecond-level measurements being made, will present formidable computational challenges in trying to estimate or predict the future state of the system.

h Phasor measurement units measure both magnitudes and phase angles of voltages and currents within the network, and are used to assess the state of a power system in real time.

If accurate information about the network can be obtained, active network management techniques, supported by distributed intelligence in the network, can help recover from faults faster than previously possible. For example, if voltages tend to drift in some parts of the network, automatic actions on transformers may be taken to reestablish the correct voltage levels, or assistance may be requested from EVs that are currently plugged into the network.[38] Furthermore, if faults are detected in one part of the network, that part of the system could be disconnected, leaving other independent parts running separated (that is, effectively 'islanded') provided they can sustain the balance between supply and demand (for example, using demand-side management). This could eventually avoid rolling blackouts or even help recover from those blackouts that do happen.

To build such self-healing mechanisms, however, will require that all these actors can communicate their action space (for example, limits on voltage regulation, generation capacity, demand reduction ability) and agree on joint actions to implement islanding strategies. Given the uncertainty that permeates the actions of some of these entities (weather patterns that affect generation or social activities that affect the movement of EVs), it will be important to predict the impact of such uncertainty on the joint actions chosen to avoid electing those that may result in cascading failures in the worst case. Moreover, given the individual preferences of all actors involved (to consume electricity for specific activities or to sell electricity to maximize profits) these joint actions may need to be negotiated rapidly among them to ensure they end up in an agreement all parties commit to.[23]

Initial approaches aiming to achieve this level of coordination express the problem as centralized (constrained) optimization problems that can be solved using (non) linear programming tools.[7] Clearly, centralizing active network management involving potentially thousands of different types of actors, each with their own energy generation and production requirements is unlikely to scale very well in both the communication and computation costs it incurs. Hence, more scalable decentralized planning approaches that rely on short range communication between individual actors (for example, distribution network nodes, consumers, and EVs) will be needed.[32,38]

The smart grid must be able to make efficient use of intermittent renewable energy sources and supply the additional electricity required by EVs; doing so will require extensive use of demand-side management and VPPs to balance supply and demand.

Hence, we summarize the AI challenges of self-healing mechanisms as follows:

- Designing computationally efficient state estimation algorithms that can predict voltage and phase information at different nodes in the (partially observable) distribution network, in real time, given the prosumers' current and predicted energy demand and supply;
- Enabling distributed coordination of automatic voltage regulators and energy providers and consumers for voltage control and balancing demand and supply during recovery from faults; and
- Automating distributed active network management strategies given the uncertainty (either because they cannot be accurately measured or there is incomplete information about certain nodes) about demand and supply at different points in the network.

Conclusion

There is a significant drive within the developed world to reduce our reliance on fossil fuels and move to a low-carbon economy in order to guarantee energy security and mitigate the impact of energy use on the environment. This transition requires a fundamental rethinking and reengineering of the electricity grid. The ensuing smart grid must be able to make efficient use of intermittent renewable energy sources and supply the additional electricity required by EVs; doing so will require extensive use of demand-side management and VPPs to balance supply and demand. It will also see large numbers of prosumers, buying and selling electricity in real time while automated network control algorithms maintain the safe operation of the grid and allow it to self-heal when something goes wrong.

The automation, information exchange, and distributed intelligence needed to deliver such technologies create many new challenges for the AI communities investigating machine learning, search, distributed control,

and optimization. In this article, we have enumerated what we believe are the main challenges that, if met, will allow the full potential of the smart grid to be realized. Our claims build upon an extensive survey of the state of the art that goes beyond the papers cited and includes a large number of references (spanning technical papers, books, and policy documents relating to the deployment of specific smart grid technologies and evaluations of these) provided in the online appendix. In particular, we have highlighted the key issues in learning and predicting demand or supply at various points in the network given the variety of demand control mechanisms (for example, demand-side management and EV charging) and energy sources, each with different degrees of uncertainty in their production capability (VPPs or renewable energy sources). Moreover, we showed that the automated decentralized coordination between such entities (to balance demand and supply while ensuring flows on the network are always secure) must factor in both the individual properties of all actors (EVs with different batteries, different types of renewable energy sources, users with their own understandings of trading decisions and their agents' decisions) involved and the incentives given to them to behave in certain ways (consumers shifting demand due to real-time pricing, or VPPs sharing profits equitably). Building upon this, we also examined some initial attempts at solving them within the various sub-areas of the smart grid.

Cutting across these various challenges are the issues of human-computer interaction, heterogeneity, dynamism, and uncertainty that are an intrinsic part of decision making and acting in the smart grid. By dealing effectively with these factors, we believe it will be possible for future generations to rely on their energy systems to deliver electricity efficiently, safely, and reliably.

Finally, we note that many of the issues present within the smart grid also arise within other domains such as water distribution, transportation, and telecommunication networks where large numbers of heterogeneous entities act and interact in a similar fashion to those within the grid. Hence, there is potential to transfer technologies across these domains and also address broader issues that affect the sustainability of such systems in a unified manner, such as cybersecurity and the ethics of delegating human decision making to intelligent systems.

Acknowledgments

The authors are supported by the iDEaS (www.ideasproject.info) and ORCHID (www.orchid.ac.uk) projects.

References

1. Aleklett, K., Höök, M., Jakobsson, K., Lardelli, M., Snowden, S. and Söderbergh, B. The peak of the oil age—Analyzing the world oil production Reference Scenario in World Energy Outlook 2008. *Energy Policy 38*, 3 (2010), 1398–1414.
2. Awerbuch, S. and Preston, A.M. *The Virtual Utility: Accounting, Technology and Competitive Aspects of the Emerging Industry.* Kluwer, Boston, MA, 1997.
3. Binczewski, G. The energy crisis and the aluminum industry: Can we learn from history? *Journal of the Minerals, Metals and Materials Society 54*, 2 (2002), 23–29.
4. Chalkiadakis, G. and Boutilier, C. Sequentially optimal repeated coalition formation under uncertainty. *Autonomous Agents and Multi-Agent Systems* (2010), 1–44.
5. Chalkiadakis, G., Robu, V., Kota, R., Rogers, A. and Jennings, N.R. Cooperatives of distributed energy resources for efficient virtual power plants. In *Proc. of the 10th Intl. Conf. on Autonomous Agents and Multiagent Systems* (May 2011), 787–794.
6. Chowdhury, S., Chowdhury, S. and Crossley, P. *Microgrids and Active Distribution Networks.* Institution of Engineering and Technology (IET), 2009.
7. Davidson, E., McArthur, S., Yuen, C. and Larsson, M. Aura-nms: Towards the delivery of smarter distribution networks through the application of multi-agent systems technology. *IEEE Power and Energy Society General Meeting* (2008), 1–6.
8. DECC. *The Climate Change Act 2008 Impact Assessment.* DECC, 2009.
9. Deffeyes, K.S. *Hubbert's peak: The impending world oil shortage.* Princeton Univ. Press, Princeton, NJ, 2008.
10. Deindl, M., Block, C., Vahidov, R. and Neumann, D. Load shifting agents for automated demand side management in micro energy grids. In *Proc. of the 2nd IIEEE Intl. Conf. on Self-Adaptive and Self-Organizing Systems* (2008), 487–488.
11. Demange, G. and Wooders, M. *Group formation in economics: networks, clubs and coalitions.* Cambridge Univ. Press, Cambridge, MA, 2005.
12. U.S. Department-Of-Energy. Grid 2030: A National Vision for Electricity's Second 100 Years. Tech. Report, Department of Energy, 2003.
13. Dimeas, A. and Hatziargyriou, N. Agent based control of virtual power plants. In *Proc. of the Intl. Conf. on Intelligent Systems Applications to Power Systems* (2007), 1–6.
14. EU SmartGrid Technology Platform. Vision and strategy for Europe's electricity networks of the future. Tech. Report, European Union, 2006.
15. Friedman, T. *Hot, Flat, and Crowded: Why We Need a Green Revolution–and How It Can Renew America.* Farrar, Straus & Giroux, 2008.
16. Froehlich, J., Findlater, L. and Landay, J. The design of eco-feedback technology. In *Proc. of the 28th Intl. Conf. on Human Factors in Computing Systems.* ACM, NY, 2010, 1999–2008.
17. Gerding, E., Robu, V., Stein, S., Parkes, D., Rogers, A. and Jennings, N.R. Online mechanism design for electric vehicle charging. In *Proc. of the 10th Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems* (May 2011), 811–818.
18. Green, R.C., Wang, L., and Alam, M. The impact of plug-in hybrid electric vehicles on distribution networks: A review and outlook. *Renewable and Sustainable Energy Reviews, 15*, 1 (2011) 544–553.
19. Harris, C. *Electricity Markets: Pricing, Structures, and Economics.* Wiley, NY, 2005.
20. International Energy Agency. World Energy Outlook 2009 Fact Sheet. Tech. Report, IEA, Paris, 2009.
21. MacDonald, R., Ault, G. and Currie, R. Deployment of active network management technologies in the UK and their impact on the planning and design of distribution networks. *SmartGrids for Distribution* (2009), 1–4.
22. MacKay, D. *Sustainable Energy: Without the Hot Air.* UIT, Cambridge, MA, 2009.
23. McDonald, J. Adaptive intelligent power systems: Active distribution networks. *Energy Policy 36*, 12 (2008), Foresight Sustainable Energy Management and the Built Environment Project, 4346–4351.
24. Mert, W., Suschek-Berger, J. and Tritthart, W. Consumer acceptance of smart appliances. Tech. Report. EIE project–Smart Domestic Appliances in Sustainable Energy Systems (Smart-A), 2008.
25. Mitchell, W., Borroni-Bird, C. and Burns, L. *Reinventing the Automobile.* MIT Press, Cambridge, MA, 2010.
26. RAE. Electric Vehicles: Charged with the Potential. Tech. Report. The Royal Academy of Engineering, 2010.
27. Rahwan, T., Ramchurn, S.D., Jennings, N.R. and Giovannucci, A. An anytime algorithm for optimal coalition structure generation. *Journal of Artif. Intel. Research 34* (Apr. 2009), 521–567.
28. Ramchurn, S., Vytelingum, P., Rogers, A., and Jennings, N.R. Agent-based homeostatic control for green energy in the smart grid. *ACM Transactions on Intelligent Systems and Technology 2*, 4 (May 2011).
29. Ramchurn, S.D., Huynh, T. and Jennings, N.R. Trust in multiagent systems. *The Knowledge Engineering Review 19*, 1 (2004), 1–25.
30. Ramchurn, S.D., Vytelingum, P., Rogers, A. and Jennings, N.R. Agent-based control for decentralised demand side management in the smart grid. In *Proc. of the 10th Intl. Conf. on Autonomous Agents and Multiagent Systems* (May 2011), 5–12.
31. Ribeiro, P., Johnson, B., Crow, M., Arsoy, A. and Liu, Y. Energy storage systems for advanced power applications. In *Proc. of the IEEE 89*, 12 (2001), 1744 –1756.
32. Rogers, A., Farinelli, A., Stranders, R. and Jennings, N.R. Bounded approximate decentralised coordination via the max-sum algorithm. *Artif. Intel. 175*, 2 (2011), 730–759.
33. Scerri, P., Pynadath, D. and Tambe, M. Towards adjustable autonomy for the real world. *Journal of Artif. Intel. Research 17*, 1 (2002), 171–228.
34. Schweppe, F., Daryanian, B. and Tabors, R. Algorithms for a spot price responding residential load controller. *Power Engineering Review 9*, 5 (1989), IEEE, 49–50.
35. Schweppe, F.C., Caramanis, M.C., Tabors, R.O. and Bohn, R.E. *Spot Pricing of Electricity.* Kluwer Academic Publishers, 1988.
36. Strbac, G. Demand side management: Benefits and challenges. *Energy Policy 36*, 12 (2008), 4419–4426.
37. Sundramoorthy, V., Cooper, G., Linge, N. and Liu, Q. Domesticating energy-monitoring systems: Challenges and design concerns. *IEEE Pervasive Computing 10* (2011), 20–27.
38. Vovos, P., Kiprakis, A., Wallace, A. and Harrison, G. Centralized and distributed voltage control: Impact on distributed generation penetration. *Power Systems, IEEE Transactions on ? 22*, 1 (2007), 476 –483.
39. Vytelingum, P., Voice, T.D., Ramchurn, S.D., Rogers, A. and Jennings, N.R. Agent-based micro-storage management for the smart grid. In *Proc. of the 9th Intl. Conf. on Autonomous Agents and MultiAgent Systems*, (May 2010), 39–46.
40. Ygge, F. Akkermans, J.M., Andersson, A., Krejic, M. and Boertjes, E. The HOMEBOTS system and field test: A multi-commodity market for predictive power load management. In *Proc. of the 4th Intl. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology 1* (1999), 363–382.

Sarvapali D. Ramchurn (sdr@ecs.soton.ac.uk) is a Lecturer at the University of Southampton, U.K.

Perukrishnen Vytelingum (pv@ecs.soton.ac.uk) is a Senior Research Fellow at the University of Southampton, U.K.

Alex Rogers (acr@ecs.soton.ac.uk) is a Reader at the University of Southampton, U.K.

Nicholas R. Jennings (nrj@ecs.soton.ac.uk) is a professor at the University of Southampton, U.K. and a Distinguished Adjunct Professor at King Abdulaziz University in Jeddah, Saudia Arbia.

Columbus State University
Assistant/Associate Professor in Computer Science

The TSYS School of Computer Science at Columbus State University invites applications for a tenure-track position in Computer Science. Responsibilities include delivering quality instruction in undergraduate and graduate courses, conducting research including student mentoring, student advising, developing and coordinating research proposals, and participating in university and community services. Applicants with interests in any field of Computer Science are encouraged to apply. Starting date is August 2012. Candidates are required to have an earned Ph.D. in Computer Science or a closely related field by the time of appointment (August 1, 2012). For more information about the school, visit our web site at http://cs.columbusstate.edu. Screening will begin immediately and continue until the position is filled. For a detailed job description, requirements and application information, please visit our website at http://hr.columbusstate.edu/jobs.asp or e-mail cs@columbusstate.edu.

