COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

03/2010 VOL.53 NO.03

Chasing the AIDS Virus

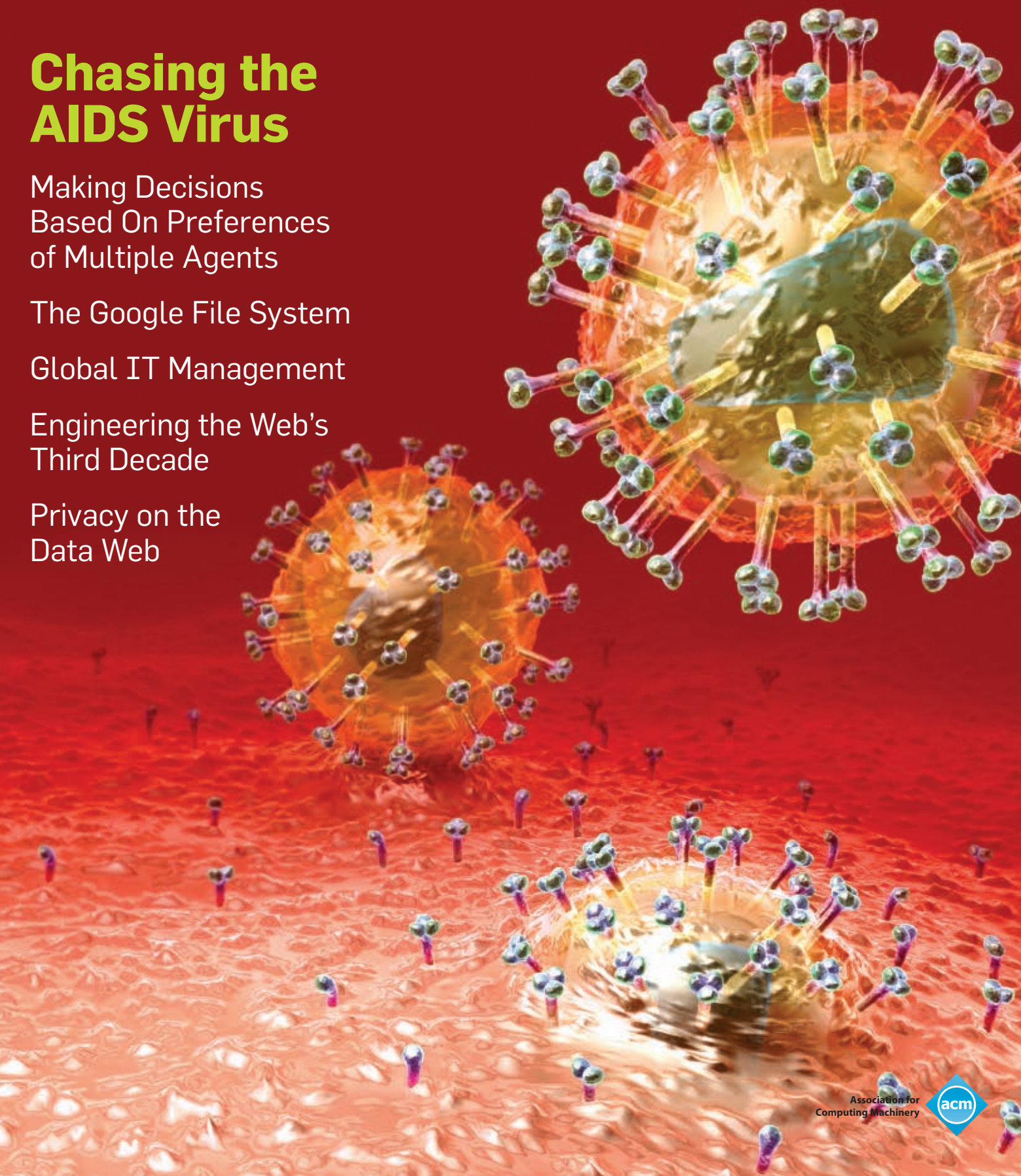
Making Decisions Based On Preferences of Multiple Agents

The Google File System

Global IT Management

Engineering the Web's Third Decade

Privacy on the Data Web



Association for Computing Machinery

Chair@…
Siobhán Clarke

Papers@…
Kevin Sullivan

Essays@…
Daniel Steinberg

Workshops@…
Jonathan Edwards

Films@…
Bernd Brügge

Onward!

2010

ACM Conference
on New Ideas
in Programming
and Reflections
on Software

October 17–21, 2010
Co-located with Splash
John Ascuaga's Nugget
…@onward-conference.org
SIGPLAN

Reno-Tahoe

Association for Computing Machinery

Advancing Computing as a Science & Profession

membership application & *digital library* order form

Priority Code: ACACM10

You can join ACM in several easy ways:

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

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

Fax
+1-212-944-1318

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

Special rates for residents of developing countries:
http://www.acm.org/membership/L2-3/

Special rates for members of sister societies:
http://www.acm.org/membership/dues.html

Please print clearly

Name

Address

City | State/Province | Postal code/Zip

Country | E-mail address

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

Purposes of ACM

ACM is dedicated to:
1) advancing the art, science, engineering, and application of information technology
2) fostering the open interchange of information to serve both professionals and the public
3) promoting the highest professional and ethics standards

I agree with the Purposes of ACM:

Signature

ACM Code of Ethics:
http://www.acm.org/serving/ethics.html

choose one membership option:

PROFESSIONAL MEMBERSHIP:

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

STUDENT MEMBERSHIP:

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

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

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

RETURN COMPLETED APPLICATION TO:

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

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

Satisfaction Guaranteed!

payment:

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

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

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

❑ ACM Digital Library ($99) $ _____

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

Total Amount Due $ _____

Card # Expiration date

Signature

COMMUNICATIONS OF THE ACM

Departments

Last Byte

News

Viewpoints

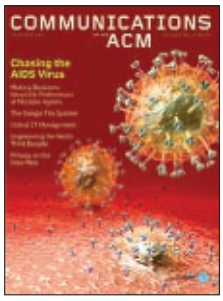
Viewpoints

Association for Computing Machinery
Advancing Computing as a Science & Profession

03/2010
VOL. 53 NO. 03

Practice

Articles' development led by ACM queue
queue.acm.org



About the Cover:
In "Chasing the AIDS Virus" (p. 66), authors detail how clinical databases and statistical-learning methods are helping virologists explore drug therapies to fight the virus that continues to confound the scientific community. Gary Carlson, a Pound Ridge, NY-based artist whose work is inspired by the medical and biological sciences (www.gcarlson.com/), captures the elusive HIV virus on this month's cover.

ILLUSTRATION BY CELIA JOHNSON

Contributed Articles

Review Article

Research Highlights

Virtual Extension

As with all magazines, page limitations often prevent the publication of articles that might otherwise be included in the print edition. To ensure timely publication, ACM created *Communications*' Virtual Extension (VE).

VE articles undergo the same rigorous review process as those in the print edition and are accepted for publication on their merit. These articles are now available to ACM members in the Digital Library.

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.

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:10.1145/1666420.1666421

Moshe Y. Vardi

Revisiting the Publication Culture in Computing Research

In my May 2009 Editor's Letter, "Conferences vs. Journals in Computing Research" (p. 5), I addressed the publication culture of our field: "As far as I know, we are the only scientific community that considers conference publication as the primary means of publishing our research results. In contrast, the prevailing academic standard of 'publish' is 'publish in archival journals.' Why are we the only discipline driving on the conference side of the 'publication road?'"

In response to my editorial, Lance Fortnow wrote a Viewpoint column (Aug. 2009, p. 33), entitled "Time for Computer Science to Grow Up," in which he concluded: "Computer science has grown to become a mature field where no major university can survive without a strong CS department. It is time for computer science to grow up and publish in a way that represents the major discipline it has become."

The May 2009 editorial and the August 2009 column attracted a lot of attention in the blogosphere. The reaction has been mostly sympathetic to the point of view reflected in both pieces. For example, Jeanette Wing asked in her blog: "How can we break the cycle of deadline-driven research?", and Filippo Menczer, in a Letter to the Editor published in the November 2009 issue, said: "I propose the abolition of conference proceedings altogether."

Not everyone, however, agreed with this point of view. For example, in another Letter to the Editor from the November 2009 issue, Jano van Hemert said: "For CS to grow up, CS journals must grow up first." Mr. van Hemert's issue with computing-research journals is that they are known to have "slow turnaround, with most taking at least a year to make a publish/reject decision and some taking much longer before publishing." Such end-to-end times, he argued, "are unheard of in other fields where journal editors make decisions in weeks, sometimes days."

While I have not see concrete data comparing publishing turnaround times for computing-research journals to those in other technical fields, there is abundance of anecdotal data supporting the claim that computing-research journals are indeed quite slow. (The average time to editorial decision for *Communications* is under two months; that takes a concerted effort by the editorial board to ensure that the editorial process does not stall.)

What is the reason for the unacceptably slow turnaround time in computing-research journals? When considering this question, one must factor the problem into two separate issues: time from submission to editorial decision, and time from positive editorial decision to publication.

First let us address the latter issue. All periodical journals have editorial "pipelines." No publisher wants to face the threat of an empty issue; it's akin to the dreaded dead air on television! Successful journals that attract many submissions often see their pipelines extend for up to two years. With the advent of electronic publishing, this problem can be eliminated or at least minimized. *Communications* uses its Virtual Extension (VE) to ensure its pipeline does not get longer than six months. VE articles undergo the same rigorous review process as those in the print edition and are accepted for publication on their merit. These articles are available in ACM's Digital Library.

Let us now consider the editorial process in computing-research journals. Why is it soooo slow? Consider who is in charge of that process. It is not the publishers; it is the editors and referees. In other words, it is *us.* The process is slow because that is the way we run it. If we want it changed, it is up to us to change it! I suspect that we cannot separate our conference-focused publication culture from our sluggish journal editorial process. Conferences have sharp deadlines, journals do not. We simply do not take our roles as editors and referees as seriously as we do as program committee members because we do not take journals as seriously as other fields. If we, as a community, decide that we need to shift from conference-based publication to journal-based publication, we definitely must address the slow editorial process, but we should not complain about "them journals." We have found the enemy, and it is us!

The 2010 Conference of the Computing Research Association (July 18–20, Snowbird, UT) will have a plenary panel on "Peer Review in Computing Research." I look forward to that discussion and hope it will help our community reach consensus on this issue.

Moshe Y. Vardi, EDITOR-IN-CHIEF

DOI:10.1145/1666420.1666422

Too Much Debate?