CSU is an Affirmative Action/Equal Opportunity Employer, Committed to Diversity in Hiring.

Northern Arizona University
Lecturer in Computer Science

Northern Arizona University invites applications for at least one lecturer in Computer Science, to begin August 2012. Applicants should be committed educators and are expected to promote student learning and help students achieve academic outcomes.

The position requires a Bachelor's degree AND one of the following: five years of professional software development experience, OR a Master's degree, OR an earned PhD.

All degrees must be in Computer Science or Software Engineering or a closely-related field and conferred by the start date. Please see www.nau.edu/hr for full position announcements and application process.

AA/EEO/MWDV employer.

University of Maryland, Baltimore County
Computer Science and Electrical Engineering, Computer Science Professor of Practice

The University of Maryland, Baltimore County invites applications for a non-tenure track position in Computer Science at the rank of Professor of the Practice to begin August 2012. All CS areas will be considered, but we are especially interested in security related areas. Applicants should have a Ph.D. or equivalent stature by virtue of experience. Ideal candidates will have a history of research, publication, teaching, and industry experience. Duties include teaching both graduate and undergraduate courses and helping develop cyber security programs. For best consideration, apply by March 15, 2012. Reviews will begin immediately and applications will be accepted until the position is filled.

For more information and to apply, see http://bit.ly/UMBCPoP. UMBC is an AA/EOE

University of Wisconsin-Platteville
Computer Science & Software Engineering

Position starting August 22, 2012. See http://www.uwplatt.edu/csse for details. Application review begins April 10, 2012. AAEEO employer.

Leslie Pack Kaelbling

MIT's Professor of Computer Science and Engineering Research Director of the Computer Science and Artificial Intelligence Laboratory (CSAIL) Coordinator of SUTD's curriculum development for Information Systems Technology & Design Pillar

INSPIRE THE NEXT GENERATION TO BUILD A BETTER WORLD.

The Singapore University of Technology and Design (SUTD), established in collaboration with the Massachusetts Institute of Technology (MIT), is seeking exceptional faculty members in the area of Information Systems Technology and Design for this new university slated to matriculate its first intake of students in April 2012.

SUTD, the first university in the world with a focus on design accomplished through an integrated multi-disciplinary curriculum, has a mission to advance knowledge and nurture technically grounded leaders and innovators to serve societal needs. SUTD is characterized by a breadth of intellectual perspectives (the "university"), a focus on engineering foundations ("technology") and an emphasis on innovation and creativity ("design"). The University's programmes are based on four pillars leading to separate degree programmes in Architecture and Sustainable Design, Engineering Product Development, Engineering Systems and Design, and Information Systems Technology and Design. Design, as an academic discipline, cuts across the curriculum and will be the framework for novel research and educational programmes.

MIT's multi-faceted collaboration with SUTD includes the development of new courses and curricula, assistance with the early deployment of courses in Singapore, assistance with faculty and student recruiting, mentoring, and career development, and collaborating on a major joint research projects, through a major new international design centre and student exchanges. Many of the newly hired SUTD faculty will spend up to year at MIT in a specially tailored programme for collaboration and professional development.

A BETTER WORLD BY DESIGN.

FACULTY MEMBERS (INFORMATION SYSTEMS TECHNOLOGY AND DESIGN)

The qualifications for the faculty position include: an earned doctorate in Computer Science, Computer Engineering or Information Systems, a strong commitment to teaching at the undergraduate and graduate levels, a demonstrated record of or potential for scholarly research, and excellent communication skills. SUTD invites applicants for tenure-track or tenured appointments in all areas of computer science, computer engineering, and information technology, with particular interest in candidates with expertise in operating systems, databases, networking, security, cryptography, information retrieval, embedded systems, and applied algorithms. Duties include teaching of graduate and undergraduate students, research, supervision of student research, advising undergraduate student projects, and service to SUTD and the community. Faculty will be expected to develop and sustain a strong research programme. Attractive research grant opportunities are also available. Successful candidates can look forward to internationally competitive remuneration, and assistance for relocation to Singapore.

If you want to be part of the founding faculty with a focus on Information Systems Technology and Design, please apply to SUTD at **www.sutd.edu.sg**

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Established in collaboration with MIT

research highlights

DOI:10.1145/2133806.2133827

Technical Perspective
Building Robust Dynamical Simulation Systems

By Dinesh Manocha

THE FIELD OF classical dynamics is regarded as one of the success stories of applied mathematics. A dynamical simulation refers to simulation of a system of objects that move according to laws of physics. The computational power of modern computers has been widely used to predict the behavior of dynamical systems using computer simulations, which are often used as an adjunct to or substitute for a dynamical system when a simple closed form analytic solution is not possible.

Dynamical simulation systems are widely used in different applications, including industrial design or engineering simulation, where such systems are used to simulate the motion of moving parts or crash testing. Computer-based surgical simulators are also used to train medical students or physicians, as they can provide safe, realistic learning environments for repeated practice. Physics-based simulation techniques are increasingly used in computer graphics or animation to generate realistic motion of rigid or deformable objects and fluids. More recently, they are used in video games to generate realistic behaviors of objects.

One major focus in the field has been on building robust and accurate dynamical simulation systems using discrete computational capabilities of current computers. The motion of underlying objects is typically governed by differential equations corresponding to Newton's Second Law of Motion. These equations tend to be nonlinear and are solved using numerical methods. These methods step through a finite time interval and integrate the equations of motion for each object over that interval.

A key challenge in these dynamical systems is to handle the collisions or contacts between various objects or parts of an object. This process includes accurate detection of collisions between the boundaries or the interior meshes of the geometric models of the simulated objects. Furthermore, techniques for robust handling of various contacts, including resting or sliding contacts, are needed. It is well known that even a single missed collision can result in an invalid simulation and noticeable visual artifacts.

The following work by Harmon et al. on *asynchronous contact mechanics* presents a robust method to reliably simulate the contacts in a dynamical system. The paper poses three fundamental requirements for a reliable simulation: that no collisions are missed (a geometric requirement); that physically conserved quantities such as energy and momentum are numerically conserved (a numerical requirement); and that the computation terminates in finite time (an algorithmic requirement). The authors present an algorithm that guarantees fulfillment of all three of these requirements.

A key aspect of this work is robust collision handling. A collision between the primitives can be regarded as a discontinuity in the motion. Prior methods to detect collisions can be classified into two categories: retroactive methods that analyze the preceding time interval and check for contacts during that interval; or predictive methods that use some kind of motion bounds or continuous techniques to estimate the first time of contact between the geometric features in the future. Many conservative methods have been proposed in the literature based on physical properties or equations of motion, which can compute a lower bound on when such a contact may occur in the future, and this bound is used to compute the size of time step.

The authors propose a contact model in which the predictive step is performed in a decoupled or asynchronous manner for every boundary element or feature (for example, a triangle) of the objects. This approach makes it possible to choose varying or large time steps for some of the elements and still guarantees nonpenetration. Moreover, by using asynchronous variational integrators from the mechanics literature, along with this novel contact model, it becomes possible to handle configurations corresponding to sharp boundaries or dispersed points of contacts. The resulting contact handling algorithm can satisfy many challenging constraints corresponding to conservation of momentum and energy. This combination results in a simulation algorithm that is reliable and correct, and does not involve any tweaking of parameters to generate the desired motion. The authors have successfully demonstrated its performance on challenging benchmarks, such as complex deformable simulation of thin objects such as cloth, including tying ribbons into a reef knot.

The authors present a major advancement in terms of building robust dynamical simulation systems for deformable objects. In posing these three desiderata and a preliminary solution, many avenues of exploration are opened: Is asynchrony fundamentally required to satisfy the three requirements? Are there other integrators, perhaps implicit, that might be used instead? The proposed simulation algorithm can robustly perform complex simulations that are regarded as nontrivial for previous methods. However, for simpler scenarios the proposed algorithm is quite slow compared to the state of the art and can take many hours of CPU time to simulate just a few seconds of deformable motion. Could it be accelerated using better collision detection algorithms, including efficient culling methods based on a continuous collision formulation, or parallelization using multicore CPUs or many-core GPUs? In summary, this work by Harmon et al. opens the door to many new directions that together have the exciting potential to achieve the robust, accurate, and fast simulation of deformable models. C

Dinesh Manocha (dm@cs.unc.edu) is currently a Phi Delta Theta/Mason Distinguished Professor of computer science at the University of North Carolina at Chapel Hill. He is a Fellow of ACM, AAAS, and IEEE.

DOI:10.1145/2133806.2133828

Asynchronous Contact Mechanics

By David Harmon,* Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun

Figure 1. A prescribed particle slowly moves through a set of curtains, then impulsively shifts to a very high velocity. The slow and fast phases highlight the method's ability to handle smooth resting and sliding with deep stacking, and arbitrarily fast penetration-free movements in which collisions are treated when (as opposed to well before or after) they occur. The curtains continue to swing for a long time, even as controlled internal dissipation damps high frequencies.

1. MOTIVATION

Physicists have long observed physical phenomena, such as the motion of fluids and the interaction of galaxies, and developed mathematical models to describe these systems. More recently, the advent of computers has allowed us to implement these models as software in a computational environment, launching the field of physical simulation. On a computer we are able to recreate and study physical phenomena within a controlled setting both for descriptive as well as exploratory purposes, leading to advancements in design, engineering, and entertainment.

However, even as computer hardware benefits from Moore's Law, our ability to program, debug, and maintain software advances at a slower pace. This observation shapes our priorities as we develop physical simulation tools for computer graphics. While making choices that yield up-front simplicity and blazing performance is important today, we prefer that these choices do not obstruct our long-term goals of extending functionality and improving physical realism. Laying aside ad hoc models in favor of physical approaches might require a deeper initial investment, but it promises to pay off handsomely in predictability, controllability, and extensibility.

1.1. Safety, correctness, progress

One particularly difficult aspect of simulation is the modeling of complex collisions. A collision occurs when two objects attempt to occupy the same point in space at the same time. Even simple scenarios, like a crumpled shirt, contain an extraordinary number of these contacting points that arise and disappear through the course of a simulation. Robust simulation of complex contact scenarios is critical to applications spanning graphics (training, virtual worlds, entertainment) and engineering (product design, safety analysis, experimental validation). The presence of frequent and plentiful collisions (Figure 1), interactions involving sharp boundaries, resting and sliding contact, and all combinations thereof make it challenging to simulate contact reliably. The inability to handle these difficult situations results in interpenetration, visual artifacts where objects intersect one another—a clearly unphysical configuration. Useful resolution of these scenarios requires consideration of the fundamental issues of geometric *safety*, physical *correctness*, and computational *progress*. These have the respective meanings that (a) for well-posed problems the simulation does not enter an invalid (interpenetrating) state, (b) collision response obeys physical laws of causality and conservation (of mass, momentum, energy, etc.), and (c) the algorithm completes a simulation in finite, preferably short, time. An ideal algorithm offers provable guarantees of safety, correctness, and progress that hold even in the discrete setting of a computer. A safety guarantee eliminates the need to iterate through the animation-design process because of unsightly penetration artifacts; such a guarantee should not fall on a user overburdened with tunable parameters. Respecting discrete conservation laws allows for the development of controllable dissipation without artificial numerical damping. Respect for causality is critical to capturing chain reactions and phenomena such as wave propagation and stacking. If, however, these two guarantees are not accompanied by guaranteed progress, the simulation may never complete, no matter how fast or parallel the hardware.

1.2. Shortcomings of synchrony

Dynamic simulations progress by integrating differential equations, such as Newton's familiar second law, over small steps in time. Most of these integration methods are synchronous, moving the entire configuration forward in lock-step from one instant in time to the next. Such synchrony is fundamentally at odds with safety, correctness, and progress: the first two goals are assured by attending to collisions in order of causality, which, since collisions may propagate at unbounded speed, can require arbitrarily small time steps. The number of possible impact events in a single

The original version of this paper was published in *Proceedings of SIGGRAPH '09*, July 2009, ACM.

*Now at New York University. Work was done while at Columbia University.

"reasonable" time step can be enormous: in their analysis of contact, Cirak and West[5] present a counting argument and conclude that synchronous "contact simulation algorithms cannot attempt to exactly compute the sequence and timing of all impacts," as this would preclude reasonable progress.

The graphics community's prevailing emphasis on *progress* has motivated many efforts to find, *retroactively*, a physically plausible resolution to a given set of collisions that occurred over a preceding time interval.[4, 20] Such methods typically have adjustable parameters that must be carefully chosen to balance safety and progress; other methods discard causality in favor of progress.[19] The principled, faithful simulation of complex collisions for deformable objects, such as cloth and other flexible materials, remains an open, challenging, and important problem.

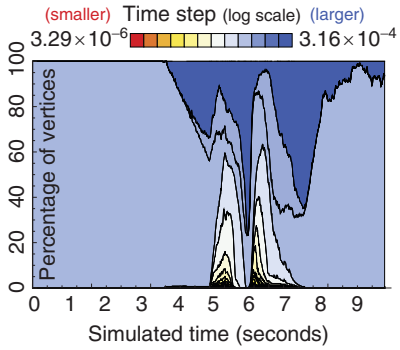
1.3. Asynchrony

We propose to place safety and correctness on an equal footing with progress. To overcome the fundamental opposition between these requirements, we turn to *asynchronous integration*, which integrates each geometric element of a discrete shape (e.g., the stretching resistance of cloth defined across a triangle) at its own pace, *not* in lockstep with the entire object. Asynchrony offers compelling long-term advantages for simulations of deformable objects in complex contact—advantages that remain unexplored, in particular, in terms of safety, correctness, and progress. For scenarios involving sharp boundaries or dispersed points of contact, such as crumpled clothing, asynchrony renders noninterpenetration and momentum conservation tractable. Because elements advance at their own pace, those not entangled in collisions can proceed at large time steps. As shown in Figure 2, the median time step of an asynchronous method can be moderate even when high-impact collisions force some elements to proceed at small time steps.