In his Editor's Letter "More Debate, Please!" (Jan. 2010), Moshe Y. Vardi made a plea for controversial topics on these pages, citing a desire to "let truth emerge from vigorous debate." Though we support the sentiment as well, we question Vardi's judgment in using his editorial position to mount an attack on a 30-year-old article whose authors were neither forewarned nor given the opportunity to respond. Vardi's target was our 1979 critique of formal program verification, "Social Processes and Proofs of Theorems and Programs," co-authored with the late Alan J. Perlis, winner of the first ACM A.M. Turing Award and lifelong proponent for the kind of open discussion Vardi himself advocates.

It is an extraordinary event when the Editor-in-Chief of a professional journal uses his position to declare ex cathedra that a published article is "misguided," its arguments "off the mark," and prior editors "did err in publishing [the] article... without publishing a counterpoint article..." The irony is not lost on us that we were offered no such opportunity to respond prior to publication of Vardi's Letter.

We completely disagree with Vardi's assessment and will respond to the technical substance of his comments at a later time. However, we stand by the article's two major predictions:

- That human-written proofs of real systems would not work due to the lack of the "social processes" that drive confidence in mathematical proofs. Even today, there are no human proofs of real systems; and
- That formally specifying real systems would continue to be impossibly difficult, a position since vindicated by history. Where are the formal specifications for Windows 7, thousands of iPhone apps downloaded daily, and hundreds of thousands of other systems used every day in research, commerce, and government? They do not exist.

Publication of "Social Processes and Proofs of Theorems and Programs" was not a singular event. It was refereed. A preliminary version was accepted by a highly selective conference program committee in 1976—predating by more than a year the article by Amir Pnueli that Vardi criticized us for not citing—and its presentation was attended by virtually every living contributor to the field. It was then submitted to *Communications* and reviewed by anonymous referees. Its publication was followed by months of public presentations and workshops, letters to the editor, written reinforcements and rebuttals, and—years later—a special issue of *Communications* devoted to the topic.

The article was widely read and commented on by computer scientists, engineers, and mathematicians but, rather than spark debate in the formal verification community, provoked only stony silence. A quick scan of the formal verification literature in the years 1979–1990 reveals virtually no citations to the article. In what sense is an article "controversial" if one side refuses to engage in discussion? Indeed, email circulating among the principals in the field aimed to tamp down debate and ignore our argument that many outside the field still consider substantial and prescient.

The field of formal program verification has changed substantially since 1979. Its goals have become more modest and its claims less sweeping. New methods have emerged. An equally compelling reading of history suggests that, during the long silence, the formal verification research community realized it had been misguided in 1979 and used the arguments—without attribution—set forth in the article as a roadmap to reorient its agenda.

The article itself has been reprinted dozens of times, as well as in several anthologies in the philosophy of mathematics. Donald MacKenzie's book *Mechanizing Proof: Computing, Risk, and Trust* (MIT Press, Cambridge, MA, 2001) remains the definitive sociological and historical analysis of both the article and its implications for the field. If, to Vardi, our arguments seem off the mark, then perhaps the right course is to resurrect the social process that led to the article's publication in the first place and jump into the fray. Until that time, the correct editorial position for *Communications* and its Editor-in-Chief is to let both the article (and the written record that surrounds it) speak for itself.

It is inappropriate, after 30 years of silence, to use the cover of an editorship to attack unsuspecting passersby, especially while touting the moral virtues of free and vigorous debate.

Richard A. DeMillo and Richard J. Lipton, Atlanta, GA

Author's Response:

It seems both DeMillo and Lipton feel slighted by my Editor's Letter (Jan. 2010). I had no intention of slighting them or the article in question and apologize for unintentionally causing them to feel this way.

Now to the substantive points in their comment:

1. *I am accused of using my editorial position to "mount an attack" on an article published in* Communications *in 1979. DeMillo and Lipton imply that it is inappropriate for an Editor-in-Chief to comment negatively on an article published in* Communications.

The article in question is more than 30 years old. History, it is said, "judges and rejudges." I hardly view my offering of some comments, even if critical, on such a historically important article as "mounting an attack." Personally, if someone saw the need to disagree with an article of mine 30 years after its publication, I'd feel complimented. Most articles are long forgotten after 30 years.

Regarding whether it is appropriate for an Editor-in-Chief to comment on articles published decades earlier, one should note that even the U.S. Supreme Court occasionally reverses itself. I never heard of "stare decisis," the principle that precedent decisions are to be followed by the courts, being applied to editorial matters across such a time span. (In contrast, when I assumed the position of Editor-in-Chief, I committed to respecting all prior editorial decisions in regard to pending submissions to Communications.*)*

2. I am accused of not offering DeMillo and Lipton an opportunity to respond prior to publication of my Editor's Letter. As Editor-in-Chief I write such bimonthly Editor's Letters in which I often express opinions on controversial matters. The proper way to disagree with them, and many people do, is to leave comments online or submit a letter to the editor. This is standard operating procedure in all publications I am aware of.

As Editor-in-Chief, I am committed to a scrupulous peer-review process for submitted articles, but I have not taken a vow of silence, nor does it make sense for me to do so. Furthermore, I gladly welcome the Editor-in-Chief in 2040 to reexamine my editorial decisions.

3. It seems that DeMillo and Lipton were offended by my use of the word "misguided." But one should read the full context of the word: "With hindsight of 30 years, it seems that DeMillo, Lipton, and Perlis' article has proven to be rather misguided. In fact, it is interesting to read it now and see how arguments that seemed so compelling in 1979 seem so off the mark today."

In the paragraph that preceded these sentences, I referred to two Turing Awards given for works in formal verification. Due to lack of space, I did not include references to two ACM Kanellakis Awards and two ACM Software System Awards for works in formal verification.

It is in this context that I expressed an opinion that the 1979 article, which implied the futility of formal verification as an activity and, by implication, as a research area was "misguided," with "hindsight of 30 years" in spite of "its compelling arguments."

4. DeMillo and Lipton disagree with my opinion that "the editors of Communications *in 1979 did err in publishing an article that can fairly be described as tendentious without publishing a counterpoint article in the same issue."*

The subject (and title) of my editorial was "More Debate, Please!" The article in question is one of the most controversial and influential ever published in Communications. *I read it as a graduate student and was deeply affected by it. I singled it out because it was the perfect example for making the point of my editorial, which did not focus on analyzing the 1979 article. Rather, its main point was that, in my opinion, even with 30-year hindsight, the editors in 1979 did absolutely the right thing in publishing it.*

It is precisely because the 1979 article was so influential that I chose it as an example. I honestly feel that its authors should be pleased that it is still trenchant, even if some people disagree with its major thrust.

I am well aware of the process that led to its publication in 1979. I stand behind my opinion about the lack of a counterpoint article. DeMillo and Lipton are entitled to a different opinion. We may need to agree to disagree on this one. I do not see why this is an issue that deserves such a strongly worded response, when I expressed strong support for the editorial decision to publish the article, even with the hindsight of 30 years.

5. I'd rather not respond here to DeMillo and Lipton on the merits of their article. I would, however, welcome a new article from them examining the issues they covered in 1979. I would of course seek to publish a counterpoint article in the same *issue.*

Moshe Y. Vardi, Editor-in-Chief

Give Scratch an Abstraction Mechanism

I welcome the efforts described by Mitchel Resnick et al. in "Scratch: Programming for All" (Nov. 2009) to familiarize more people with programming. However, when I downloaded Scratch from the Scratch Web site (http://scratch.mit.edu) and looked over the Scratch programming constructions, I found no convenient abstraction mechanism, as in, say, a facility to define and call parameterized functions.

Such a mechanism could be viewed as advanced and not easily digested by the intended users of the Scratch programming language. But some projects on the Scratch Web site feature significant code redundancy and could be reduced in size and simplified if the code could be restructured through a few suitable functions.

Though not all Scratch programmers would be comfortable with an abstraction mechanism, it seems a pity that something so fundamental does not even exist, and so cannot be conveniently demonstrated and disseminated.

Second in importance and also missing from the Scratch programming language is a data-structuring mechanism.

Thorkil Naur and Karen Brahes, Odense, Denmark

Authors' Response:

Abstraction is an important computational concept, and a simple form of procedural abstraction is provided by Scratch's "broadcast" mechanism. That's why we added parameterized procedures to some experimental versions of Scratch, though we have not yet come up with a design that satisfies our goals of simplicity and understandability. We're continuing to experiment, hoping to include more forms of abstraction in future versions.

The Scratch Team, Cambridge, MA

Recognition for the Unaffiliated, Too

I was heartened by Wendy Hall's interest, as expressed in her President's Letter "ACM Europe" (Oct. 2009), in student chapters, award nominations, and conferences sponsored by the ACM in Europe.

I regularly seek out opportunities for public recognition and awards for ACM members not affiliated with universities. For example, the traditional rule requiring three or more endorsements for a researcher to be considered for an award is a barrier to would-be nominees not affiliated with universities or in the pool of preferred students of their academic mentors. The situation is even more problematic if an individual's research is based on his/her long-standing experience in an area of expertise not currently "popular" in universities.

I therefore suggest the ACM in Europe establish a committee to consider self-nominations and invite volunteers from among the young researchers who promote computer science in their spare time, rather than as salaried academics.

Concerning conferences and other events, I'd also like to propose ACM set up summer schools open to all enthusiasts who promote electrical engineering and computer science. Locating them in popular tourist areas would be another way for ACM in Europe to increase interest in more traditional ACM activities and individual memberships.

Miroslav Skoric, Novi Sad, Serbia

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

DOI:10.1145/1666420.1666424

In the Virtual Extension

Communications' *Virtual Extension brings more quality articles to ACM members. These articles are now available in the ACM Digital Library.*

Is Stickiness Profitable for Electronic Retailers?

Lin Lin, Paul Jen-Hwa Hu, Olivia R. Liu Sheng, and Johnny Lee

Current e-commerce practices suffer from a lack of accurate bottom-line performance metrics. Although conventional wisdom suggests that measurements such as stickiness or number of visitors might offer a clue, no empirical evidence to date has shown any conclusive proof for such assumptions. The authors analyze the relationship between customers' in-session visiting behavior measured by "stickiness" and their conversion behavior. This study directly answers the question: "Does visiting behavior measurement really serve as an effective tool for predicting customer purchase intentions?" Their findings greatly improve our understanding of the phenomenon under study and would have immediate impact on current business practice.

Practitioner-Based Measurement: A Collaborative Approach

S.T. Parkinson, R.M. Hierons, M. Lycett, and M. Norman

It is widely understood that a program to improve software quality can be expected to recoup its cost many times over. The authors put forward two distinctly different models as the way to successfully implement such programs. This work defines a hybrid, practitioner-based model and evaluates the implementation of a measurement framework in a major insurance organization. Research was conducted to understand the critical success factors in implementing software measurement programs, develop a measurement framework to address these factors, implement a pilot program, and reflect on the outcomes.

Organizational Adoption of Open Source Software: Barriers and Remedies

Del Nagy, Areej M. Yassin, and Anol Bhattacherjee

Considerable excitement in the business community surrounds open source software as these applications appear to offer increased contextual functionality or technical performance along with reduced costs. Several barriers, however, prevent organizations from easily adopting these technologies. The authors examine adoption barriers surrounding organizational knowledge, legacy integration, open source software forking, sunk costs, and technological immaturity, as well as provide potential remedies to these barriers for organizations looking to adopt open source software.

Aligning Undergraduate IS Curricula With Industry Needs

John H. Benamati, Zafer D. Ozdemir, and H. Jeff Smith

Industry executives now seek IS graduates with higher-level skills. The vast majority of the top business schools (69%) have made recent curricular changes consistent with changing industry demands. Across MIS programs in top 50 business schools, the collective number of IS graduates was down 60% from 2003 to 2007. From 2006 to 2007, schools with changes combined to graduate 19% more MIS students while the number of graduates continued to decline in schools with no curricular changes. A coordinated effort by industry executives and academics is required to address IS industry demand for both skills and number of graduates.

Agent-Oriented Embedded Electronic Measuring Systems

Hing Kai Chan

Most of the reported literature regarding agent technology have been focusing on the theoretical foundations of agent applications. This article, in contrast, sets out to discuss two real-life applications of agent technology on embedded electronic measuring systems. The author discusses the reason why agent technology was employed in each case as well as addresses the difficulties that occur during the course of design, agent-based software development, and implementation. Pros and cons with respect to the two applications are presented, allowing readers to gain insights into why, and how, agent-technology could be applied in real-life applications.

Business Continuity and the Banking Industry

Fabio Arduini and Vincenzo Morabito

Recent natural disasters and acts of terrorism have propelled renewed interest in emergency planning in both the private and public sector. Business continuity (BC) is fast becoming a key task within all industrial and business specificities. The authors focus on the importance of BC strategies throughout any organization, particularly the banking industry where management has often depended on technologies it does not fully understand. BC planning should be considered a businesswide approach and not an IT-focused one, the authors warn. Moreover, such planning must be an ongoing commitment adopted among the various levels of management within an organization.

User Participation in Software Development Projects

Ramanath Subramanyam, Fei Lee Weisstein, and M.S. Krishnan

Eliciting user input has been considered crucial for successful software development. Consistent with this notion, both researchers and practitioners have viewed user participation as an important way to improve software quality, increase user satisfaction, and promote user acceptance. Product development leaders and project managers might lean toward increasing the users' input into the development process. However, empirical evidence also shows that user participation might negatively influence performance by making the process more difficult, lengthy, and less effective. In this study, the authors empirically examine both the 'developer-side' and 'user-side' impacts of user participation and underscore the need to carefully manage customer-team interactions.

A Framework for Health Care Information Assurance Policy and Compliance

Sherrie Drye Cannoy and A.F. Salam

As many as 400 people may have access to one's personal medical information throughout the typical care process. Patients and consumers need to feel their sensitive electronic records or information are protected against unauthorized access, transmission, and disclosure. HIPAA and related policies ensure that health records are kept confidential. However, if employees fail to understand compliance policies, it becomes difficult to keep patient information confidential. Based on a multi-site case study, this article presents a framework of Information Assurance Policy and Compliance factors addressing the behavioral dimension in the context of patient health care information.

25TH ANNUAL ACM CONFERENCE ON SYSTEMS, PROGRAMMING, LANGUAGES, APPLICATIONS: SOFTWARE FOR HUMANITY

MARCH 25, 2010
Submission deadline for
OOPSLA Research Papers,
Practitioner Reports,
Educators' and Trainers' Symposium,
and proposals for Tutorials,
Workshops, Panels

APRIL 23, 2010
Onward! Papers and Essays

JUNE 24, 2010
Submission deadline for Posters,
Demonstrations, Doctoral Symposium,
Onward! Films, and Student Research
Competition and Volunteers

LOCATION
John Ascuaga's Nugget Hotel
Reno/Tahoe Nevada USA

COLOCATED CONFERENCES
Onward!
Dynamic Language Symposium (DLS)
Pattern Languages of Programs (PLoP)
and more

CONFERENCE CHAIR
William R. Cook, UT Austin
chair@splashcon.org

OOPSLA PROGRAM CHAIR
Martin Rinard, MIT
program@splashcon.org

For information, please contact
ACM Member Services Department
1-800-342-6626 (US & Canada)
+1-212-626-0500 (global)
info@splashcon.org

SPLASH/OOPSLA is sponsored by
ACM SIGPLAN and SIGSOFT

WWW.SPLASHCON.ORG

ACM SIGPLAN/SIGSOFT
announce
OOPSLA
is now part of
SPLASH

SPLASH

RENO 2010
OCTOBER 17-21

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 excerpts from selected posts.

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

DOI:10.1145/1666420.1666425 http://cacm.acm.org/blogs/blog-cacm

Too Much Programming Too Soon?

Mark Guzdial and Judy Robertson discuss the role of programming in introductory computer science.

From Mark Guzdial's "How We Teach Introductory Computer Science is Wrong"
http://cacm.acm.org/blogs/blog-cacm/45725

I've been interested in John Sweller and *Cognitive Load Theory* since reading Ray Lister's 2008 Australasian Computing Education Conference keynote paper, "After the Gold Rush: Toward Sustainable Scholarship in Computing." I assigned several papers on the topic to my educational technology class. Those papers have been influencing my thinking about how we teach computing.

In general, we teach computing by asking students to engage in the activity of professionals in the field: by programming. We lecture to them and have them study texts, of course, but most of the learning is expected to occur through the practice of programming. We teach programming by having students program.

The original 1985 Sweller and Cooper paper on worked examples had five studies with similar setups. There are two groups of students, each of which is shown two worked-out algebra problems. Our experimental group then gets eight more algebra problems, completely worked out. Our control group solves those eight problems. As you might imagine, the control group takes *five times* as long to complete the eight problems than the experiment group takes to simply read them. Both groups then get new problems to solve. *The experimental group solves the problems in half the time and with fewer errors than the control group. Not* problem-solving leads to better problem-solving skills than those doing problem-solving. That's when educational psychologists began to question the idea that we should best teach problem-solving by having students solve problems.

The paper by Kirschner, Sweller, and Clark (KSC) is the most outspoken and most interesting of the papers in this thread of research. Their title states their basic premise: "Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching." What exactly is minimal instruction? And are they really describing *us*? I think this quote describes how we work in computing education pretty well:

"There seem to be two main assumptions underlying instructional programs using minimal guidance. First[,] they challenge students to solve "authentic" problems or acquire complex knowledge in information-rich settings based on the assumption that having learners construct their own solutions leads to the most effective learning experience. Second, they appear to assume that knowledge can best be acquired through experience based on the procedures of the discipline (i.e., seeing the pedagogic content of the learning experience as identical to the methods and processes or epistemology of the discipline being studied; Kirschner, 1992)."

That seems to reflect our practice, paraphrased as "people should learn to program by constructing a program from the basic information on the language, and they should do it in the same way that experts do it." The paper then presents all the evidence showing that this "minimally-guided instruction" does not work:

"After a half-century of advocacy associated with instruction using minimal guidance, it appears that there is no body of research supporting the technique. In so far as there is any evidence from controlled studies, it almost uniformly supports direct, strong instructional guidance rather than constructivist-based minimal guidance during the instruction of novice to intermediate learners."

There have been rebuttals to this article. What's striking about these re-

buttals is that they basically say, "But not problem-based and inquiry-based learning! Those are actually guided, scaffolded forms of instruction." What's striking is that *no one challenges KSC on the basic premise, that putting introductory students in the position of discovering information for themselves is a bad idea*! In general, the educational psychology community (from the papers I've read) says that expecting students to program as a way of learning programming is an ineffective way to teach.

What should we do instead? That's a big, open question. Pete Pirolli and Mimi Recker have explored the methods of worked examples and cognitive load theory in programming, and found that they work pretty well. Lots of options are being explored in this literature, from using tools like intelligent tutors to focusing on program "completion" problems (van Merriënboer and Krammer in 1987 got great results using completion rather than program generation).

This literature is *not* saying *never* program. Rather, it's a bad way to *start.* Students need the opportunity to gain knowledge first, before programming, just as with reading. Later, there is an *expertise reversal effect*, where the worked example effect disappears, then *reverses.* Intermediate students do learn better with real programming, real problem-solving. There *is* a place for minimally guided student activity, including programming. It's just not at the beginning.

Overall, I find this literature unintuitive. It seems obvious to me that the way to learn to program is by programming. It seems obvious to me that real programming can be motivating. But KSC respond to this, too, noting that "it is easy to share the puzzlement of Handelsman et al. (2004), who, when discussing science education, asked":

"Why do outstanding scientists who demand rigorous proof for scientific assertions in their research continue to use and, indeed defend on the bias of intuition alone, teaching methods that are not the most effective?"

This literature doesn't offer a lot of obvious answers for how to do computing education better. It does, however, provide strong evidence that what we're doing is *wrong*, and offers pointers to how *other* disciplines have done it *better*. It's a challenge to us to question our practice.

From Judy Robertson's "Introductory Computer Science Lessons—Take Heart!"

http://cacm.acm.org/blogs/blog-cacm/46781

I was somewhat alarmed to read Mark Guzdial's excellent and thought-provoking post, which argues that the way we teach introductory computer science is wrong. His argument is that some of the educational psychology literature claims that minimally guided instruction techniques, such as discovery learning, constructivism, and problem-based learning, are less effective than strongly guided instruction techniques. As an extension to this: teaching programming through the practice of programming itself is not effective for novices. As a lecturer of a first-year programming module myself, I spluttered into my cup of tea and hurried off to read the Kirschner, Sweller, and Clark article Mark recommended.

Kirschner, Sweller, and Clark have some strong words to say against minimally guided instruction approaches. For example, "The goal of instruction is rarely simply to search for or discover information. The goal is to give learners explicit guidance about how to cognitively manipulate information in ways that are consistent with a learning goal, and store the result in long-term memory." But hang on: in higher education we generally regard it as important that students know how to search and discover information for themselves. They require skills in self-directed learning. In the context of programming, for example, we may wish them to know how to look up documentation. We would also generally expect them to be able to search for information sources in the first stage of carrying out a research project. I suspect this is a question of the stage of cognitive and metacognitive development the learner is at in first year, and whether it is reasonable to expect more of them than manipulating information and storing it in long-term memory.