1.4. Asynchronous integration

As a point of departure we consider *asynchronous variational integrators* (AVIs),[16] which belong to a larger class of integrators that exactly conserve both momentum and symplecticity (loosely related to preservation of areas in phase space); such integrators are highly regarded because of their provable approximate conservation of energy over long spans of simulated time. However, a correct contact model remains unexplored.

Figure 2. Asynchrony in the curtain simulation, depicted by the time-evolving distribution of vertex time step sizes, enables adaptive allocation of computational resources in space time.



1.5. Asynchronous collision detection

To ensure safety, we require an equally principled approach to collision detection. With every object able to collide with any other object, collision detection is fundamentally a quadratic problem. Thus, efficient collision detection algorithms are necessary to prune the non-intersecting pairs. Furthermore, we must reliably find those elements which are proximate rather than actually intersecting, so that we may counteract the impending penetration. This is a heavily studied problem; alas, the many reported successes are specific to the synchronous context, and as a group current methods can be intractably slow if naïvely applied after each local asynchronous step. This motivates our interest in *kinetic data structures* (KDSs)[3]: a KDS algorithm maintains a data structure governed by formal invariants describing some discrete attribute (such as absence of collisions), in response to the continuous movement of geometric elements. Many existing collision detection methods can be reformulated from a KDS perspective. KDSs seem destined for asynchronous applications, because their focus on fast, minimal, "output-sensitive" data-structure updates makes them ideally suited for the small, local changes effected by each AVI step.

These observations motivate our interest in approaching contact mechanics for both graphics and mechanics applications from a new direction. In particular, (a) we formulate a contact model that is *safe* independent of user parameters, such as the stiffness and "bounciness" of collisions. (b) We *correctly* discretize time, using asynchrony to preserve the model's safety and to respect causality, and using a symplectic-momentum integrator to exactly conserve momentum and approximately conserve energy over long runtimes. Finally, (c) we lay out the basic foundations for the union of AVIs with KDSs, making the safe, correct integration of complex contact for highly deformable objects tractable.

2. ASYNCHRONOUS INTEGRATORS

Consider a physical system with a time-varying configuration $\mathbf{q}(t)$ in the space $\mathbf{Q}$ of all configurations; concretely, for a mesh with vertices $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in 3D we represent $\mathbf{Q} = \mathbf{R}^{3n}$ by a vector of all the vertices' Cartesian coordinates. We use a dot to denote differentiation in time, so that $\dot{\mathbf{q}}(t)$ is the velocity of the system. Let M be the mass matrix, so that $\mathbf{p} = M\dot{\mathbf{q}}$ is the momentum. The Störmer–Verlet ("leapfrog") integrator evolves a sequence of positions $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \ldots$ and momenta $\mathbf{p}^{0+\frac{1}{2}}, \mathbf{p}^{1+\frac{1}{2}}, \mathbf{p}^{2+\frac{1}{2}}, \ldots$ via the update rules

$$\mathbf{q}_{k+1} - \mathbf{q}_k = hM^{-1}\mathbf{p}^{k+\frac{1}{2}},$$
$$\mathbf{p}^{k+\frac{1}{2}} - \mathbf{p}^{k-\frac{1}{2}} = hF(\mathbf{q}_k),$$
$$t_k - t_{k-1} = h,$$

where h is the time step and $F(\mathbf{q})$ is the force. The sub/superscripted indices remind us that positions and velocities are staggered in time, with t_k associated to $\mathbf{q}_k$, and (t_k, t_{k+1}) associated to $\mathbf{p}^{k+\frac{1}{2}}$. In effect, leapfrog first updates the position at t_k using the constant momentum associated to the preceding interval (t_{k-1}, t_k), and then impulsively "kicks," obtaining

a new momentum for the following interval (t_k, t_{k+1}), yielding a piecewise linear (p.l.) trajectory over the intervals (t_k, t_{k+1}) (Figure 3). Being a *geometric integrator*,[14] leapfrog tracks conservation laws (e.g., mass, momentum, energy) and adiabatic invariants (e.g., temperature) over long runtimes, and offers more consistency and qualitatively predictable behavior across a range of time step sizes.

AVIs naturally extend leapfrog. Each force receives an independent, regular (fixed-rate) clock, fixed a priori by stability requirements. While impulses of a force are regularly spaced in time, the superposition of forces yields events irregular in time. As with leapfrog, the trajectory is p.l., interrupted by "kicks." When their clocks are nested—as quarter notes are nested in half notes—AVIs reduce to an instance of multistepping methods[14]; our developments apply to this family of methods.

For example, Lew et al.[16] assign an elastic potential to each mesh element. Irregular meshes have spatially varying element shapes and corresponding time step stability restrictions; with AVIs each element advances at its own pace. Since an elemental potential depends only on a local mesh neighborhood, each integration event is *local*, affecting the position and velocity of a small number of *stencil* vertices.

To schedule the interrupts to the p.l. trajectory, AVIs use a priority queue, conceptually populated with all event times until eternity. In practice it suffices to schedule only the next tick for each clock, since that event can schedule the subsequent tock.

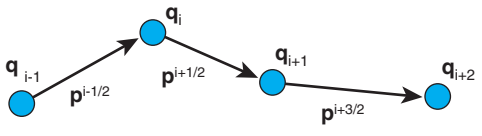
2.1. Ensuring correctness

A more complete analysis leading to the geometric and conservation properties of AVIs invokes ideas from discrete mechanics and variational integration.[16, 17] Here we stress a key outcome: Lew et al. conjecture that AVIs' remarkable properties are due to its *multisymplecticity*; the derivation requires each force to have a regular (constant-rate, ever-ticking) clock. Playing with this clock—accelerating or pausing—is strictly forbidden. Interrupting the p.l. trajectory with other mechanisms (e.g., interleaving a velocity filter) breaks multisymplecticity.

2.2. AVIs and contact

The conservation properties of AVIs rely on preservation of the multisymplectic form[17, 18] and are easily broken by naïvely incorporating existing contact-resolution methods. A principled treatment must consider a multisymplectic formulation of contact mechanics and an asynchronous computation of collision detection and response.

Figure 3. A piecewise linear trajectory where mid-step momenta $\mathbf{p}^{i+\frac{1}{2}}$ carry positions from $\mathbf{q}_i$ to $\mathbf{q}_{i+1}$.



3. DISCRETE PENALTY LAYERS

As a contact model, consider a simple penalty method that penalizes proximity between bodies. We will represent this penalty as a linear half-spring, which only counteracts compression from its designated rest length. Elongation is ignored, allowing separating bodies to move away freely.

For a given surface thickness η, the gap function

$$g_\eta(\mathbf{q}) = \|\mathbf{x}_b - \mathbf{x}_a\| - \eta$$

tracks signed proximity between moving points $\mathbf{x}_a$ and $\mathbf{x}_b$. When $g < 0$, the points are said to be *proximate*. We can express the penalty (half-spring) potential and force in terms of g

$$V_\eta^r(g(\mathbf{q})) = \begin{cases} \frac{1}{2} r g^2 & \text{if } g \le 0, \\ 0 & \text{if } g > 0, \end{cases} \qquad \mathbf{F} = \begin{cases} -rg\nabla g & \text{if } g \le 0, \\ 0 & \text{if } g > 0, \end{cases}$$

respectively, where r is the contact *stiffness*. Choosing a penalty stiffness is the most criticized problem of the penalty method.[1] For any fixed stiffness r, there exists a sufficiently large approach velocity such that the contact potential will be overcome by the momentum, allowing the configuration to tunnel illegally into a penetrating state.

The *barrier method* replaces the above contact potential by a function that grows unbounded as the configuration nears the boundary $g(\mathbf{q}) = 0$, eliminating the possibility of tunneling. However, such a function must also have unbounded second derivative, ruling out stable fixed-step time integration for *any* choice of step size.[14]

To alleviate these concerns, we propose a construction consisting of an infinite family of *nested potentials*

$$V_{\eta(l)}^{r(l)}, \qquad l = 1, 2, \ldots,$$

where $\eta(l)$ is a monotonically decreasing proximity (or "thickness") for the lth potential, and $r(l)$ is a monotonically increasing penalty stiffness. For these nested potentials to be a barrier, the cumulative energy of these potentials must diverge as the distance between two primitives vanishes:

$$\sum_l r(l)\eta(l)^2 \to \infty.$$

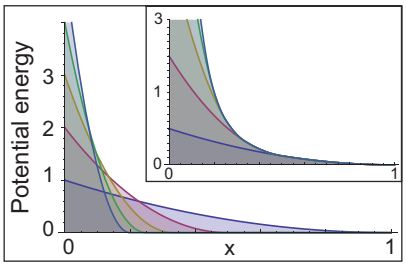
We use $r(l) = r(1)l^3$ and $\eta(l) = \eta(1)l^{-1/4}$, where $r(1)$ and $\eta(1)$ are a simulation-dependent base stiffness and thickness for the outermost layer.

We call the region $\eta(n+1) \le g(\mathbf{q}) \le \eta(n)$, where exactly n of the potentials are nonzero, the nth *discrete penalty layer* (see Figure 4).

The nested potentials' respective maximal stable time steps form a decaying sequence, and therefore this construction *requires* an adaptive or asynchronous time stepping algorithm. Each interaction potential has its own integration clock and has the opportunity to apply an impulsive change in trajectory when its clock ticks. The question is how to time step such an infinite sequence.

As we are about to see, the above construction transforms a seemingly intractable problem in Computational Mechanics—establishing a multisymplectic treatment of

Figure 4. *Discrete penalty layers*. Potential energy of layer *n* plotted against proximity. *Inset:* total potential energy contributed by all layers ″ *n*. The potential energy diverges as x_a approaches x_b, guaranteeing that constraint enforcement is robust.



contact mechanics with *guaranteed* absence of tunneling—into a challenging but addressable problem in Computer Science: efficient bookkeeping on a conceptually infinite set of interaction potentials.

3.1. Central observation

During any time interval, while conceptually the (infinite number of) clocks continues to tick, and the totality of the clock ticks is dense in time, only a *finite*, *sparse* set of clock ticks apply (nonzero) impulses. In particular, the index (l) of the discrete penalty layer (DPL) indicates the number of *active* potentials; the rest, while conceptually present, do not influence the trajectory and can be culled without approximation (Figure 5). What is needed is efficient bookkeeping to track which interaction potentials are active; each status change corresponds to a transition between penalty layers—a *discrete* change in state due to motion along a *continuous* trajectory. This is a problem that KDSs were born to solve.

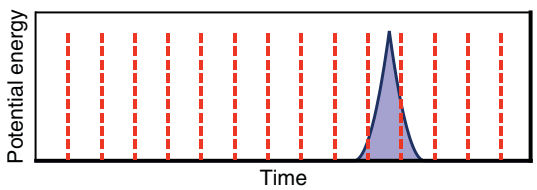
4. KINETIC DATA STRUCTURES

Guibas[12] gives an overview of kinetic data structures. Our culling of inactive forces uses an implementation of kinetic separating slabs for tracking proximity between primitives, closely related to those used by Guibas et al.[13] in the context of rigid polytopes.

4.1. Kinetic separating slabs

Proximity for triangle meshes can be written in terms of distances between vertex-triangle and edge–edge pairs. Thus, our algorithm tracks proximity between these primitives using *certificates*, a concept from the kinetic data structures literature. A certificate is a declaration of some invariant, in this case, that two primitives are separated by at least $\eta(l)$ for a penalty layer l.

Figure 5. Force evaluations (dashed lines) must be evenly spaced in time, yet only those where the potential is nonzero (blue region) must be explicitly evaluated.



To maintain the data structure, we must compute a *certificate failure*, which is the time, given current configuration and velocity, that a certificate ceases to be valid. Computing the time at which two primitives with piece-wise linear motion enter within some fixed proximity requires finding the roots of a degree-six polynomial. This is too expensive for our application, so we use the observation that a certificate failure time needs only to be conservative, not exact.

In this light, we introduce a kinetic separation slab, which we define as a plane in 3-space with constant velocity extruded by $\eta(l)$. Then, for each vertex $\mathbf{q}_i$ in the primitive pair, we can compute the time at which it enters this slab,

$$(\mathbf{q}_i \cdot \hat{\mathbf{n}} - \eta(l))/\mathbf{v},$$

where $\hat{\mathbf{n}}$ is the normal of the separating plane and $\mathbf{v}$ is the assigned constant velocity (we use the relative velocity between the closest point of the two primitives). The earliest of these times is selected as the certificate failure event time.

Because this time is conservative, at the time of a certificate failure event we must check that the primitives are indeed within proximity before creating an appropriate penalty layer event. See Section 5.1 for a walkthrough of the complete algorithm.

4.2. Broad phase

Our implementation begins with the simple separating slab KDS described above. We consider this the "narrow phase" of collision detection, the low-level processing required to track intersections between geometric elements.

While formally correct, the simple KDS used on its own will not scale efficiently to large scenes. Various sophisticated KDSs track proximity, offer better "broad-phase" scaling, and could be easily adapted to the bookkeeping of the DPL index.[6, 9, 11]

One common broad-phase algorithm in traditional (synchronous) simulations are bounding volume hierarchies (BVH).[7] For our implementation, we adopt the kinetic BVH described by Weller and Zachmann,[22] extending their axis-aligned bounding box based method to use k-Discrete Oriented Polytopes, or k-DOPs, which in general provide tighter bounds. For implementation and optimization details, we refer the reader to the full-length publication.

5. ALGORITHM

Kinetic data structures have existed for some time, but this is the first time they have been integrated with AVIs, despite their similar implementations. In this section, we walk the reader through a simple setup to reveal the logic of our algorithm. For simplicity of exposition, we will forego the existence of a k-DOP hierarchy and assume separating slabs are responsible for all proximity detection.

5.1. Walkthrough

Consider a single particle falling toward a fixed floor (Figure 6). Conceptually, the clock for the first penalty layer is always ticking; however, it is active (exerting a nonzero impulse)

only when the particle drops below height $\eta(1)$, say at time t. We must "activate the clock," no later than time t. Activating too late introduces error (misses impulses), while activating too early is correct, albeit overly conservative (some null events are not culled). The separation slab KDS is responsible for this activation.

We initialize a priority queue of events, sorted by time. Initially, this queue contains a gravity event with time step $\mathbf{g}$ (in general, internal forces will be added as well) and one certificate failure event representing the separation slab between the particle and floor. Simulation progresses by repeatedly popping events off the queue and processing them (updating velocities or rescheduling certificates). When a force event modifies velocities, all certificates which depend on that velocity must be rescheduled.

Initially, the particle's velocity is zero and the gravity event is at the front of the queue (Figure 6a). When processed, the particle is given some velocity downward (Figure 6b). The certificate must be rescheduled for the time the particle enters the separation slab, say time t_c.

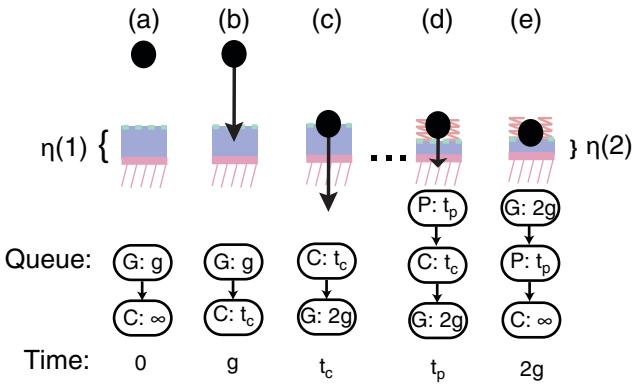
At time t_c the certificate event is popped off the queue (Figure 6c). We see that the particle is within proximity of the floor and add a layer 1 penalty event to the queue. The queue now contains two force events: gravity and a penalty layer 1 event. In general, penalty events are far more frequent than gravity events.

With the creation of a penalty layer 1 event, the certificate event switches to tracking $\eta(2)$ proximity. However, the penalty layer 1 event is still counteracting this motion to reduce further penetration (Figure 6d).

The simulation progresses with penalty force events and gravity events applying impulses in opposite directions. Eventually one of two things will happen: either the particle will enter layer 2 proximity and a second, stiffer penalty force will aid the first in counteracting gravity, or the layer 1 event's force will balance the downward force of gravity. In our illustration, the layer 1 force reaches equilibrium with gravity, a state called *resting contact* (Figure 6e).

For elastic contact where the interacting elements separate, we will need to deactivate penalty forces. The penalty layer force event serves as an opportunity to check whether the particle is transitioning to a shallower penalty layer: if (a) the penalty impulse is null, i.e., separation distance exceeds $\eta(l)$, and (b) the relative velocity is separating rather than approaching, then we deactivate the penalty force, transitioning to the next-shallower layer, and adjusting the certificates accordingly. This lazy approach to deactivation is safe by clause (a) alone; clause (b) aids in efficiency, avoiding rapid toggling of penalty layers.

Figure 6. This didactic example shows a particle falling towards a fixed floor. The queue shows the processing order of events as gravity pulls the particle downward (a)–(b), the certificate creates penalty forces (c), and the penalty force counteracts the penetration (d)–(e).



5.2. Stencils, supports, and scheduling dependencies

Consider the execution of an event at its scheduled time. The set of vertices whose velocities are altered by this event is the *stencil* of the event. The set of vertices whose trajectory was used to schedule this time is the *support* of that event. Building on the notions of stencil and support, an event *depends*, or is *contingent*, on another event if the support of the former overlaps the stencil of the latter; vice versa, an event *supports* another if the stencil of the former overlaps the support of the latter. Table 1 shows the support and stencils for a set of typical events.

KDSs were previously applied only to synchronous simulations, where the velocities of all primitives are updated at the same instant, i.e., the stencil of *the* force-integration event contains the set of *all* vertices. By contrast, in an AVI simulation, force-integration events typically bear small stencils.

Having executed a supporting event, we must reschedule all dependent events before proceeding. This is a problem of executing partially ordered instructions with dependencies, and it is thoroughly studied in the computer systems literature.[15]

Our implementation maintains a directed graph, where edges from events to vertices and vice versa denote stencil and support relations, respectively. When an event executes, the two-neighborhood of outgoing edges yields the set of events to reschedule. The graph abstraction reveals that events with large stencils, such as gravity, should cache a list of contingent events, while events with small stencils should construct the list of contingent events on-the-fly; refer to Figure 7a and b, respectively.

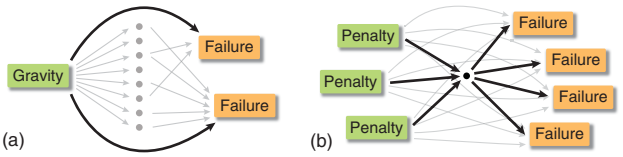
6. RESULTS

We turn our attention to challenging problems involving complex contact geometries, sharp features, and sliding during extremely tight contact.

Table 1. Events and their associated supports and stencils

Event	Supporting vertices	Stencil vertices
Gravity		Entire mesh
Stretching force[16]		Triangle
Bending force[10]		Hinge
Penalty force (Section 3)		Pair of primitives
Separation slab (Section 5.1)	Pair of primitives	
k-DOP overlap (Section 4.2)	Those in *k*-DOP	
Render frame		

Figure 7. Directed graphs depicting events (boxes), vertices (dots), and dependencies (directed edges). Integration events (left green boxes) alter vertex trajectories, forcing rescheduling of dependent events due to certificate failure (right orange boxes). (a) If an integration event has a large stencil, we store event–event dependencies. (b) If a vertex belongs to multiple stencil and support relations, we store event–vertex–event dependencies.



6.1. Knots

We simulate the tying of a ribbon into a reef knot (see Figures 8 and 9). The ribbon is modeled as a loose knot, assigned a material with stiff stretching and weak bending, and the ends are pulled by a prescribed force. The final configuration is faithful to the shape of actual "boyscout manual" knots.

This example demonstrates the strength of asynchrony in allocating resources to loci of tight contact. As the knot tightens, progressively finer time steps are used for the tightest areas of contact. If instead of prescribing reasonable forces we directly prescribe an outward motion of the two ends of the ribbon, the simulations execute to the point where the mesh resolution becomes the limiting reagent, i.e., a tighter knot cannot be tied without splitting triangles; past this point, the computation slows as penalty interactions burrow to deeper layers and the mean time step decays. This highlights both a feature and a potential artistic objection to the method: when presented with an impossible or nearly impossible situation (nonstretchy ribbon with prescribed diametrically opposing displacements at its ends) the method's safety guarantee induces Zeno's Paradox (Figure 9).

6.2. Trash compactor

We place triangle meshes of varying complexity into a virtual trash compactor consisting of a floor and four walls, and then prescribe the inward motion of opposing walls (see Figure 10 and incident image). The method is able to simulate the approach of the walls without ever allowing for seen or unseen penetrations. As with the knots, the overall rate of progress decays as the simulation approaches a limiting configuration.

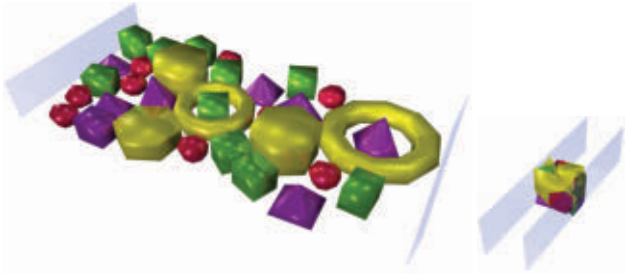
6.3. Bed of nails

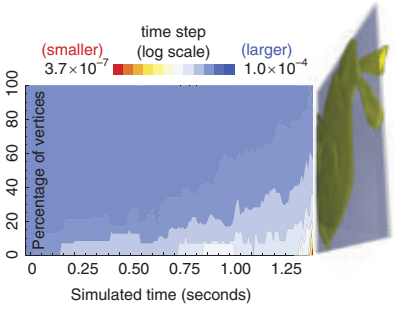
We crafted a problem to test the handling of isolated point contacts and sharp boundaries. Four sliver triangles are assembled into a nail, and many such nails are placed point-up on a flat bed. We drape two stacked fabrics over the bed of nails (see Figure 11), and observe that the simulated trajectory is both realistic and free of penetrations, oscillations, or any other artifacts typically associated to contact discontinuities. Next, we prescribe the motion of one end of the fabric, tugging on the draped configuration to demonstrate sliding over sharp features. We extend the bed of nails into a landing pad for various coarsely meshed projectiles. Variably sized to barely fit or not fit between the nails, and thrown with different initial velocities and angles, the projectiles exhibit a wide array of behaviors, including bouncing, rolling, simple stacking, ricocheting at high frequencies (this requires resolving each collision when it occurs, as resolving collisions over a fixed collision step size can cause aliasing that prevents the ricochet); sliding and getting stuck between nails (the sliding requires a deformable model and friction, since a perfectly rigid object would be constrained to a sudden stop by the distance between nails).

Figure 8. Simulated tying of ribbons into a reef knot.

Figure 9. A closeup of the reef knot.

Figure 10. Virtual trash compactor and assorted virtual trash.





6.4. Timing

We list computation time for the various examples, as executed on a single thread of a 3.06 GHz Intel Xeon with 4GB RAM. The bulk is allocated to the maintenance of the kinetic data structures used for collision detection.

As a more detailed study, consider that the reef knot simulation required 4.8% of total simulation time for integration of elastic forces and gravity, 0.09% for integration of penalty forces, 0.9% for processing and 1.0% for rescheduling

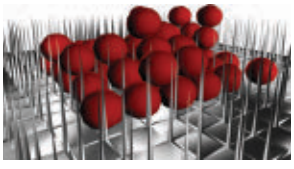


Figure 11. Experiments with a bed of nails highlight the method's ability to deal with sharp boundaries, isolated points of contact, sliver triangles, and localized points of high pressure between two nearly incident surfaces.

of separating plane events, respectively, 5.2% and 23.0% for processing and rescheduling of *k*-DOP events, respectively. The incident figure demonstrates how per frame runtime increases as the stress on the ribbons elevates.

6.5. Parameters

We list parameters for the various examples. Bending and stretching stiffness refers to the Discrete Shells[10] and common edge spring models. COR refers to coefficient of restitution, or the "bounciness" of the collisions.

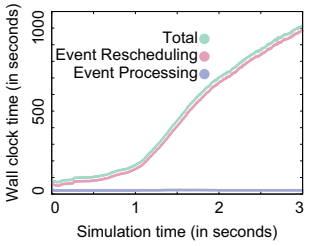
7. DISCUSSION

7.1. Parameters and the triad of safety, correctness, and progress

One of our driving goals is to investigate methods that ensure safety, correctness, and progress regardless of the choice of parameters. The method proposed here does expose some parameters to the user, such as the proximity η. These parameters affect performance, not the triad of guarantees. Our experience in running the problem

Examples	Verticles	Simulation seconds	Event processing (hours)	KDS event rescheduling (hours)	Total (hours)
Reef knot	10,642	2.00	1.5	16.7	18.5
Bowline knot	3,995	5.00	3.0	141.1	144.5
Trash compactor	714	3.08	0.5	53.0	53.6
Two sheets draped	15,982	3.95	4.5	260.8	265.5
Two sheets pulled	15,982	3.83	13.6	310.5	325.6

scenarios, therefore, was qualitatively different than when using other methods, in that we did not need to search for parameters to ensure a successful modeling of contact. On the other hand, our method does not address the spatial discretization of elasticity (stretching and bending models), which can also require user tuning.



Although in theory the nested penalty barrier has infinitely many penalty layers at its disposal, it is impractical to activate penalty layers whose stable time steps are too small, e.g., below the floating point epsilon. Simulations with thicknesses $\eta(1)$ too small, or velocities or masses too high, can thus fail to make progress (but remain safe). This limitation can be worked around by choosing a slow-shrinking layer distribution function, which is why we recommend $\eta(l) = \eta(l)l^{-1/4}$.

Example	Density	COR	r(1)	$\eta(1)$	Stretching stiffness	Stretching damping	Bending stiffness
Reef knot	0.1	0.0	1000.0	0.1	750.0	0.1	0.01
Bowline knot	0.01	0.0	1000.0	0.1	100.0	0.1	0.01
Bunny compactor	0.01	0.01	10000.0	0.05	1000.0	0.0	1000.0
Trash compactor	0.001	0.01	1000.0	0.05	1000.0	15.0	10.0
Two sheets draped	0.001	0.0	1000.0	0.1	1000.0	1.0	0.1
Reef knot untied	0.1	0.0	1000.0	0.1	1000.0	0.1	0.01
Two sheets pulled	0.001	0.0	1000.0	0.1	1000.0	1.0	0.1
Balls on nails	0.016	0.3	10000.0	0.1	50000.0	1.0	100000.0
2D sludge	–	0.0	1000.0	0.1	–	–	–

Multistepping methods such as AVIs are known to have *resonance instabilities*,[8, 14] particularly if the simulation contains adjacent mesh elements of very different size. However, we have not observed any such instabilities or artifacts that we can attribute to such instabilities in our use of the method.

7.2. Broader exploration

In this paper we were concerned with building the most robust contact implementation we could; therefore, we tied the knots as tight as possible, until each triangle was packed as tightly as possible into its neighbors. In the tightest configurations the spatial discretization becomes evident. It would therefore be interesting to introduce spatial adaptation, refining the mesh where curvature is high. Another alternative would be to improve the smoothness at render time, using for example the collision-aware subdivision of Bridson et al.[4]

Dissipation and friction are critical for expressing the widest possible range of scenarios in physical simulation. We have omitted their discussion in this extended abstract, but refer the reader to the original publication for simple models that fit this criteria. Nevertheless, future work might explore efficient algorithms to handle stacking and static friction while still fitting the multisymplectic treatment.