The authors also write: "[It] may be a fundamental error to assume that the pedagogic content of the learning experience is identical to the methods and processes (i.e., the epistemology) of the discipline being studied and a mistake to assume that instruction should exclusively focus on methods and processes."

I don't think that introductory computer science teaching *does* focus only on methods and processes. In fact, it is a bit of a straw man to consider what goes on in first-year computer science classes as pure minimally guided instruction anyway. Obviously there is a huge range of teaching approaches to novice programming across the world, but let's take Barnes and Kölling's *Objects First With Java* textbook and the BlueJ environment. It's very popular and used as an introductory text in many computer science departments. One of the features of this well-designed textbook is that it aims to teach high-level concepts as a priority over lower-level language constructs. The BlueJ environment enables students to experiment with object orientation by calling methods on objects in a graphical environment. The textbook encourages students to read code before they write it, and "wire in" small segments of their own code into a prewritten program. The lecture slides that come with the book give specific instruction and worked examples; students typically receive this sort of instruction before working on small examples in the lab. In fact, working on small examples after a lecture on programming concepts is, in my experience, a fairly common pattern in first-year instruction.

Kirschner, Sweller, and Clark recommend: a) providing worked examples for students to read and, b) providing process worksheets that explain to students the processes they should go through when solving problems. These are sensible suggestions, but I wouldn't say they are unusual for computer science teaching. I would suggest that we tend to use a mixed bag of instructional techniques rather than basing our pedagogy on pure theory. And therefore, we probably get our first-year teaching right at least part of the time. Which is a bit of a comfort.

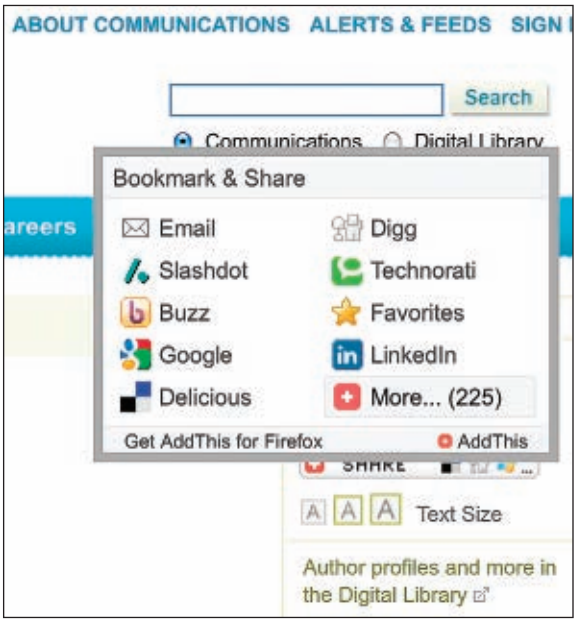
Mark Guzdial is a professor at the Georgia Institute of Technology. **Judy Robertson** is a senior lecturer at Heriot-Watt University.

DOI:10.1145/1666420.1666426 David Roman

Granting a Second Life

Almost 90% of what we learn comes from reading, estimates say, and the path that online information takes to your brain is less direct than most. Search engines send 61% of traffic to the average Web site, and referring sites send 17%. *Communications*' site reverses these numbers but still gets most of its readers from someplace else: 56.6% from referring sites, like Slashdot or Reddit, and 14.3% from search engines. Google alone sends 13.6%.

Referring sites are *Communications*' hitmakers. They sent most readers to the site's most popular stories in the past year. Over 75% of the online readers of "You Don't Know Jack About Software Maintenance" (http://cacm.acm.org/magazines/2009/11/48444) and over 78% of those who read "The Status of the P Versus NP Problem" (http://cacm.acm.org/magazines/2009/9/38904) were referred. These stories became hits days after they were posted. "What Should We Teach New Software Developers? Why?" (http://cacm.acm.org/magazines/2010/1/55760) spiked three days after it was posted. Referring sites can give stories a second life as well. "Logic of Lemmings in Compiler Innovation" (http://cacm.acm.org/magazines/2009/5/24633) took off more than two months after it was posted.



Referring sites are a Web version of word of mouth. The force behind them is readers like you. Motive doesn't matter. Magnanimity, didacticism, and egoism all produce the same results. They put *Communications*' articles in front of a larger, mostly different audience. If you think something you've read on http://cacm.acm.org deserves extra attention, the simplest way to share it is with the SHARE button located on the right column of every article page. The 224 sites accessible from the button at last count testify to its ease of use. Raising a story's profile not only brings more traffic to the site, but shares and extends ACM's profile and membership value to an even greater, global audience.

ACM Member News

SIGGRAPH'S DEBEVEC WINS ACADEMY AWARD

SIGGRAPH Executive Committee Director-at-Large Paul Debevec recently received a Scientific and Engineering Award from the Academy of Motion Picture Arts and Sciences. Debevec, with Tim Hawkins, John Monos, and Mark Sagar, were recognized for the design and engineering of the Light Stage capture devices and the image-based facial rendering system developed for character relighting in motion pictures.

In an email interview, Debevec, who leads the graphics laboratory at the University of Southern California's Institute for Creative Technologies, discussed the Light Stage capture devices' computational challenges. "Our classic Light Stage process built realistic computer graphics models of actors by taking photographs of the actor's face under hundreds or thousands of different lighting directions, often from multiple viewpoints. This allowed an image-based approach to rendering the actor under complex lighting environments by computing linear combinations of the images taken under the different conditions. The imagery was a huge amount of data, especially when it was scanned at film resolution. One observation we made was that the linear combinations could be computed even from image data projected onto a compressed basis. We could thus relight the actor's face by directly recombining the compressed image data, and then decompressing the result. Using this technique, our Face Demo software (http://www.debevec.org/FaceDemo/) could relight human faces in real time even back in the year 2000."

Currently, Debevec and colleagues are making the process of creating animated digital characters from their Light Stage data much more automatic, trying to improve on the results of their recent Digital Emily project. —*Jack Rosenberger*

Science | DOI:10.1145/1666420.1666427

Gregory Goth

CS and Biology's Growing Pains

Biologists can benefit from learning and using the tools of computer science, but several real-world obstacles remain.

The compatibility of computer science and biology—two disparate yet increasingly symbiotic branches of knowledge—is becoming a hot topic among academic scientists. Recent publications in popular and academic journals have called for mandating stronger computer and mathematics courses for undergraduate biology majors. Those treatises have been met by equally ardent responses among some biologists claiming that mandating additional background in computer science and math will not necessarily advance a budding biologist's academic and career success.

"To grossly oversimplify it, computer science is all about the binary, and in biology, things don't lend themselves to binary distinction," says John Timmer, the science editor of Arstechnica.com, who has a Ph.D. in molecular and cell biology. Timmer recently wrote an opinion piece, "Should Biologists Study Computer Science?", that took to task advocates of increased emphasis on undergraduate computer science and math. Timmer argued that knowing how to use a given tool, and having enough domain knowledge to be able to flag outlying results, should be sufficient for most biologists.

A New Jersey high school student works in a Rutgers University lab as part of a research project on decoding a DNA sequence.

"Obviously, computer scientists can do things that are far more subtle than binary logic," Timmer says, "but the fact that the most basic concepts in biology, like genes and species, exist along a full spectrum and can often be defined using different definitions doesn't lend itself to definitive computerized analysis very cleanly."

Computer scientist Nir Piterman, a research fellow at Imperial College, says Timmer may be right, but that "the central role of the computer in our lives" will mandate that biologists learn some foundational basics of computation such as algorithmic thinking and some sort of formal expression.

"The advantages are not only in being able to do the things that are required in order to do modeling or more computational biology," says Piterman, "but this way of thinking can help many fields of biology to communicate better, and to harness computing better, by being able to share information more formally. Maybe it's less natural to do it in biology, but the power of computing makes it less than optimal to avoid this."

A High-School Solution?

The goal to strengthen biologists' computer science and math backgrounds faces a major obstacle within college curricular structures. For instance, trying to design a quantitative thinking and computer science offering that would satisfy all fields of biology is extremely difficult. Also, students' schedules are already filled with existing requirements. Adam Siepel, assistant professor of biological statistics and computational biology at Cornell University, says the university is grappling with this issue.

"There's such a broad spectrum of activities going on under the rubric of biology, from what is essentially physiology to organismal biology, to ecology," Siepel says. "These disciplines have almost nothing to do with one another. I was part of a task force last year that was reviewing the undergraduate

PHOTOGRAPH BY MARTIN NEMEROFF, RUTGERS UNIVERSITY

curriculum for biology and it was really a struggle."

Siepel says the math requirements were examined closely, but the faculty concluded that sending biology students out of the department for math, computer science, and statistics survey courses was unpopular and counterproductive.

"There was general agreement the students should have something that really connects better with biology, maybe less calculus, more statistics and computer science, maybe something about computational sequence analysis or something along those lines," Siepel says. "But it's a struggle. The students already have a full set of requirements and any time you add a new one, you have to bump something else. We didn't get very far on that issue. You get in a situation where you almost have to require a five-year instead of a four-year degree if you're really going to educate them in the physical sciences and math and statistics and computer science as well as all the biology requirements."

Siepel says he has had numerous students who want to take an upper-level course and express an interest in some aspect of computational biology, only to discover they lack a sufficient background in math or computer science to really pursue that interest. And, Siepel reiterates, their schedules are already too full.

"To be frank," says Siepel, "part of it is the failure of high schools to be providing basic education in mathematics and sciences before the students get to universities."

That shortcoming may be addressed soon. In 2009, the College Board released the draft of its revised Advanced Placement (AP) biology curriculum for high school seniors in response to the National Research Council's 2002 report *Learning and Understanding: Improving Advanced Study of Mathematics and Science in U.S. High Schools*. The new curriculum includes significant changes in four areas, including quantitative and computational thinking. According to the College Board draft, "Students will be encouraged to develop their ability to apply mathematics to wide sectors of biology so that they can better test hypotheses, model biological phenomena, interrogate complex data sets, and represent and interpret visualizations of relationships."

Raina Robeva, chair of the mathematical sciences department at Sweet Briar College, says the new AP curriculum should have a profound effect on incoming students' capabilities. "Whether we like it or not, the College Board drives a lot of this, so if they are saying they are changing all of this, the AP exams will change to reflect this and all those students will have more of a quantitative background when they get to college, and will take those skills to higher-level biology courses."

As part of a National Science Foundation-funded project, New Jersey high school students conduct bioinformatics research on lab computers at Rutgers University.

PHOTOGRAPH BY MARTIN NEMEROFF, RUTGERS UNIVERSITY

Real-World Dilemmas

Even if fundamental concepts are added to advanced secondary school curricula and undergraduate courses, the workaday problem of reconciling the principles of computer science and math with the realities of biologic research remains. Sarah Killcoyne and John Boyle, senior software engineer and senior research scientist, respectively, at the Institute for Systems Biology, co-authored "Managing Chaos: Lessons Learned Developing Software in the Life Sciences," in the November-December 2009 issue of *Computing in Science and Engineering*.

In their paper, Killcoyne and Boyle pointed out that biology, due to its descriptive nature, lacks the grand underlying mathematical theory, and hence formalized body of expression, that is present in physics. This makes software development far more difficult in life sciences, and the two communities remain struggling to communicate their needs. Boyle says teaching biologists and computer scientists an appreciation for each other's discipline might be more useful than trying to convince biologists they need a certain amount of computing and math proficiency to do their jobs.

"You hate to say this, but a lot of people don't care, and rightly so," Boyle says. "They're busy people. Should they know the ins and outs of how to use a bioinformatics tool? In a perfect world, yes, they should. But is it something that's holding back scientific progress? Can they go to someone else and get that person to help them? Yes. Can they get by without it? Sometimes.

"We tend to be a little bit pragmatic here. 'Is it something that's holding us back doing research?' is always going to be the fundamental question," says Boyle.

Perhaps the debate over exactly how computationally savvy the majority of biologists should be will devolve simply due to the fact that certain areas of biology will naturally lend themselves to more computationally intensive ap-

Part of the problem, says Adam Siepel, "is the failure of high schools to be providing basic education in mathematics and sciences before the students get to universities."

proaches than others. Boyle says the contention that a number of computationally skilled biologists specializing in these areas will advance the cross-pollination of the disciplines in a kind of natural selection process may have credence. A researcher at Microsoft Research Cambridge, Jasmin Fisher is a pioneer of this sort of "executable biology," which she says will not only winnow out false steps in the process of evaluating an idea, but also illuminate hypotheses for which noncomputational calculations would be prohibitively difficult or missed altogether.

"Serious biological research with living material takes a long time," Fisher says. "The thing we're trying to say here is this kind of modeling will help to focus and direct the next experiment and save time and resources. This is the key point."

One example of such an approach is work Fisher and colleagues, including Piterman (who is married to Fisher), computer scientist Tom Henzinger (who is president of the Institute of Science and Technology Austria), and University of Zurich biology professor Alex Hajnal, performed while studying earthworm vulva development.

"While modeling the crosstalk between two signaling pathways operating in the cells that eventually become the worm's egg-laying system, we predicted a very specific order of events related to this particular developmental process," Fisher says. "This then led to the design of an experiment that was performed in the lab, and validated experimentally the prediction provided by the modeling work. The point here is that, one, without the modeling work this prediction would not have been thought of, and, two, without the prediction, the experiment would not have been designed and performed in the lab. I think this is a beautiful example of how this kind of knowledge from computer science can be channeled to direct lab experiments and shed new light on the biological system that we study."

Whatever approach the two disciplines' practitioners ultimately decide upon to create a more seamless interaction between them, Robeva says the heightened level of discussion, disagreements and all, is beneficial for both disciplines in crafting a more compatible future.

"It used to be the case that biology needed math, and mathematicians would answer a biologist's problem out of a sense of community service," she says. "But now biology problems are generating way more math questions than mathematicians can answer. It seems at this juncture that momentum is going for both the biologists and the mathematicians, so it seems the stars are aligning." C

Further Reading

Fisher, J. and Henzinger, T.A.
Executable cell biology. *Nature Biotechnology 25*, 11, November 2007.

Pevzner, P. and Shamir, R.
Computing has changed biology—biology education must catch up. *Science 325*, 5940, July 2009.

Robeva, R. and Laubenbacher, R.
Mathematical biology education: Beyond calculus. *Science 325*, 5940, July 2009.

Siepel, A.
Computational education for molecular biology and genetics. *Transform Science: Computational Education for Scientists*, Yan Xu (ed.), Microsoft Corp., Redmond, WA, 2009.

Boyle, J., Cavnor, C., Killcoyne, S., Shmulevich, I
Systems biology driven software design for the research enterprise. *BMC Bioinformatics 9*, 295, June 2008.

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

Artificial Intelligence

Israel's Robotic Warfare

Israel is leading the world in the development of robotic fighting machines, according to a recent article by Charles Levinson in *The Wall Street Journal*. The article, "Israeli Robots Remake Battlefield," attributed Israel's role as one of the world's top military robotic innovators to its six decades of almost uninterrupted warfare, a low acceptance for enduring human causalities, and an agile and robust high-tech industry.

"We're trying to get to unmanned vehicles everywhere on the battlefield for each platoon in the field," Lt. Col. Oren Berebbi, head of the Israel Defense Forces' technology branch told *The Wall Street Journal*. "We can do more and more missions without putting a solider at risk."

One-third of Israel's military machines will be unmanned in the next 10–15 years, predicts Giora Katz, vice president of Rafael Advanced Defense Systems Ltd., a leading Israeli weapons manufacturer.

Israel's robotic machines include the long-range Heron drone, which can fly continuously at an altitude of 30,000 feet for 30 hours; the Guardium unmanned ground vehicle, an armored golf cart equipped with optical sensors and surveillance gear, which is used to patrol the Gaza and Lebanese borders; remote-controlled bulldozers that open supply routes and transport food and ammunition through hostile territory to the front lines; and the Protector SV, a nine-meter-long, well-armed speedboat that constitutes a growing part of the Israeli navy.

Coming soon is a six-wheeled Rex robot, which can carry 550 pounds of equipment alongside advancing troops.

More than 40 nations possess military robotics programs, with many of them focusing on aerial drones. A military robotics milestone was reached last year when the U.S. Air Force, for the first time ever, trained more drone operators for its unmanned aircraft than it did pilots for its manned fighters and bombers.

Technology | DOI:10.1145/1666420.1666428 Kirk L. Kroeker

Engineering the Web's Third Decade

As Web technologies move beyond two-way interactive capabilities to facilitate more dynamic and pervasive experiences, the Web is quickly advancing toward its third major upgrade.

Researchers working on the next generation of Web technology tend to avoid hyperbole, using language more cautious than the erstwhile bravado frequently exhibited by Internet evangelists prior to the big dot-com bust. Today, the Web is quickly advancing toward its third decade and to what many are calling its third major upgrade. It is moving beyond mere two-way interactive Web 2.0 technologies to a more dynamic, pervasive, and perhaps even more human experience. Indeed, as Web 3.0 emerges, those working at the forefront of Internet technology research tend to speak with guarded language, suggesting the next major advancements in Web technologies might be more evolutionary than revolutionary—at least for now.

A Rensselaer Polytechnic Institute application for location-aware phones accesses Facebook and other online sources to make wine recommendations for a particular group of friends.

Use of the term "Web 3.0" to describe the Web's next major developments has become loaded, capable of connoting very different implications for technology and society. At least one popular idea hovering around use of the term Web 3.0 is that Web 3.0 technologies will help filter the "wisdom of the crowd" so that it doesn't become the "madness of the mob." Critics of this position suggest this way of thinking will contribute to a reduction of the kind of democratization on the Web that made it so popular as a medium for information sharing, social interaction, and other forms of expression.

Richard Stanton, chief executive of Bintro.com, a NY-based company that bases its business model on emerging Web 3.0 technologies, sidesteps the Web 3.0 terminology controversy and points out that Web 3.0's social implications can be defined simply by focusing on the personal. "Data becomes much more valuable and has a much bigger return when we tailor users' experiences to their individual needs," Stanton says. "The more fulfillment one gains from personal experiences on the Web, the better off the masses will be, whether it is democratic, meritocratic, or anything in between."

From a technology standpoint, researchers suggest a key aspect of Web 3.0 technology is moving beyond Web 2.0's popular Asynchronous JavaScript and XML (AJAX) model to one more infused with semantic technologies that facilitate interlinked data and customizable, portable applications that are device- or system-neutral. Jim Hendler, for example, suggests viewing Web 3.0 simply as Semantic Web technologies powering large-scale Web apps. "The problem is that, like Web 2.0 before it, the term can be taken many ways," says Hendler, a professor in the computer and cognitive science departments at Rensselaer Polytechnic Institute (RPI).

"Many people use Web 3.0 to mean Web applications that use semantic technologies, while others tend to use it to mean anything that fixes the many known problems with Web 2.0," he says. "I tend to like [Radar Networks CEO] Nova Spivack's idea that the version numbers correspond more to Web decades than to specific technologies, and that 3.0 will be the term used for all the new technologies emerging over the coming third decade of the Web."

Debates about the merits of Web 3.0 as a label for emerging Internet technologies aside, Hendler's own work focuses largely on Semantic Web technology and in particular on scalable reasoning and data-on-demand systems. "We are looking at technologies that could, on the fly, find and merge appropriate pieces of very large data sets into custom data caches and make those available in Web applications," Hendler says. The key, he notes, is finding a trade-off that is more efficient than the traditional knowledge rela-

tionships that researchers working in AI might use, but more powerful than the relational models that have been the hallmark of database research.

Hendler is working with the data that the U.S. government is releasing in the data.gov project with the purpose of making it available in Semantic Web formats. In practical terms, Hendler and his team are focused on linking the data to other data sets and connecting it into information sources in what researchers are now calling the "linked open data cloud," a set of data sets that have partial mappings to other data sets and domains, so that developers can mash up the data and write Web apps on top of it.

In another project, Hendler is using supercomputers to scale Semantic Web algorithms to extremely large data sets. "We've been playing with graphs that have over a billion triples [the assertions underlying the Semantic Web]," he says. "There's really only a small number of groups working on this approach, and we think we're the only U.S. group in the space, so it is great fun." As it turns out, Hendler and his team at RPI have been able to engineer new kinds of parallelization for Semantic Web processes. He says these developments might soon enable his team to migrate the algorithms to commodity hardware to power large-scale Web apps used by millions of people.

Billions of Triples

Despite the promising developments, challenges in this area remain. While Hendler and his team are experimenting with billions of triples, the Open Calais project, one of many new endeavors in this area, is creating 800 million triples each week. "In the way the Web has of making scale critical, the numbers are growing really big, really fast," says Hendler. Such scale will become even more of an issue as more applications begin linking to other apps through the Semantic Web layer.

Another researcher working on how Semantic Web technologies can facilitate information handling and more precise representations is Ora Lassila, a senior data technologist at Nokia Services and a member of the Nokia CEO Technology Council. Lassila is focused on a specific aspect of Semantic Web technologies, which he calls "provenance"—that is, where a piece of information came from, who generated it, and when. One of his goals is to facilitate the transformation of Web 2.0 mashups so that information bits from multiple sources can still retain their provenance.