7.3. Immediate and future impact

In considering this method for immediate industrial use, we anticipate two important hurdles. From the standpoint of incorporation into animation systems the first hurdle is the method's insistence on safety even at the cost of artistic freedom. This effectively disallows all pinching,[2, 21] as well as commencing from invalid configurations. We believe that the method can be extended to permit shallow ("skimming") pinching, but handling extremely unphysical boundary conditions within this framework seems at least initially at odds

with the basic premise, and will require further research.

Second, the proposed method is not competitive in performance compared to existing methods, which do not attempt to make strong safety and correctness guarantees; if an artist is willing to search for parameters that provide non-penetrating good-looking results, they may become impatient with the method proposed here.

From the standpoint of long-term, curiosity-driven research, however, this method is appealing not just in its formalism but also in terms of performance, since it lays out a formal asynchronous framework from which one can investigate parallelization, optimization, and even approximation techniques that preserve guarantees of safety, correctness, and progress. To aid such future investigation, source code for our initial C++ implementation, along with data files needed to generate the examples shown in this paper, is available online [a]. C

References

1. Baraff, D. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *SIGGRAPH'89: Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques* (1989), ACM, New York, NY, USA, 223–232.
2. Baraff, D., Witkin, A., Kass, M. Untangling cloth. *ACM Trans. Graph. 22*(3) (2003), 862–870.
3. Basch, J., Guibas, L. J., Hershberger, J. Data structures for mobile data. *J. Algorithms 31* (1999), 1–28.
4. Bridson, R., Fedkiw, R., Anderson, J. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH'02* (2002), 594–603.
5. Cirak, F., West, M. Decomposition-based contact response (DCR) for explicit finite element dynamics. *Int. J. Numer. Methods Eng., 64*(8) (2005), 1078–1110.
6. Erickson, J., Guibas, L. J., Stolfi, J., Zhang, L. Separation-sensitive collision detection for convex objects. In *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms* (1999), 102–111.
7. Ericson, C. *Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3D Technology)*. Morgan Kaufmann, December 2004.
8. Fong, W., Darve, E., Lew, A. Stability of asynchronous variational integrators. *J. Comput. Phys., 227*(18) (2008), 8367–8394.
9. Gao, J., Guibas, L. Hershberger, J., Zhang, L., Zhu, A. Discrete mobile centers. *Discrete Comput. Geom. 30*(1):45–65, 2003.
10. Grinspun, E., Hirani, A., Desbrun, M., Schröder, P. Discrete shells. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (August 2003), 62–67.
11. Guibas, L., Xie, F., Zhang, L. Kinetic collision detection: Algorithms and experiments. In *Proceedings of the International Conference on Robotics and Automation* (2001), 2903–2910.
12. Guibas, L. J. Kinetic data structures—a state of the art report. In *Proceedings of the 3rd Workshop on Algorithmic Foundations of Robotics (WAFR)* (1998), 191–209.
13. Guibas, L. J., Xie, F., Zhang, L. Kinetic collision detection: Algorithms and experiments. In *ICRA* (2001), 2903–2910. http://www.cs.columbia.edu/cg/ACM/
14. Hairer, E., Lubich, C., Wanner, G. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2002.
15. Korneev, V., Kiselev, A. *Modern Microprocessors*. Charles River Media, 2004.
16. Lew, A., Marsden, J. E., Ortiz, M., West, M. Asynchronous variational integrators. *Arch. Rational Mech. Anal. 167* (2003), 85–146.
17. Marsden, J., Patrick, G., Shkoller, S. Multisymplectic geometry, variational integrators, and nonlinear PDEs. *Commun. Math. Phys. 199*(2) (1998), 351–395.
18. Marsden, J., Pekarsky, S., Shkoller, S., West, M. Variational methods, multisymplectic geometry and continuum mechanics. *J. Geom. Phys. 38*(3–4) (June 2001), 253–284.
19. Milenkovic, V. J., Schmidl, H. Optimization-based animation. In *SIGGRAPH'01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), ACM, New York, NY, USA, 37–46.
20. Provot, X. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation '97* (1997), Springer Verlag, Wien, 177–189.
21. Volino, P., Magnenat-Thalmann, N. Resolving surface collisions through intersection contour minimization. In *SIGGRAPH'06: ACM SIGGRAPH 2006 Papers* (2006), ACM, New York, NY, USA,1154–1159.
22. Weller, R. and Zachmann, G. Kinetic separation lists for continuous collision detection of deformable objects. In *Third Workshop in Virtual Reality Interactions and Physical Simulation (Vriphys)* (Madrid, Spain, 6–7 November 2006).

[a] http://www.cs.columbia.edu/cg/ACM

David Harmon Columbia University, New York, NY.

Etienne Vouga Columbia University, New York, NY.

Breannan Smith Columbia University, New York, NY.

Rasmus Tamstorf Walt Disney Animation Studios, Burbank, CA.

Eitan Grinspun Columbia University, New York, NY.

DOI:10.1145/2133806.2133829

Technical Perspective
Who Knows? Searching for Expertise on the Social Web

By Ed H. Chi

IT IS DIFFICULT to remember what people had to do to find the answer to a question before the Web. Imagine it is 1990, before the age of search engines, and of course, Wikipedia. You have no access to a library, and available reference books are not enough. The only option you might have is to call a friend who might know the answer. In fact, this option is so important, it is baked into the game "Who Wants to be a Millionaire?" as one of the three lifeline options to take when you are stumped for an answer. This natural instinct to call someone is also baked into the DNA of Aardvark, the social question and answering (QA) engine described in the following paper by D. Horowitz and S. Kamvar.

When you turn to the phone, one of the first steps you have to figure out is who to call. This is the expertise location or question-routing problem in social QA research. At a high level, this seems like a great computer science problem. You have people as nodes, and their relationships and interactions as edges, and you want to model people's interests and expertise, as well as the frequency and recency of their interactions with each other. You will use these models to route questions. Your mind races with possible algorithms and user modeling approaches to apply. Yes, conceptually you would probably be correct with many of these ideas, but in practice, building Aardvark is much more complex and difficult.

First, you must figure out how to build accurate user models for each user on the whole Web, even with sparse data for many users, including brand new users of your system.

Second, you have to scale this system to millions of users, and be able to do the question routing in milliseconds. You also want to try and get answers that are good and return them as quickly as possible.

Third, potential answerers (read: humans) are finicky: they do not want you to spam them; they do not like being interrupted if they are in the middle of another conversation; and they don't like it when you call them up at weird hours of the day (no matter if you know their time zone or not). In order words, you must deal with the real human context and its associated social interaction.

Finally, you need to socially engineer the growth of this system, so that early users get experiences good enough to rave about your service and recommend it to other users. You want to build trust, and you want a network effect, such that, as each user joins the system, the whole system becomes even more useful to those already there.

Before Aardvark, social QA systems used a wide variety of techniques to route questions, most often using experience/reputation points or monetary rewards as incentives. Many services, such as Yahoo! Answers focused on building communities, and turning the act of answering into a point-based game. Instead, Horowitz, Kamvar and their team at Aardvark pushed the envelope and built a different service that focused on getting answers as quickly as possible; from someone you are socially connected to; and who is likely to be an expert on the topic. How did they do it?

On speed: By connecting Aardvark to a chat service (Google Chat), it exploited and delivered on that expectation of immediacy. In our age of instant knowledge via search engines and Wikipedia, this impedance match is a particularly nice touch that, perhaps ironically, humanizes the experience so it feels like a phone call. Interestingly enough, in a pioneering QA system called AnswerGarden, Mark Ackerman observed that users were often more satisfied when an answer came back quickly, even if the answer was somewhat less than perfect.[1]

On social interactions: Aardvark places emphasis on the social interaction just as much as getting the information—just as if there was a smart assistant who knew your Rolodex and made that phone call for you. Indeed, in Evans and Chi,[2] we showed how social interactions were present and pervasive throughout the information seeking episode—before, during, and after the core search task.

On expertise: Aardvark skillfully exploited the design knowledge gained from years of search engine research to scale the algorithms so that it can route those questions to others who are most likely able to answer it at that very moment. In fact, they followed the original meme of describing the "anatomy" of a search engine in describing their own system—a meme worth repeating in all areas of computer science involved in engineering real-world systems.

Users want one thing—getting their questions answered immediately. Search engines have played that role for many years now. It can be argued that the greatest impact computers have had on the human endeavor is the Web search engine, whose development and refinement seems to be the epitome of computer science. That was before the Web truly became social. In the brave new social Web, search will be different, and reading this paper will give you a sense of the direction social search engines are headed. C

References

1. Ackerman, M.S. Augmenting organizational memory: A field study of answer garden. *ACM Trans. Inf. Syst. 16*, 3 (July 1998), 203-224; http://doi.acm.org/10.1145/290159.290160
2. Evans, B.M. and Chi, E.H. An elaborated model of social search. *Information Processing & Management*; http://dx.doi.org/10.1016/j.ipm.2009.10.012

Ed H. Chi (chi@acm.org) is staff research scientist at Google Research, Mountain View, CA.

DOI:10.1145/2133806.2133830

Searching the Village: Models and Methods for Social Search

By Damon Horowitz and Sepandar D. Kamvar

Abstract

We describe Aardvark, a social search engine. With Aardvark, users ask a question, either by instant message, e-mail, Web input, text message, or voice. Aardvark then routes the question to the person in the user's extended social network most likely to be able to answer that question. As compared to a traditional Web search engine, where the challenge lies in finding the right *document* to satisfy a user's information need, the challenge in a social search engine like Aardvark lies in finding the right *person* to satisfy a user's information need. Further, while trust in a traditional search engine is based on authority, in a social search engine like Aardvark, trust is based on intimacy. We describe how these considerations inform the architecture, algorithms, and user interface of Aardvark, and how they are reflected in the behavior of Aardvark users.

1. INTRODUCTION

1.1. The library and the village

Traditionally, the basic paradigm in information retrieval (IR) has been the library. Indeed, the field of IR has roots in the library sciences, and Google itself came out of the Stanford Digital Library project.[19] While this paradigm has clearly worked well in several contexts, it ignores another age-old model for knowledge acquisition, which we shall call the "village paradigm." In a village, knowledge dissemination is achieved socially—information is passed from person to person, and the retrieval task consists of finding the right person, rather than the right document, to answer your question.

The differences in how people find information in a library versus a village suggest some useful principles for designing a social search engine. In a library, people use keywords to search, the knowledge base is created by a small number of content publishers before the questions are asked, and trust is based on authority. In a village, by contrast, people use natural language to ask questions, answers are generated in real time by anyone in the community, and trust is based on intimacy. These properties have cascading effects—for example, real-time responses from socially proximal responders tend to elicit (and work well for) highly contextualized and subjective queries. For example, the query "Do you have any good babysitter recommendations in Palo Alto for my 6-year-old twins? I'm looking for somebody who won't let them watch TV." is better answered by a friend than a library. These differences in information retrieval paradigm require that a social search engine have very different architecture, algorithms, and user interfaces than a search engine based on the library paradigm.

1.2. Aardvark

In this paper, we describe Aardvark, a social search engine based on the village paradigm. We describe in detail the architecture, ranking algorithms, and user interfaces in Aardvark, and the design considerations that motivated them. We believe this to be useful to the research community for two reasons. First, the argument made in the original Anatomy paper[3] still holds true—as most search engine development is done in industry rather than academia, the research literature describing end-to-end search engine architecture is sparse. Second, the shift in paradigm opens up a number of interesting research questions in information retrieval, for example, around expertise classification, implicit network construction, and conversation design.

Following the architecture description, we present a statistical analysis of usage patterns in Aardvark. We find that, as compared to traditional search, Aardvark queries tend to be long, highly contextualized, and subjective—in short, they tend to be the types of queries that are not well serviced by traditional search engines. We also find that the vast majority of questions get answered promptly and satisfactorily, and that users are surprisingly active, both in asking and answering.

Finally, we present example results from the Aardvark system, and a comparative evaluation experiment. What we find is that Aardvark performs very well on queries that deal with opinion, advice, experience, or recommendations, while traditional corpus-based search engines remain a good choice for queries that are factual or navigational.

2. OVERVIEW

2.1. Main components

The main components of Aardvark are as follows:

1. *Indexer*. To find and label resources that contain information—in this case, users, not documents (Section 3.2).
2. *Query Analyzer*. To understand the user's information need (Section 3.3).
3. *Ranking Function*. To select the best resources to provide the information (Section 3.4).
4. *UI*. To present the information to the user in an accessible and interactive form (Section 3.5).

A previous version of this paper appeared in the *Proceedings of WWW2010* (Raleigh, NC, Apr. 26–30, 2010).

Most corpus-based search engines have similar key components with similar aims,[3] but the means of achieving those aims are quite different.

Before discussing the anatomy of Aardvark in depth, it is useful to describe what happens behind the scenes when a new user joins Aardvark and when a user asks a question.

2.2. The initiation of a user

When a new user first joins Aardvark, the Aardvark system (Figure 1) performs a number of indexing steps in order to be able to direct the appropriate questions to her for answering.

Because questions in Aardvark are routed to the user's extended network, the first step involves indexing friendship and affiliation information. The data structure responsible for this is the *Social Graph*. Aardvark's aim is not to build a social network, but rather to allow people to make use of their existing social networks. As such, in the sign-up process, new users have the option of connecting to a social network such as Facebook or LinkedIn, importing their contact lists from a Webmail program, or manually inviting friends to join. Additionally, anybody whom the user invites to join Aardvark is appended to their Social Graph—and such invitations are a major source of new users. Finally, Aardvark users are connected through common "groups" which reflect real-world affiliations they have, such as the schools they have attended and the companies they have worked at; these groups can be imported automatically from social networks or manually created by users. Aardvark indexes this information and stores it in the Social Graph, which is a fixed width ISAM index sorted by userId.

Simultaneously, Aardvark indexes the topics about which the new user has some level of knowledge or experience. This topical expertise can be garnered from several sources: a user can indicate topics in which he believes himself to have expertise; a user's friends can indicate which topics they trust the user's opinions about; a user can specify an existing structured profile page from which the *Topic Parser* parses additional topics; a user can specify an account on which they regularly post status updates (e.g., Twitter or Facebook), from which the *Topic Extractor* extracts topics (from unstructured text) in an ongoing basis (see Section 3.2 for more discussion); and finally, Aardvark observes the user's behavior on Aardvark, in answering (or electing not to answer) questions about particular topics.

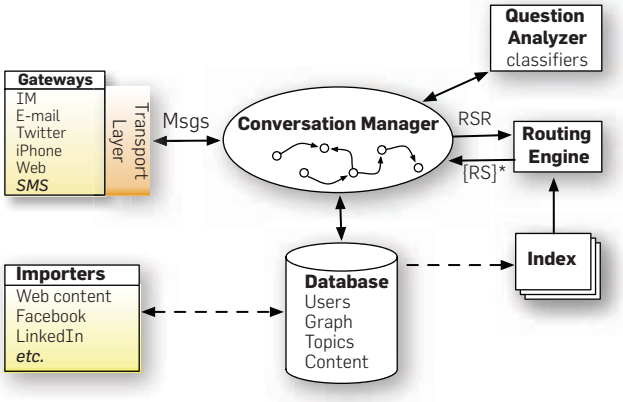
The set of topics associated with a user is recorded in the *Forward Index,* which stores each userId, a scored list of topics, and a series of further scores about a user's behavior (e.g., responsiveness or answer quality). From the Forward Index, Aardvark constructs an *Inverted Index*. The Inverted Index stores each topicId and a scored list of userIds that have expertise in that topic. In addition to topics, the Inverted Index stores scored lists of userIds for features like answer quality and response time.

Once the Inverted Index and the Social Graph for a user are created, the user is now active on the system and ready to ask her first question.

2.3. The life of a query

A user begins by asking a question, most commonly through instant message or text message. The question gets sent from the input device to the *Transport Layer,* where it is normalized to a *Message* data structure and sent to the *Conversation Manager*. Once the Conversation Manager determines that the message is a question, it sends the question to the *Question Analyzer* to determine the appropriate topics for the question. The Conversation Manager informs the asker which primary topic was determined for the question and gives the asker the opportunity to edit it. It simultaneously issues a *Routing Suggestion Request* to the *Routing Engine*. The Routing Engine plays a role analogous to the ranking function in a corpus-based search engine. It accesses the *Inverted Index* and *Social Graph* for a list of candidate answerers, and ranks them to reflect how well it believes they can answer the question and how good of a match they are for the asker. The Routing Engine returns a ranked list of *Routing Suggestions* to the Conversation Manager, which then contacts the potential answerers—one by one or a few at a time, depending upon a *Routing Policy*—and asks them if they would like to answer the question, until a satisfactory answer is found. The Conversation Manager then forwards this answer along to the asker and allows the asker and answerer to exchange follow-up messages.

Figure 1. Schematic of the architecture of Aardvark.



3. ANATOMY

3.1. The model

The core of Aardvark is a statistical model for routing questions to potential answerers. We use a network variant of what has been called an *aspect model*,[11] which has two primary features. First, it associates an unobserved class variable $t \in T$ with each observation (i.e., the successful answer of question q by user u_i). In other words, the probability $p(u_i|q)$ that user i will successfully answer question q depends on whether q is about the topics t in which u_i has expertise[a]:

$$p(u_i \mid q) = \sum_{t \in T} p(u_i \mid t) p(t \mid q) \quad (1)$$

The second main feature of the model is that it defines a

[a] Equation 1 is a simplification of what Aardvark actually uses to match queries to answerers, but we present it this way for clarity and conciseness.

query-independent probability of success for each potential asker/answerer pair (u_i, u_j), based upon their degree of social connectedness and profile similarity. In other words, we define a probability $p(u_i|u_j)$ that user u_i will deliver a satisfying answer to user u_j, regardless of the question.

We then define the scoring function $s(u_i, u_j, q)$ as the composition of the two probabilities.

$$\begin{aligned} s(u_i,u_j,q) &= p(u_i|u_j)\cdot p(u_i|q) \\ &= p(u_i|u_j)\sum_{t\in T} p(u_i|t)p(t|q) \end{aligned} \quad (2)$$

This scoring function is derived using a Bayesian approach described in Kamvar and Horowitz.[15] The ranking problem thus becomes: given a question q from user u_j, return a ranked list of users $u_i \in U$ that maximizes $s(u_i,u_j, q)$.

Note that the scoring function is composed of a query-dependent *relevance score* $p(u_i|q)$ and a query-independent *quality score* $p(u_i|u_j)$. This bears similarity to the ranking functions of traditional corpus-based search engines such as Google.[3] The difference is that unlike quality scores like PageRank,[19] Aardvark's quality score aims to measure intimacy rather than authority. And unlike the relevance scores in corpus-based search engines, Aardvark's relevance score aims to measure a user's *potential* to answer a query, rather than a document's existing capability to answer a query.

Computationally, this scoring function has a number of advantages. It allows real-time routing because it pushes much of the computation offline. The only component probability that needs to be computed at query time is $p(t|q)$. Computing $p(t|q)$ is equivalent to assigning topics to a question—in Aardvark we do this by running a probabilistic classifier on the question at query time (see Section 3.3). The distribution $p(u_i|t)$ assigns users to topics, and the distribution $p(u_i|u_j)$ defines the Aardvark Social Graph. Both of these are computed by the Indexer at signup time and then updated continuously in the background as users answer questions and get feedback (see Section 3.2). The component multiplications and sorting are also done at query time, but these are easily parallelizable, as the index is sharded by user.

3.2. Indexing people

The central technical challenge in Aardvark is selecting the right user to answer a given question from another user. In order to do this, the two main things Aardvark needs to learn about each user u_i are (1) the topics t he might be able to answer questions about $p_{smoothed}(t|u_i)$; (2) the users u_j to whom he is connected $p(u_i|u_j)$.

Topics. Aardvark computes the distribution $p(t|u_i)$ of topics known by user u_i from many several information sources, for example:

- Users are prompted to provide at least three topics which they believe they have expertise about.
- Friends of a user (and the person who invited a user) are encouraged to provide a few topics that they trust the user's opinion about.
- Aardvark parses out topics from users' existing online profiles (e.g., Facebook profile pages, if provided).

The motivation for using these latter sources of profile topic information is a simple one: if you want to be able to predict what kind of content users will generate (i.e., $p(t|u_i)$), first examine the content they have generated in the past. In this spirit, Aardvark uses Web content not as a source of existing answers about a topic, but rather as an indicator of the topics about which a user is likely able to give new answers on demand.

In essence, this involves modeling a user as a content-generator, with probabilities indicating the likelihood she will likely respond to questions about given topics. Each topic in a user profile has an associated score, depending upon the confidence appropriate to the source of the topic. In addition, Aardvark learns over time which topics *not* to send a user questions about by keeping track of cases when the user: (1) explicitly "mutes" a topic; (2) declines to answer questions about a topic when given the opportunity; (3) receives negative feedback on his answer about the topic from another user.

Periodically, Aardvark will run a topic-strengthening algorithm, the essential idea of which is, if a user has expertise in a topic and most of his friends also have some expertise in that topic, we have more confidence in that user's level of expertise than if he were alone in his group with knowledge in that area. Mathematically, for some user u_i, his group of friends U, and some topic t, if $p(t|u_i) \neq 0$, then $s(t|u_i) = p(t|u_i) + \gamma \sum_{u\in U} p(t|u)$ where γ is a small constant. The s values are then renormalized to form probabilities.

Aardvark then runs two smoothing algorithms, the purpose of which are to record the possibility that the user may be able to answer questions about additional topics not explicitly recorded in her profile. The first uses basic collaborative filtering techniques on topics (i.e., based on users with similar topics), the second uses semantic similarity.[b]

Once all of these bootstrap, extraction, and smoothing methods are applied, we have a list of topics and scores for a given user. Normalizing these topic scores so that $\sum_{t\in T} p(t|u_i) = 1$, we have a probability distribution for topics known by user u_i. Using Bayes' Law, we compute for each topic and user:

$$p(u_i|t) = \frac{p(t|u_i)p(u_i)}{p(t)}, \quad (3)$$

using a uniform distribution for $p(u_i)$ and observed topic frequencies for $p(t)$. Aardvark collects these probabilities $p(u_i|t)$ indexed by topic into the Inverted Index, which allows for easy lookup when a question comes in.

Connections. Aardvark computes the connectedness between users $p(u_i|u_j)$ in a number of ways. While social proximity is very important here, we also take into account similarities in demographics and behavior. Many factors are

[b] In both the Person Indexing and the Question Analysis components, "semantic similarity" is computed by using an approximation of distributional similarity computed over Wikipedia and other corpora; this serves as a proxy measure of the topics' semantic relatedness.

considered, including Social connection (common friends and affiliations), Demographic similarity, Profile similarity (e.g., common favorite movies), Vocabulary match (e.g., IM shortcuts), and Verbosity match (the average length of messages). Connection strengths between people are computed using a weighted cosine similarity over this feature set, normalized so that $\sum_{u_i \in U} p\left(u_i \middle| u_j\right) = 1$, and stored in the Social Graph for quick access at query time.

Both the distributions $p(u_i|u_j)$ in the Social Graph and $p(t|u_i)$ in the Inverted Index are continuously updated as users interact with one another on Aardvark.

3.3. Analyzing questions

The purpose of the Question Analyzer is to determine a scored list of topics $p(t|q)$ for each question q representing the semantic subject matter of the question. This is the only probability distribution in Equation 2 that is computed at query time.

It is important to note that in a social search system, the requirement for a Question Analyzer is only to be able to understand the query sufficiently for routing it to a likely answerer. This is a considerably simpler task than the challenge facing an ideal Web search engine, which must attempt to determine exactly what piece of information the user is seeking (i.e., given that the searcher must translate her information need into search keywords) and evaluate whether a given Web page contains that piece of information. By contrast, in a social search system, it is the human answerer who has the responsibility for determining the relevance of an answer to a question—and this is a function which human intelligence is extremely well suited to perform! The asker can express his information need in natural language, and the human answerer can simply use her natural understanding of the language of the question, of its tone of voice, sense of urgency, sophistication or formality, and so forth to determine what information is suitable to include in a response. Thus, the role of the Question Analyzer in a social search system is simply to learn enough about the question that it may send to appropriately interested and knowledgeable human answerers.

As a first step, several classifiers are run in order to determine whether the input is actually a question, if it is an inappropriate question, if it is a trivial question, and if it is a location-sensitive question.

Next, the list of topics relevant to a question is produced by merging the output of several distinct TopicMapper algorithms, for example:

- A *KeywordMatchTopicMapper* passes any terms in the question which are string matches with user profile topics through a classifier which is trained to determine whether a given match is likely to be semantically significant or misleading.[c]
- A *Taxonomy TopicMapper* classifies the question text into a taxonomy of roughly 3000 popular question topics using an SVM trained on an annotated corpus of several millions questions.
- A *SalientTermTopicMapper* extracts salient phrases from the question—using a noun-phrase chunker and a tf-idf-based measure of importance—and finds semantically similar user topics.
- A *UserTagTopicMapper* takes any user "tags" provided by the asker (or by any would-be answerers) and maps these to semantically similar user topics.[d]

At present, the output distributions of these classifiers are combined by weighted linear combination. It would be interesting future work to explore other means of combining heterogeneous classifiers, such as the maximum entropy model in Klein et al.[17]

The Aardvark TopicMapper algorithms are continuously evaluated by manual scoring on random samples of 1000 questions. The topics used for selecting candidate answerers, as well as a much larger list of possibly relevant topics, are assigned scores by two human judges, with a third judge adjudicating disagreements. For the current algorithms on the current sample of questions, this process yields overall scores of 89% precision and 84% recall of relevant topics. In other words, 9 out of 10 times, Aardvark will be able to route a question to someone with relevant topics in her profile; and Aardvark will identify five out of every six possibly relevant answerers for each question based upon their topics.

3.4. The Aardvark ranking algorithm

Ranking in Aardvark is done by the Routing Engine, which determines an ordered list of users (or "candidate answerers") who should be contacted to answer a question, given the asker of the question and the information about the question derived by the Question Analyzer. The core Ranking Function is described by Equation 2; essentially, the Routing Engine can be seen as computing Equation 2 for all candidate answerers, sorting, and doing some postprocessing.

The main factors that determine this ranking of users are Topic Expertise $p(u_i|q)$, Connectedness $p(u_i|u_j)$, and Availability.

Topic Expertise. First, the Routing Engine finds the subset of users who are semantic matches to the question: those users whose profile topics indicate expertise relevant to the topics which the question is about. Users whose profile topics are closer matches to the question's topics are given higher rank. For questions which are location sensitive (as defined earlier), only users with matching locations in their profiles are considered.

Connectedness. Second, the Routing Engine scores each user according to the degree to which she herself—as a person, independently of her topical expertise—is a good

[c] For example, if the string "camel wrestling" occurs in a question, it is likely to be semantically relevant to a user who has "camel wrestling" as a profile topic; whereas the string "running" is too ambiguous to use without further validation, as it might route a question about "running a business" to a user who knows about fitness.

[d] A general principle in the design of Aardvark is to use human intelligence wherever possible to improve the quality of the system. For the present task of Question Analysis, this involves giving both askers and answerers prompts and simple commands for telling Aardvark directly what the subject matter of a question is.

"match" for the asker for this information query. The goal of this scoring is to optimize the degree to which the asker and the answerer feel kinship and trust, arising from their sense of connection and similarity, and meet each other's expectations for conversational behavior in the interaction.