Web 3.0's semantic technologies will facilitate interlocked data and customizable, portable apps that are device- or system-neutral.

"Thus, you would be able to dissect information and better understand its reliability and trustworthiness," Lassila says. In this line of thinking, Lassila rejects the notion that Web 3.0 might lead to a reduction in democratization. "It seems to me," he says, "that making it easier to disseminate trustworthy information would have the opposite, positive effect."

Lassila says he is surprised at how quickly Semantic Web ideas have been embraced by Internet developers, particularly in the past few years during which many Semantic Web ideas and formats have been adopted by even large Internet companies. For example, Google now supports a technology called Rich Snippets and Yahoo has created Search Monkey, both of which rely on Semantic Web strategies. Powerset's semantic technology—acquired by Microsoft in 2008—is reportedly a significant component of Bing, Microsoft's search engine.

As another example, Bintro.com is using semantics to enhance matching technologies and simplify the way users' needs are fulfilled online. Bintro's technology combines public semantic knowledge bases with the company's own knowledge base, which includes subject-specific terminology and jargon. Bintro uses semantic data that in

Employment

A Bright Job Outlook

If you're having second thoughts about a career in IT or CS in the United States, you should set those thoughts aside, according to Ed Lazowska, the Bill & Melinda Gates Chair in Computer Science & Engineering at the University of Washington. In "Where the jobs are...." (http://www.cccblog.org/2010/01/04/where-the-jobs-are/), a post for The Computing Community Consortium blog, Lazowska recently analyzed the U.S. Bureau of Labor Statistics' new 10-year forecast of job growth in all fields of employment and found the outlook for computer and mathematical jobs to be truly rosy.

In the category of professional and related occupations, which includes computer science, the projected growth between 2008 and 2018 is 16.8%. In contrast, the average growth across all occupations is projected to be 10.1%.

Of the eight occupational clusters in the professional and related category, computer and mathematical occupations "are projected to grow by the largest percentage between now and 2018—by 22.2%," Lazowska notes. "In other words, 'Computer and mathematical' occupations are the fastest growing occupational cluster within the fastest growing major occupational group.

"Looking at all science and engineering occupations—computer and mathematical, architecture and engineering, and life, physical, and social science—computer science occupations are projected to be responsible for nearly 60% of all job growth between now and 2018," Lazowska writes. "The next largest contributor—*all fields of engineering combined*—is projected to contribute 13.4% of total growth. All of the life sciences combined: 5.6%. All of the physical sciences combined: 3.1%."

"In other words," Lazowska concludes, "among *all* occupations in *all* fields of science and engineering, computer science occupations are projected to account for nearly 60% of *all* job growth between now and 2018."

The future Web will facilitate a more pervasive and intuitive user experience, providing content or services specific to the user's implied needs, suggests Richard Stanton.

most cases was not compiled for the purpose of matchmaking, making the effective organization of it a challenge that Stanton says is unique to the company. One aspect of this challenge, in particular, is replacing existing multi-select fields by using semantic data relationships from narrative fields.

According to Stanton, Bintro's goal is not only to demonstrate the use of Web 3.0 technologies today, but also to build an engine for powering other Web 3.0 apps in the future. "Web 3.0 is all about personalization," he says. "Instead of simply looking at the user as an eyeball, Web 3.0 aims to look at the user as an engaged personality with multiple facets from which the context of a user's statement can draw a better result."

Lassila points to this type of highly customized user experience as an ongoing challenge at Nokia. "This is good for the users, but I am not entirely convinced how sustainable this is, as the implementation part becomes more and more difficult," according to Lassila. Still, he says Semantic Web technologies hold great promise, particularly in situations in which users might require useful information from multiple data sources. "There are plenty of existing opportunities for clever data management," he says.

As for future research, Lassila says he is committed to working toward a "substantial convergence" of technologies, with computing machinery such as phones, PCs, and appliances connecting with communication systems to facilitate seamless interaction with family, friends, and colleagues, regardless of the different technologies involved. "It should not matter where the data comes from, where it resides, or what applications or systems create it," Lassila says. "What matters is how I want to use it."

Echoing this sentiment, Stanton suggests the future Web will facilitate a more pervasive and intuitive user experience, providing content or services specific to the user's implied needs. "I was a big fan of *The Jetsons* as a kid and I always loved how effortless their interaction with technology was," he says. "The Semantic Web puts us one step closer to such a reality."

For his part, RPI's Hendler predicts that in five years when Web 3.0 strategies have begun to mature, Web applications might still look a lot like they do today but will have much more data available to them, will have search-like capabilities far more sophisticated than current search engines, and will be able to exploit query context much more effectively. Hendler also predicts that much more of our access to the Web will be from mobile devices, with location and social context more readily available to applications that are given elevated privileges.

Still, like many researchers working in this area, Hendler is already looking beyond emerging Semantic Web strategies and related technologies that are now collectively called Web 3.0. "This stuff is new and exciting," he says. "But I look at it this way: I started playing with the Semantic Web back in the 1990s. As a researcher, I'm not content to sit around and exploit Web 3.0; my job is to help create Web 4.0." C

Further Reading

Berners-Lee, T., Hendler, J., and Lassila, O.
The Semantic Web. *Scientific American 284*, 5, May 2001.

Harris, D.
Web 2.0 Evolution into The Intelligent Web 3.0, Emereo Publishing, Brisbane, Australia, 2008.

Hendler, J.
Web 3.0 emerging. *Computer 42*, 1, January 2009.

Shadbolt, N., Berners-Lee, T., and Hall, W.
The Semantic Web revisited. *Intelligent Systems 21*, 3, May 2006.

Warren, P., Davies, J., and Brown, D.
The Semantic Web: from vision to reality. *ICT Futures: Delivering Pervasive, Real-time and Secure Services*, John Wiley & Sons, Hoboken, NJ, 2008.

Based in Los Angeles, **Kirk L. Kroeker** is a freelance editor and writer specializing in science and technology.

Networking

Bell Labs to Reduce Networks' Energy Usage

Could today's communications networks be 1,000 times more energy efficient? Bell Labs thinks so, and has launched a global consortium, Green Touch, which aims to make networks 1,000 times more energy efficient than they are today.

"A thousand-fold reduction is roughly equivalent to being able to power the world's communications networks, including the Internet, for three years using the same amount of energy that it currently takes to run them for a single day," Bell Labs said in a statement.

The thousand-fold efficiency target is based on Bell Labs research that indicates current information and communications technology (ICT) networks have the potential to be 10,000 times more energy efficient than they presently are. "A concerted effort to bring energy efficiency closer to these theoretical limits would not only shrink the estimated 2% of the world's carbon emissions ICT contributes directly, but also lower the 98% contributed by all the other sectors touched directly and indirectly by ICT," according to Bell Labs.

The Green Touch consortium will explore the fundamental properties of communication networks and technologies—optical, wireless, electronics, processing, routing, and architecture—and study their physical limits by applying established formulas such as Shannon's law.

For more information, visit greentouch.org.

Society | DOI:10.1145/1666420.1666429 Samuel Greengard

Tracking Garbage

Researchers are focusing on the so-called "removal chain" in an attempt to save landfill space, improve recycling rates, and trim the flow of toxic materials into the environment.

In a world where the movement of goods—everything from pallets of breakfast cereal to computer components—is tracked with precision, it's nothing short of remarkable that trash and recyclables are generally discarded without a thought. Worldwide, humans generate more than 2 billion tons of waste annually. In the U.S., each individual produces about 1.5 tons of solid waste per year. Unfortunately, no one knows exactly how all the waste flows, where it goes, and how it can be managed more effectively.

This situation may soon change, however. Researchers are now focusing on the so-called "removal chain" in an attempt to address a long-standing problem: how to save landfill space, improve recycling rates, and trim the flow of toxic materials into the environment. Using barcodes, passive and active radio frequency identification (RFID) tags, cellular transmitters, and other technologies, they're putting a high-tech spin on what has long been a low tech and mostly unmanageable problem.

It's certainly more than a throwaway idea. Trash-tracking technology provides a number of benefits, including the ability to follow individual items, components, and subcomponents through the disposal process to ensure that they are recycled or disposed of correctly; gauge how effectively curbside recycling programs work and use incentives to boost participation rates; and weigh trucks as they go to landfills to better understand loads and how to establish more efficient routes and service patterns.

"The study of what we could call the 'removal chain' is becoming as important as that of the supply chain," states Carlo Ratti, director of the SENSEable City Laboratory at the Massachusetts Institute of Technology (MIT). Ratti and a select group of researchers are among those tagging trash and exploring how society can deal with it more effectively. Notes Valerie Thomas, associate professor in the School of Public Policy at Georgia Institute of Technology, "Waste is a topic that society must address more effectively. We must find ways to reduce waste and make recycling easier and more streamlined."

In July 2009, MIT's TrashTrack team deployed 3,000 smart tags on waste objects in New York, Seattle, and London, facilitating the monitoring of the trash's path in real time.

PHOTOGRAPH BY KATE RAYNES. VISUALIZATION BY SENSEABLE CITY LAB. MIT

Trash Gets Smart

The idea of giving trash brains is ultimately about dollars, yen, euros, and good sense. At present, it's often next to impossible to assure that trash is routed to the best possible destination for disposal or recycling. "The problem with the current system is that there is little understanding or control of the waste stream. In many cases, trash and recycling materials don't wind up where they are supposed to go to," observes Lewis Girod, a research scientist at MIT who designed the tags for the SENSEable City Laboratory project.

That may soon change. The SENSEable City project, in place in New York and Seattle, aims to better understand the removal chain and boost recycling rates. A system called TrashTrack uses hundreds of small, smart, and location-aware tags as a first step toward the deployment of "smart-dust" networks of tiny locatable and addressable microelectromechanical systems. Researchers attach the tags to different types of trash in order to follow objects through a city's waste management system. This reveals the final journey of items in a series of real-time visualizations. MIT displays the information at the Seattle Public Library and the Architectural League of New York.

So far, researchers have tagged more than 3,000 pieces of Seattle and New York City garbage with electronic-tracking devices that use a GSM chipset, SIM card, and cellular radio contained within a 2-inch-long device. The units—chosen because of the low cost and ubiquity of GSM—rely on an algorithm to shut off when they haven't moved for a few

minutes, and a timer and motion sensor to wake them up when movement is detected. When tags come into contact with a new cell tower they send a status report via SMS. Researchers match the time stamps with the reports to create a movement map. The method of using cellular signals is accurate to about 100 meters, which is sufficient for tracking trash movement.

Over the short-term, the units have proved effective. However, because they draw from a 900-milliamp lithium ion battery, they do not provide a long-term solution to trash tracking. At most, they can last about six months. Another challenge, Girod says, is ensuring that the antennas attached to the individual pieces of trash have exposure to the sky so that they can transmit continuous signals during the transport process. In some instances, other objects, vehicles, or facilities have obscured the units. The transmitters are enclosed in a small fiberglass shell to help them survive movement and possible compacting. Girod says the use of 3G GSM will provide better signal accuracy and dependability.

MIT isn't the only group to experiment with trash tagging. Georgia Institute of Technology's Thomas has examined tagging technology as well. She has focused on using conventional barcodes and RFID tags to track items as they move through the waste stream. The primary value, she says, is for managing items like batteries, toys, electronics, office equipment, shop equipment, household tools, garden equipment, and even clothes. "Many of these items can be recycled but they often aren't," says Thomas. "Some of them—including household chemicals, light bulbs, and fixtures—may contain toxic substances that could be more easily tracked and removed."

She advocates placing barcodes and more advanced optical barcode labels on items, and using passive and active RFID tags for situations where automated scanning and tracking makes sense. "Right now, one of the biggest problems is the way items are packaged," Thomas notes. "There is often a barcode on the package when something is sold, but the actual appliance or device—and its subcomponents—are not identified." Claudia Binder, a professor at the University of Zürich's Institute of Science, Innovation, and Sustainability Research, has studied the use of RFID in trash tagging and believes it is feasible and could play an important role in changing behavior and improving environmental awareness. RFID, she says, speeds data collection and eliminates line-of-sight issues. It would almost certainly "lead to an improvement in the current recycling rate," she says.

Nevertheless, tagging garbage presents a few obstacles. For one, there's the cost of adding labels, tags, and readers to the removal chain—something that could boost per-item costs from a few cents to a few dollars. Tags themselves would have to be recycled, and privacy issues could enter the picture, Binder says. Without adequate protection, someone could glean details about a person's life and consumption habits. Finally, Binder worries that too much automation could have the unintended consequence of decreasing environmental awareness and shifting responsibility away from recycling.

Waste Not, Want Not

Despite the challenges, the idea of using technology to track and manage trash is gaining momentum. In Aspropyrgos, Greece, a suburb of Athens, city officials implemented a three-month pilot study in 2007. Altogether, 15 of the 2,500 city-supplied garbage bins used by residents and businesses were equipped with an RFID tag mounted near the base of the bin. Each of the city's 15 garbage-collection trucks was equipped with an RFID reader and when workers emptied any of the tagged bins into the truck, the antenna picked up a unique ID encoded to the bin's tag. The system allowed the town's sanitation department to optimize routes and manage vehicles more efficiently. It also helped the city gauge the productivity of crews in the field.

In Philadelphia, an RFID-based recycling system called RecycleBank (developed by Texas Instruments) was piloted in 2006. A high-tech bin measures the volume of recyclables contained within it and when a truck picks up the items, it transmits the data to an onboard computer. Households receive cash awards based on the amount of plastic, glass, and other materials they contribute. Recycling participation rates among the 2,500 residents who initially subscribed to the program rose from 25% to 90%. Moreover, the average household increased the volume of recyclables from less than 5% to more than 50%.

An effective removal-chain system would eventually create a more efficient disposal system and slash landfill requirements, Thomas says. It would also create new economies and opportunities. Ultimately, Thomas would like to see a system where items that cannot be recycled—everything from banana peels to soiled napkins—can be composted and combusted, with the latter method producing power. MIT's Girod believes a better understanding of waste would lead to important changes in public behavior and public policy.

In fact, governments are beginning to take notice, Girod says. In the United Kingdom, the Department for Environment, Food and Rural Affairs is studying trash tagging in order to better understand waste flow and how to trim refuse collection costs, improve recycling rates, and lessen the environmental impact of garbage, including hazardous waste. In the U.S., the Environmental Protection Agency has indicated interest in boosting compliance, and tagging would likely create a viable framework for managing consumption from purchase to landfill.

"There's no question that tracking trash has economic and social benefits," concludes Thomas. "It will likely play an important role in the future. We must become more efficient in the way we dispose of waste." C

Further Reading

MIT Trash | Track Web site
http://senseable.mit.edu/trashtrack/

Saar, S. and Thomas, V.
Toward trash that thinks. *Journal of Industrial Ecology 6*, 2, January 2002.

Smart trash cans (video and text)
ScienceDaily, October 1, 2006.
http://www.sciencedaily.com/videos/2006/1001-smart_trash_cans.htm

Binder, C.R., Quirici, R., Domnitcheva, S. and Stäubli, B.
Smart labels for waste and resource management. *Journal of Industrial Ecology 12*, 2, April 2008.

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

Milestones | DOI:10.1145/1666420.1666430

Katayanagi Prizes and Other CS Awards

DONALD E. KNUTH, Jon Kleinberg, Andrew Herbert, and other members of the computer science community were recently honored for their innovative research and service.

Katayanagi Prizes in Computer Science

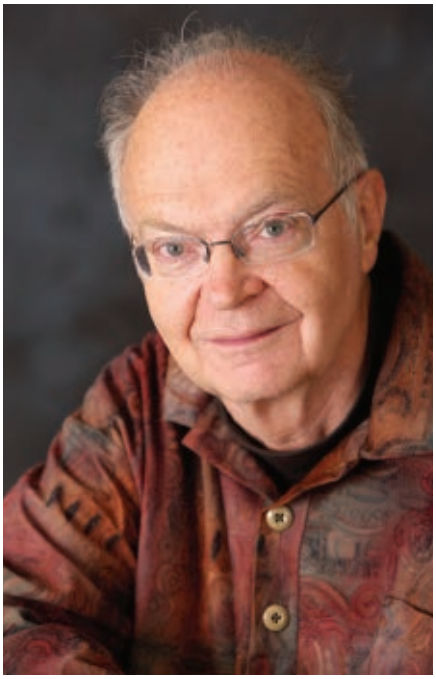
Donald E. Knuth, who has made fundamental contributions in theoretical computer science and is the author of the seminal multi-volume *The Art of Computer Programming*, and Jon Kleinberg, a computer scientist whose work explores the interface between networks and information, were awarded the Katayanagi Prizes in Computer Science.

Knuth, an emeritus professor at Stanford University and recipient of the 1974 ACM A.M. Turing Award, received the 2009 Katayanagi Prize for Research Excellence, which recognizes an established researcher with a record of outstanding, sustained achievement. Kleinberg, the Tisch University Professor of Computer Science at Cornell University, received the 2009 Katayanagi Emerging Leadership Prize, which honors a researcher who demonstrates the promise of becoming a leader in the field.

The prizes are presented annually by Carnegie Mellon University in cooperation with the Tokyo University of Technology (TUT). The prizes are endowed with a gift from Japanese entrepreneur and education advocate Koh Katayanagi, who founded TUT and several other technical institutions in Japan.

Order of the British Empire

Microsoft Research Cambridge Managing Director Andrew Herbert was appointed Officer of the Order of the British Empire (OBE) by Queen Elizabeth II for services to computer science. The appointment was announced by Buckingham Palace as part of the 2010 New Year Honours list. A Microsoft Distinguished Engineer, Herbert has worked in the computer science field for 35 years, conducting research into computer networking, operating systems, and distributed computing. He is the fifth Microsoft employee to be honored by the OBE, in addition to Bill Gates, Tony Hoare, Tony Hey, and Roger Needham.



Donald E. Knuth, winner of the Katayanagi Prize for Research Excellence.

PHOTOGRAPH BY HECTOR GARCIA-MOLINA

AAAS Fellows

The American Association for the Advancement of Science recognized 14 individuals as Fellows, in the Section on Information, Computing, and Communication, for their contributions to science and technology. They are: Marc Auslander, IBM Watson Research Center; Richard G. Baraniuk, Rice University; Alok Choudhary, Northwestern University; Narsingh Deo, University of Central Florida; James A. Gosling, Sun Microsystems; Anthony J.G. Hey, Microsoft Corporation; Eric Horvitz, Microsoft Corporation; Henry C. Kelly, U.S. Department of Energy; Thomas F. Knight, Massachusetts Institute of Technology; David B. Lomet, Microsoft Corporation; Keshav K. Pingali, University of Texas, Austin; Sanguthevar Rajasekaran, University of Connecticut; Jeffrey S. Vitter, Texas A&M University; and Ouri Wolfson, University of Illinois, Chicago.

ISSA Distinguished Fellows

Information Systems Security Association honored 22 individuals as Distinguished Fellows, the association's highest tribute. They are: Mary Ann Davidson, Dorothy Denning, Donald Evans, Susan Hansche, Steve Hunt, Sandra Lambert, Richard Mosher, William Hugh Murray, Lynn McNulty, Alan Paller, George Proeller, Marcus Ranum, Ron Ross, Howard A. Schmidt, Bruce Schneier, Eugene Schultz, Sanford Sherizen, Eugene Spafford, Harold Tipton, William Tompkins, Roy Wilkinson, and Ira Winkler.

IEEE Awards

The Institute of Electrical and Electronics Engineers (IEEE) Computer Society honored Michael T. Heath, the University of Illinois at Urbana-Champaign's Fulton Watson Copp Chair in computer science, with the 2009 Taylor L. Booth Education Award for his "contributions to computational science and engineering education, curriculum, and scholarship."

IEEE Computer Society also honored Judy Robertson, senior lecturer in computer science at Heriot-Watt University (and a blogger for *Communications*' Web site), with the 2009 Computer Science and Engineering Undergrad Teaching Award for her "outstanding contributions to the undergraduate education through teaching and the innovative use of pioneering technologies in teaching."

DOI:10.1145/1666420.1666431

Tim Chang

Economic and Business Dimensions

Gaming Will Save Us All

How gaming, as the first media market to successfully transition toward media-as-a-service, is an exemplar for a similar evolutionary transition of content and entertainment.

I often proclaim at digital media and gaming conferences that gaming (2.0) will save us all. By this, I don't mean that we will spend our leisure hours reaching level 68 Dark Elf Druids in *World of Warcraft*. I mean that well-proven, hybrid revenue models from the cutting edge of Gaming 2.0 will revive traditional media industries, many of which have been disrupted by digital formats and irreversibly fragmented by the Internet.

Gaming 1.0 is the traditional packaged-goods, retail-based model we are all familiar with. Historically it will be remembered as the domain of nerdy, young males playing old-school downloadables and ad-based online products. The hits-based business model of Gaming 1.0 was as unpredictable as the hits themselves. Gaming startups had difficulty scaling up to large, standalone businesses. The success of Electronic Arts and a few others was overshadowed by countless business failures. Gaming 1.0 companies spent fortunes to make products that might or might not make money. As with the movie business, Gaming 1.0 companies increasingly bet the farm on "tentpole" titles, often sequels of prior successes. They did not invest in newer and riskier products.