Availability. Third, the Routing Engine prioritizes candidate answerers in such a way so as to optimize the chances that the present question will be answered, while also preserving the available set of answerers (i.e., the quantity of "answering resource" in the system) as much as possible by spreading out the answering load across the user base. This involves factors such as prioritizing users who are currently online (e.g., via IM presence data, iPhone usage, etc.), who are historically active at the present time of day, and who have not been contacted recently with a request to answer a question.

Given this ordered list of candidate answerers, the Routing Engine then filters out users who should not be contacted, according to Aardvark's guidelines for preserving a high-quality user experience. These filters operate largely as a set of rules: do not contact users who prefer to not be contacted at the present time of day; do not contact users who have recently been contacted as many times as their contact frequency settings permit; etc.

Since this is all done at query time, and the set of candidate answerers can potentially be very large, it is useful to note that this process is parallelizable. Each shard in the Index computes its own ranking for the users in that shard and sends the top users to the Routing Engine. This is scalable as the user base grows, since as more users are added, more shards can be added.

The list of candidate answerers who survive this filtering process are returned to the Conversation Manager. The Conversation Manager then proceeds with opening channels to each of them, serially, inquiring whether they would like to answer the present question and iterating until an answer is provided and returned to the asker.

3.5. User interface

Since social search is modeled after the real-world process of asking questions to friends, the various user interfaces for Aardvark are built on top of the existing communication channels that people use to ask questions to their friends: IM, e-mail, SMS, iPhone, Twitter, and Web-based messaging. Experiments were also done using actual voice input from phones, but this is not live in the current Aardvark production system.

In its simplest form, the user interface for asking a question on Aardvark is any kind of text input mechanism, along with a mechanism for displaying textual messages returned from Aardvark. (This kind of very lightweight interface is important for making the search service available anywhere, especially now that mobile device usage is ubiquitous across most of the globe.)

However, Aardvark is most powerful when used through a chat-like interface that enables ongoing conversational interaction. A private one-to-one conversation creates an intimacy which encourages both honesty and freedom within the constraints of real-world social norms. (By contrast, answering forums where there is a public audience can both inhibit potential answerers[18] or motivate public performance rather than authentic answering behavior.[23]) Further, in a real-time conversation, it is possible for an answerer to request clarifying information from the asker about her question or for the asker to follow-up with further reactions or inquiries to the answerer.

There are two main interaction flows available in Aardvark for answering a question. The primary flow involves Aardvark sending a user a message (over IM, e-mail, etc.), asking if she would like to answer a question: for example, "You there? A friend from the Stanford group has a question about *search engine optimization* that I think you might be able to answer." If the user responds affirmatively, Aardvark relays the question as well as the name of the questioner. The user may then simply type an answer to the question, type in a friend's name or e-mail address to refer it to someone else who might answer, or simply "pass" on this request.[e]

A key benefit of this interaction model is that the available set of potential answerers is not just whatever users happen to be visiting a bulletin board at the time a question is posted, but rather the entire set of users that Aardvark has contact information for. Because this kind of "reaching out" to users has the potential to become an unwelcome interruption if it happens too frequently, Aardvark sends such requests for answers usually less than once a day to a given user (and users can easily change their contact settings, specifying preferred frequency and time of day for such requests). Further, users can ask Aardvark "why" they were selected for a particular question and be given the option to easily change their profile if they do not want such questions

Figure 2. Example of Aardvark interacting with an answerer.

[e] There is no shame in "passing" on a question, as nobody else knows that the question was sent to you. Similarly, there is no social cost to the user in asking a question, as you are not directly imposing on a friend or requesting a favor; rather, Aardvark plays the role of the intermediary who bears this social cost.

in the future. This is very much like the real-world model of social information sharing: the person asking a question, or the intermediary in Aardvark's role, is careful not to impose too much upon a possible answerer (Figure 2). The ability to reach out to an extended network beyond a user's immediate friendships, without imposing too frequently on that network, provides a key differentiating experience from simply posting questions to one's Twitter or Facebook status message.

In order to play the role of intermediary in an ongoing conversation, Aardvark must have some basic conversational intelligence in order to understand where to direct messages from a user: is a given message a new question, a continuation of a previous question, an answer to an earlier question, or a command to Aardvark? The details of how the Conversation Manager manages these complications and disambiguates user messages are not essential, so they are not elaborated here; but the basic approach is to use a state machine to model the discourse context.

In all of the interfaces, wrappers around the messages from another user include information about the user that can facilitate trust: the user's Real Name nametag, with their name, age, gender, and location; the social connection between you and the user (e.g., "Your friend on Facebook,", "A friend of your friend Marshall Smith," "You are both in the Stanford group," etc.); a selection of topics the user has expertise in; and summary statistics of the user's activity on Aardvark (e.g., number of questions recently asked or answered).

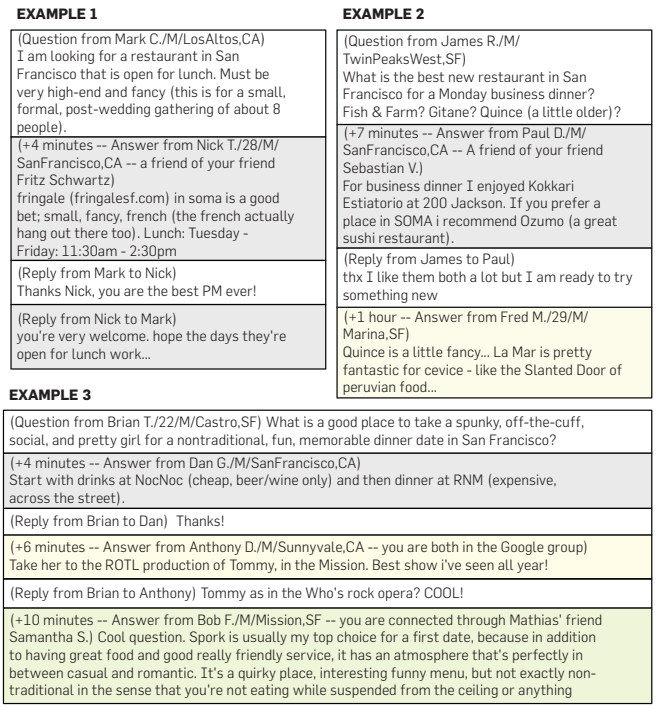
Finally, it is important throughout all of the above interactions that Aardvark maintains a tone of voice which is friendly, polite, and appreciative. A social search engine depends upon the goodwill and interest of its users, so it is important to demonstrate the kind of (linguistic) behavior that can encourage these sentiments, in order to set a good example for users to adopt. Indeed, in user interviews, users often express their desire to have examples of how to speak or behave socially when using Aardvark; as it is a novel paradigm, users do not immediately realize that they can behave in the same ways they would in a comparable real-world situation of asking for help and offering assistance. All of the language that Aardvark uses is intended both to be a communication mechanism between Aardvark and the user and an example of how to interact with Aardvark.

Overall, a large body of research[2, 6, 7, 22, f] shows that when you provide a one-to-one communication channel, use real identities rather than pseudonyms, facilitate interactions between existing real-world relationships, and consistently provide examples of how to behave, users in an online community will behave in a manner that is far more authentic and helpful than pseudonymous multicasting environments with no moderators. The design of the Aardvark's UI has been carefully crafted around these principles.

4. EXAMPLES

In this section, we take a qualitative look at user behavior on Aardvark. Figure 3 examines three questions sent to Aardvark during this period, all three of which were categorized by the Question Analyzer under the primary topic "restaurants in San Francisco."[6]

Figure 3. Three complete Aardvark interactions.



In Example 1, Aardvark opened three channels with candidate answerers, which yielded one answer. An interesting (and not uncommon) aspect of this example is that the asker and the answerer in fact were already acquaintances, though only listed as "friends of friends" in their online social graphs; and they had a quick back-and-forth chat through Aardvark.

In Example 3, Aardvark opened 10 channels with candidate answerers, yielding three answers. The first answer came from someone with only a distant social connection to the asker; the second answer came from a coworker; and the third answer came from a friend of friend of friend. The third answer, which is the most detailed, came from a user who has topics in his profile related to both "restaurants" and "dating."

One of the most interesting features of Aardvark is that it allows askers to get answers that are hypercustomized to their information need. Very different restaurant recommendations are appropriate for a date with a spunky and spontaneous young woman, a post-wedding small formal family gathering, and a Monday evening business meeting—and human answerers are able to recognize these constraints. It is also interesting to note that in most of these examples (as in the majority of Aardvark questions), the asker took the time to thank the answerer for helping out.

5. ANALYSIS

The following statistics give a picture of the current usage and performance of Aardvark.

Aardvark was first made available semi-publicly in a beta release in March of 2009. From March 1, 2009 to October

[f] Names and affiliations have been changed to protect privacy.

20, 2009, the number of users grew to 90,361, having asked a total of 225,047 questions and given 386,702 answers. All of the statistics below are taken from the last month of this period (9/20/2009–10/20/2009).

Aardvark is actively used. As of October, 2009, 90,361 users have created accounts on Aardvark, growing organically from 2272 users since March 2009. In this period, 50,526 users (55.9% of the user base) generated content on Aardvark (i.e., asked or answered a question), while 66,658 users (73.8% of the user base) passively engaged (i.e., either referred or tagged other peoples' questions). The average query volume was 3167.2 questions per day in this period, and the median active user issued 3.1 queries per month.

Mobile users are particularly active. Mobile users had an average of 3.6322 sessions per month, which is surprising on two levels. First, mobile users of Aardvark are more active than desktop users. (As a point of comparison, on Google, desktop users are almost 3 times as active as mobile users.[13]) Second, mobile users of Aardvark are almost as active in absolute terms as mobile users of Google (who have on average 5.68 mobile sessions per month[13]). This is quite surprising for a service that has only been available for 6 months.

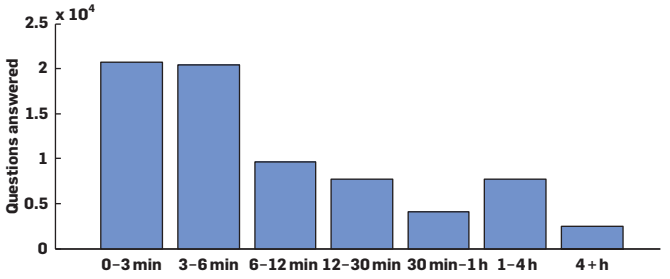
Questions are highly contextualized. As compared to Web search, where the average query length is between 2.2 and 2.9 words,[13, 20] with Aardvark, the average query length is 18.6 words (median = 13). While some of this increased length is due to the increased usage of function words, 45.3% of these words are content words that give context to the query. In other words, as compared to traditional Web search, Aardvark questions have 3–4 times as much context.

The addition of context results in a greater diversity of queries. While in Web search, between 57 and 63% of queries are unique,[20, 21] in Aardvark 98.1% of questions are unique (and 98.2% of answers are unique).

Questions often have a subjective element. A manual tally of 1000 random questions between March and October of 2009 shows that 64.7% of queries have a subjective element to them (for example, "Do you know of any great delis in Baltimore, MD?" or "What are the things/crafts/toys your children have made that made them really proud of themselves?"). In particular, advice or recommendations queries regarding travel, restaurants, and products are very popular. A large number of queries are locally oriented. About 10% of questions related to local services, and 13% dealt with restaurants and bars.

Questions get answered quickly. Of the questions submitted to Aardvark, 87.7% received at least one answer, and 57.2% received their first answer in less than 10 min. On average, a question received 2.08 answers, and the median answering time was 6 min and 37 s (Figure 4). By contrast, on public question and answer forums such as Yahoo! Answers,[10] most questions are not answered within the first 10 min, and for questions asked on Facebook, only 15.7% of questions are answered within 15 min.[18] (Of course, corpus-based search engines such as Google return results in milliseconds, but many of the types of questions that are asked from Aardvark require extensive browsing and query refinement when asked on corpus-based search engines.)

Figure 4. Distribution of questions and answering times.



Answers are high quality. Aardvark answers are both comprehensive and concise. The median answer length was 22.2 words; 22.9% of answers were over 50 words (the length of a paragraph); and 9.1% of answers included hypertext links in them. In the inline feedback which askers provided on the answers they received, 70.4% rated the answers as "good," 14.1% rated the answers as "OK," and 15.5% rated the answers as "bad."

There are a broad range of answerers. Aardvark has contacted 78,343 users (86.7% of users) with a request to answer a question, and of those, 70% have asked to look at the question, and 38.0% have been able to answer. Additionally, 15,301 users (16.9% of all users) have contacted Aardvark of their own initiative to try answering a question. Altogether, 45,160 users (50.0% of the total user base) have answered a question; this is 75% of all users who interacted with Aardvark at all in the period (66,658 users). As a comparison, only 27% of Yahoo! Answers users have ever answered a question.[10] While a smaller portion of the Aardvark user base is much more active in answering questions—approximately 20% of the user base is responsible for 85% of the total number of answers delivered to date—the distributions of answerers across the user base is far broader than on a typical user-generated content site.[10]

Social proximity matters. Of questions that were routed to somebody in the asker's social network (most commonly a friend of a friend), 76% of the inline feedback rated the answer as "good," whereas for those answers that came from outside the asker's social network, 68% of them were rated as " 'good."

6. EVALUATION

To evaluate social search compared to Web search, we ran a side-by-side experiment with Google on a random sample of Aardvark queries. We inserted a "Tip" into a random sample of active questions on Aardvark that read "Do you want to help Aardvark run an experiment?" with a link to an instruction page that asked the user to reformulate their question as a keyword query and search on Google. We asked the users to time how long it took to find a satisfactory answer on both Aardvark and Google and to rate the answers from both on a 1–5 scale. If it took longer than 10 min to find a satisfactory answer, we instructed the user to give up. Of the 200 responders in the experiment set, we found that 71.5% of the queries were answered successfully on Aardvark, with a mean rating of 3.93 ($\sigma = 1.23$), while 70.5% of the queries were answered

successfully on Google, with a mean rating of 3.07 (σ = 1.46). The median time to satisfactory response for Aardvark was 5 min (of passive waiting), while the median time to satisfactory response for Google was 2 min (of active searching).

Of course, since this evaluation involves reviewing questions which users actually sent to Aardvark, we should expect that Aardvark would perform well—after all, users chose these particular questions to send to Aardvark because of their belief that it would be helpful in these cases.[g]

Thus, we cannot conclude from this evaluation that social search will be equally successful for all kinds of questions. Further, we would assume that if the experiment were reversed, and we used as our test set a random sample from Google's query stream, the results of the experiment would be quite different. Indeed, for questions such as "What is the train schedule from Middletown, NJ?," traditional Web search is a preferable option.

However, the questions asked of Aardvark do represent a large and important class of information need: they are typical of the kind of subjective questions for which it is difficult for traditional Web search engines to provide satisfying results. The questions include background details and elements of context that specify exactly what the asker is looking for, and it is not obvious how to translate these information needs into keyword searches. Further, there are not always existing Web pages that contain exactly the content that is being sought; and in any event, it is difficult for the asker to assess whether any content that is returned is trustworthy or right for them. In these cases, askers are looking for personal opinions, recommendations, or advice from someone they feel a connection with and trust. The desire to have a fellow human being understand what you are looking for and respond in a personalized manner in real time is one of the main reasons why social search is an appealing mechanism for information retrieval.

7. RELATED WORK

There is an extensive literature on query-routing algorithms, particularly in P2P Networks. In Condie et al.,[4] queries are routed via a relationship-based overlay network. Peers route queries preferentially to peers with whom they have had positive past interactions. In Kamvar et al.,[16] answerers of a multicast query are ranked via a decentralized authority score. These authority scores are computed by aggregating local trust scores based on previous interactions. Davitz et al.[5] describe a query-routing system in which queries are routed through a supernode that routes to answerers based on a weighted linear combination of authority, responsiveness, and expertise. In this case, expertise is computed by an aspect model, authority is computed in a similar manner to Kamvar et al.,[16] and responsiveness is a function of response rates and response accuracy. The authors provide a general model for open-source content production and choose FAQ generation as a specific application. In Faye et al.,[9] supernodes maintain expertise tables for routing queries to appropriate neighboring peers. These expertise tables along with a matching technique form a semantic overlay network. Banerjee and Basu[1] introduce a routing model for decentralized search that has PageRank and certain Markov Decision Processes as special cases. Aspect models have been used widely in information retrieval, for example, to match queries to documents based on topic similarity in Hofmann[11] and queries to users based on topic expertise in Davitz et al.[5] Many implementations of personalized search use a scoring function similar to Equation 2, in which a personalized authority score is composed with an unpersonalized text IR score.[14] Evans and Chi[8] describe a social model of user activities before, during, and after search, and Morris et al.[18] present an analysis of questions asked on social networks that mirrors some of our findings on Aardvark. A longer version of this paper originally appeared in WWW2010. [12]

[g] In many cases, users in the experiment noted that they sent their question to Aardvark specifically because a previous Google search was difficult to formulate or did not give a satisfactory result. For example: "Which golf courses in the San Francisco Bay Area have the best drainage/are the most playable during the winter (especially with all of the rain we've been getting)?"

Acknowledgments

We would like to thank Max Ventilla, Rob Spiro, and the entire Aardvark team for their contributions to the work reported here.

References

1. Banerjee, A., Basu, S. A social query model for decentralized search. In *SNAKDD* (2008).
2. Bechar-Israeli, H. From <Bonehead> to <cLoNehEAd>: nicknames, play, and identity on Internet relay chat. *J. Comput. Mediat. Commun.* (1995).
3. Brin, S., Page, L. The anatomy of a large-scale hypertextual Web search engine. In *WWW* (1998).
4. Condie, T., Kamvar, S.D., Garcia-Molina, H. Adaptive peer-to-peer topologies. In *P2P Computing* (2004).
5. Davitz, J., Yu, J., Basu, S., Gutelius, D., Harris, A. iLink: search and routing in social networks. In *KDD* (2007).
6. Dennis, A.R., Kinney, S.T. Testing media richness theory in the new media: the effects of cues, feedback, and task equivocality. *Inform. Syst. Res. 9*, 3 (Sept. 1998), 256–274.
7. Donath, J. Identity and deception in the virtual community. In *Communities in Cyberspace*, M. Smith and P. Kollock, eds. Routledge, 1999.
8. Evans, B.M., Chi, E.H. Towards a model of understanding social search. In *CSCW* (2008).
9. Faye, D., Nachouki, G., Valduriez, P. Semantic query routing in SenPeer, a P2P data management system. In *NBiS* (2007).
10. Gyongyi, Z., Koutrika, G., Pedersen, J., Garcia-Molina, H. Questioning Yahoo! Answers. In *WWW Workshop on Question Answering on the Web* (2008).
11. Hofmann, T. Probabilistic latent semantic indexing. In *SIGIR* (1999).
12. Horowitz, D., Kamvar, S.D. The anatomy of a large-scale social search engine. In *WWW* (2010).
13. Kamvar, M., Kellar, M., Patel, R., Xu, Y. Computers and iphones and mobile phones, oh my!: a logs-based comparison of search users on different devices. In *WWW* (2009).
14. Kamvar, S.D. *Numerical Algorithms for Personalized Search in Large-Scale Self-Organizing Networks*. Princeton University Press, 2010.
15. Kamvar, S.D., Horowitz, D. A Bayesian derivation of personalized and social search. Stanford University Technical Report, 2011.
16. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H. The EigenTrust algorithm for reputation management in P2P networks. In *WWW* (2003).
17. Klein, D., Toutanova, K., Ilhan, H.T., Kamvar, S.D., Manning, C.D. Combining heterogeneous classifiers for word-sense disambiguation. In *SENSEVAL* (2002).
18. Morris, M.R., Teevan, J., Panovich, K. What do people ask their social networks, and why? A survey study of status message Q&A behavior. In *CHI* (2010).
19. Page, L., Brin, S., Motwani, R., Winograd, T. The PageRank citation ranking: bringing order to the Web. Stanford University Technical Report, 1998.
20. Silverstein, C., Henziger, M., Marais, H., Moricz, M.. Analysis of a very large Web search engine query log. In *SIGIR Forum* (1999).
21. Spink, A., Jansen, B.J., Wolfram, D., Saracevic, T. From e-sex to e-commerce: Web search changes. *IEEE Comput. 35*, 3 (Mar. 2002).
22. Sproull, L., Kiesler, S. Computers, networks, and work. In *Global Networks: Computers and International Communication*. MIT Press, 1993.
23. Wesch, M. An anthropological introduction to YouTube. *Library of Congress*, 2008.

Damon Horowitz (dhoro@google.com), Google.

Sepandar D. Kamvar (sdkamvar@stanford.edu), Stanford University.

[CONTINUED FROM P. 120] "Ronald Mallett is a theoretical physicist at the University of Connecticut who suggested you can produce the same effect as the neutron-star cylinder (it's called frame dragging) in a lab using a spiraling light source instead of stars. Frame dragging is one of the weirder bits of general relativity. There's a small component of the gravitational pull at right angles to the main attraction. If you spin an object, that sideways pull drags spacetime along with it, like a spoon stirring, say, molasses. Mallett reckons there'll be a measurable effect if you can get a big enough stack of rotating light beams."

Gillian knew the team working on Mallett's idea at Caltech, intellectual home of the late Nobel physicist Richard Feynman. It wasn't difficult to get access to its frame-dragging lab when the researchers were out. Like every physics experiment I've ever seen (okay, with the exception of the Large Hadron Collider) the machine looked more Tinkertoy than Doc Brown DeLorean. The tower of laser spirals did reflect a certain majesty, admittedly, climbing up the center of a stairwell maybe 30 feet high. But the whole thing looked as if it would fall apart if you sneezed on it.

We put the copy of *Communications* I was sent in an envelope, with instructions for recipient(s) unknown to mail it to me, unopened. But there was no knowing what would really happen. We just had to hope whoever found it wouldn't get too curious and open it, sending the future veering off into some uncharted direction, perhaps sweeping us along with it.

A relativity time machine doesn't make time flow backward but acts as a shortcut to a place where time runs slower.

It just receded endlessly, timelessly, into the distance as it fell.

I saw the envelope fall down, down, down the tower like Alice in the rabbit hole. But it didn't disappear. That was the strange thing. It just receded endlessly, timelessly, into the distance as it fell. I ran down the stairs to wait by the base of the column. It ought to have dropped through. It should have been there on the oil-stained concrete floor. But it wasn't.

And so I received this message from the future. Who wrote it? Not me. I just copied it. And not that future me. It was already published when that other me put it in the envelope. But that's not the thing that puzzles me most.

I don't have a physicist friend named Gillian. I don't live near Caltech. I'm in Swindon, England, not far from Stonehenge. Ronald Mallett exists, but many physicists believe his theory is flawed, and it has yet to undergo a practical trial. I see only one possible explanation. According to the "many worlds" interpretation of quantum theory, there are countless parallel universes, and each quantum decision brings a switch of worldline for the universe we occupy. Maybe creating the paradox of this self-produced story pushed me into a different alternate universe, one with no Gillian. So... is the timelessness of the publishing schedule finished at last? Will I continue to exist myself? Or will I go out like a candle? C

Brian Clegg was a senior manager in the IT department of British Airways and is now a popular science author. His most recent book on the physics of time travel *How to Build a Time Machine* (in the U.K. *Build Your Own Time Machine*), St. Martin's Press, New York, has been featured in a variety of publications.

Future Tense, one of the revolving features on this page, presents stories and essays from the intersection of computational science and technological speculation, their boundaries limited only by our ability to imagine what could be.

DOI:10.1145/2133806.2133831

Brian Clegg

Future Tense
The Deadline Paradox

Prepare for the past ahead of time.

FIRST, I MUST admit... I copied this entire piece from the April 2012 *Communications* I had received in the mail six months ago. It was a lifesaver. I was late on deadline, so I typed it out word for word and sent it in.

When in October 2011 I opened April 2012 and read the piece supposedly written by me, my first thought was the whole thing was an elaborate prank by someone who knew I had a professional interest in time travel. Yet it was just too good, too convincing. So I took the issue to Gillian, a physicist friend at Caltech; I have a rusty physics degree but needed expert guidance.

She read it through, smiling at the mention of her name in the text. "It's clever," she said. "Whoever wrote this knows you well. But they clearly don't understand relativity. It's easy enough to build a time machine to travel into the future. Special relativity makes it inevitable that anyone moving fast enough will shift forward through time relative to those left behind. But getting an object into the past is much more of a challenge. The engineering would be extreme."

"Extreme as in difficult? Or as in impossible?"

"Impossible for a good few thousand years, at least. We're talking about crossing interstellar space, collecting maybe 10 neutron stars, slamming them together into a cylinder, somehow preventing them from collapsing into a black hole, and rotating the whole thing at cosmically high speed. And even if this happened way into the future, they couldn't have sent your piece back to you. They could never travel back to our time. That was, in fact, the problem with the MIT time travellers' convention."

"Time travellers' convention?" I shook my head.

"Back in 2005 a group of time-travel enthusiasts and physicists held a convention at the Massachusetts Institute of Technology. The idea was to attract people from the future. But no one arrived. At least no one who would admit to having arrived. They couldn't, because a relativity time machine can't travel further back than the point it was first switched on. It doesn't make time flow backward but acts as a shortcut to a place where time runs slower. There's no reaching the deep past. Unless someone has an operating time machine right now, this story couldn't have come from the future."

"And that would mean manipulating neutron stars?"

"Unless the experiment was based on Mallett's theory works."

"Which is?" [CONTINUED ON P. 119]

ILLUSTRATION BY LISEYKINA/SHUTTERSTOCK.COM

SIGMOD 2012
CALL FOR PARTICIPATION

ACM SIGMOD International Conference on Management of Data

Hyatt Regency Scottsdale Resort , Scottsdale, Arizona, USA

May 20-24, 2012

http://www.sigmod.org/2012/

Areas

- Benchmarking and performance evaluation
- Data analytics
- Data cleaning, integration, and provenance
- Data mining and knowledge discovery
- Data models, semantics, and query languages
- Data privacy and security
- Data streams and sensor networks
- Data visualization
- Database monitoring and tuning
- Databases for emerging hardware
- Distributed and parallel databases
- Indexing and physical database design
- Information extraction
- Service-oriented computing and cloud data management
- Information retrieval and text mining
- Mobile databases
- Modeling approximation and uncertainty in databases
- Query processing and optimization
- Scientific databases
- Semi-structured data
- Social networks and graph databases
- Storage systems
- Transaction management

Conference Events

- Research papers
- Industrial papers
- Technical demonstrations
- Keynotes
- Tutorials
- Panels on cutting-edge topics
- Undergraduate research papers
- New researcher symposium
- Workshops
- Industrial exhibits
- Recruitment events

General Chairs: K. Selçuk Candan (ASU)
Yi Chen (ASU)
Program Chair: Luis Gravano (U. Columbia)

Platinum: EMC², io, Microsoft, ORACLE, SAP

Gold: facebook, Google, IBM, LinkedIn, MARK LOGIC, SYBASE, turn, twitter, VERTICA, vmware

Silver: Qction, hp, GREENPLUM, @WalmartLabs, YAHOO!

Academic: ASU Arizona State University

Student Travel: NSF

Join the Conversation

GAMES ART/DESIGN FILM/TV PRODUCTION RESEARCH
PRODUCT DEVELOPMENT EDUCATION STUDENT OTHER

Bring your artistic ability, scientific innovation, and everything in between to inspire and be inspired by the most diverse gathering in computer graphics and interactive techniques.

You Are SIGGRAPH2012

The **39th** International **Conference** and **Exhibition** on **Computer Graphics** and **Interactive Techniques**

Conference 5–9 August 2012
Exhibition 7–9 August 2012
Los Angeles Convention Center

Sponsored by ACM**SIGGRAPH**

www.siggraph.org/s2012