In contrast, Gaming 2.0 makes games frictionless, ubiquitous, social, and service-oriented. Gaming 2.0 provides critical clues and guidance toward new, sustainable business models that could benefit social media, content, and e-commerce businesses in the future. Gaming 2.0 evens the playing field for game startups. It is not just the application of new computing and Internet technologies to old gaming paradigms; it combines new technologies, new designs, and new business paradigms. It is fueled by major changes in consumer behavior:

▸ **Rise of the Digital Natives:** Consumers under the age of 30 have grown up with the Internet, social networks, instant messaging, email, search, blogs, P2P downloading, and podcasts. These users spend more time on Web 2.0 sites like YouTube and social networks like Facebook and Twitter. They eschew broadcast TV for online video services like Hulu. Digital natives favor online, digital formats instead of visiting retail outlets to discover and purchase their media. They are adept at seeking out new offerings, and they share their findings virally.

▸ **Irreversible Fragmentation and Short Attention Spans:** The Long Tail theory suggests the Web allows every user to find offerings that suit his or her particular tastes. Such wide choice makes it increasingly difficult for a publisher or media company to reach mass audiences effectively. This is borne out by dying formats such as broadcast TV, radio, and newspapers. Thousands of niche offerings replace a few massively popular hits that everyone likes. On top of this there is a very short attention span from users faced with unprecedented options and the ability to surf instantly on to the next site or link with

a single click if faced with something that isn't instantly appealing.

▸ **New, Open, and Lightweight Platforms:** The most exciting new formats for interactive experiences include the iPhone (for which there are now more than 100,000 applications, of which more than one-third are games) and other smartphone platforms, social networks such as Facebook and Myspace with open application support, and increasingly powerful Internet browsers capable of rich-media experiences, streamed fully in-browser, without the need for heavy downloads or additional software. These formats are widely available and have experienced wide adoption, attaining huge active audience bases worldwide.

In light of these disruptive changes, we've entered what many consider to be a Renaissance Era of Indie Gaming, in which small upstart teams have just as much a chance to launch a profitable title directly into the market as the large traditional publishers. In many cases the startups are innovating much faster than the incumbents, who continue to fight the Gaming 1.0 battle under pressure from a slowing retail-based model.

More specifically, the following innovative drivers define Gaming 2.0:

Ubiquitous Gaming. The most exciting aspect of Gaming 2.0 is the rise of games on new, popular platforms with mass audiences such as iPhone and Facebook. Through these gamemakers can reach whole new audiences willing to try out games, but who would never self-identify as the stereotypical "gamers" who only make up 10% of the population. At least two-thirds of iPhone and iPod Touch owners have downloaded and played a game. Most did not expressly purchase the device to play games, as opposed to those who purchase of a dedicated game console, but a mass market of non-core gamers is now investing significant amounts of time and money in lightweight games, expressly built for these non-dedicated and ubiquitous platforms.

Approaching Frictionless Distribution. Gaming 2.0 is about bringing games to the broader market of non-core gamer consumers, so the games themselves need to be as easy as possible to try out and play. Typical practices from the Gaming 1.0 world, such as asking prospective players to first download heavy client software, register to play, or purchase the game upfront, represent friction points for user adoption and scare off many consumers. The gaming market is giving way to browser-based, thin-apps and free-to-play frictionless models, aimed at getting the consumer to try out a game quickly and get hooked on the experience, eventually resulting in deeper engagement, viral sharing, and monetization. Frictionless gaming also emphasizes direct-to-consumer publishing, without the need for retail distribution or a packaged game product. Many experiments in Gaming 2.0 involve open platforms such as Facebook, MySpace, iPhone, and Xbox Live Community Games, which enable startup developers to release games into the market unfettered by traditional gatekeepers or distributors.

ILLUSTRATION BY GLUEKIT

Social as a Means Toward Distribution. "Social networking" is not a market: it's a kind of functionality that will be woven into all offerings as a mechanism to enhance distribution, marketing/promotion, and self-expression/engagement. Similarly, "mobile" is not a market, but a logical extension of the cloud media model, an increasing overlay of the social graph across all content and media. This has already begun to reduce the anonymity of the Web and increase the accountability and quality of conversation in and around content/product/media. Potentially, this could lead to multi-region, cross-pollination in interactions. Social gaming is interesting because both sides (social media mavericks, traditional games folks) are learning from each other. Social can grow audience bases virally, while gaming can retain and engage these audiences long after the initial viral buzz wears off. Today's social games, such as Zynga's *MafiaWars*, have proved that you don't even need a real game engine or fancy graphics to get large audiences playing and spending money. These simple x-Wars social games are just the beginning. Soon we will see improvement in formats, quality of production, and interactive storytelling—the start of "premium social gaming."

Designing Compelling Content for "Snack Gaming" and Voluntary Repetition. Gamemakers are learning to optimize the user experience for "snack gaming" (many frequent sessions of only a few minutes of play), as opposed to deep engagement with multi-hour gameplay sessions. Most consumers are time-starved and do not want to invest significant amounts of time per play like hardcore gamers. Compelling stories and content that induces "compulsion loops" drive users to keep coming back for more participation. The stories may be episodic and involve user-modifiable game content. Users come back because they like the experience and get hooked on one or more compulsion loops. These users do not need constant "app and social newsfeed spam" to get them to reengage. Such alerts are good for initial awareness building, but the core compulsion loop quickly takes over.

Many aspects of Gaming 2.0 can be transported to music, broadcast TV, print journalism/magazines, and packaged media in general.

The art of the "meta-game" will be the most important long-run design skill for Games-as-a-Service (GaaS): when each piece of content or activity ultimately becomes a "mini-game," it is important to design the meta-game wrapper around everything to encourage users to level-up, collect, share, buy/sell/trade, explore, and try again.

Games as a Service. Games packaged as "fire and forget" releases are quickly becoming an obsolete. Online pioneers such as a Blizzard (*World of Warcraft*) and Nexon (*Maple Story*) have proved that GaaS and Cloud Gaming are viable new paradigms. Content creators must rethink how they design their offerings, moving away from discrete offerings sold and handed off to players, and toward establishing ongoing relationships with their users.

Games are increasingly "living" on Internet servers as ongoing experiences in which players touch each other frequently across a multitude of devices (Web browsers, consoles, iPhones, and even within social networks). This shifts expenses, planning, and spending priorities. Only half or less of budgets will be spent on "launch," with increasing amounts used to "operate" the ongoing service. Activities such as community management, expansion packs/dynamic content updates, microtransactions processing, virtual goods refreshes, and related activities become more important. This is an early case study for the shift of media overall toward Media-as-a-Service (MaaS). Imagine subscribing to all-you-can-eat on-demand music services instead of buying individual songs or albums, or think of the emerging online Netflix model. Getting content into the hands of the audience is just the first step; the real trick is keeping the consumer engaged over the long-term, continually monetizing the relationship and cross-promoting other offerings.

Reinventing the Business Model for Media. MaaS as inspired by the gaming industry is the new Holy Grail for media. The reinvigoration of the music, TV, movie, and print media industries will come from adaptations of this model. The model blends revenue streams, including free-to-play, microtransactions (around virtual goods and virtual currencies), and premium membership and subscriptions. A healthy model shows what I call the "85/15/2" pattern: the majority (85%) of participants plays for free, and do not engage in microtransactions or subscriptions. They can be lightly monetized via ads, but they contribute indirect value by enriching the game world and experience for other players through their participation. A smaller fraction of participants (10%–15%) pays small amounts for microtransactions. A very small fraction of participants (1%–3%) pays for premium services or subscriptions. Subscribers might engage in microtransactions for rare items, collectibles, vanity goods, and so forth. A parallel scenario in the music industry would have ad-based "free to play" for a limited on-demand streaming music experience; per-song and per-item microtransaction purchases; and all-you-can-eat unlimited subscription services with additional perks and benefits.

Many aspects of Gaming 2.0 can be transported to music, broadcast TV, print journalism/magazines, and packaged media in general. Gaming is the first media market to shift successfully toward MaaS (Media-as-a-Service), and is a terrific poster child for how content and entertainment might transition toward XaaS (Everything-as-Service). C

Tim Chang (tchang@nvp.com) is Principal at Norwest Venture Partners, which has supported over 400 companies in its 45-year history. This column is derived from his keynote speech from the Casual Connect 2009 conference.

DOI:10.1145/1666420.1666432

Pamela Samuelson

Legally Speaking
Only Technological Processes Are Patentable

The U.S. Supreme Court will narrow the universe of process innovations that can be patented to those that are "technological," but what will that mean for software?

On November 9, 2009, the U.S. Supreme Court heard oral argument in the *Bilski v. Kappos* case. The question is whether a method for hedging risks of price fluctuations of commodities is eligible for patent protection.

My most recent *Communications* Legally Speaking column, "Are Business Methods Patentable?" (November 2009), suggested the Court's ruling in *Bilski* would have implications for the patentability of computer programs. After attending the oral argument in the case, I am now less sure of that.

One thing I am sure of, though, is that Bilski is not going to get his patent. The Court made mincemeat out of Bilski's main arguments in favor of the patentability of his method. The Justices peppered him with questions and made comments indicating that they thought his arguments were preposterous.

Hearing the oral argument also convinced me that the Court is unlikely to proclaim that business methods, as such, are ineligible for patenting. The Court instead seems likely to rule that Bilski's method is unpatentable because it is a nontechnological process.

To implement this standard, the Court is likely to adopt a "machine or transformation" test so that the Patent and and Trademark Office (PTO) and the courts can distinguish between technological and nontechnological processes. Under this test, Bilski's method is unpatentable because it is neither tied to a specific machine, nor does it transform anything from one state to another.

Pamela Samuelson holding the Bilski brief in front of the U.S. Supreme Court building.

PHOTOGRAPH COURTESY OF PAMELA SAMUELSON

The main reason *Bilski* is unlikely to address software patent issues is that dozens of software companies and organizations filed amicus curiae (friend of the court) briefs explaining that a broad patent subject matter ruling in *Bilski* could sweep away patents in this field. (Some amici wanted software patents to be swept away, while others sought to preserve software patents.) The Court will likely leave questions about the patentability of software innovations to future cases.

Alphabets, Horse Whispering, and Speed Dating

Most Justices came to the oral argument with their favorite examples of innovations they thought were unpatentable and tested them out on Bilski's lawyer, Michael Jakes. Justices Kennedy and Roberts, for instance, quizzed Jakes about whether a new alphabet could be patented. Dutifully sticking to his script, Jakes said yes insofar as it was a practical application of knowledge that could be expressed in a series of steps.

Under Bilski's theory of patent subject matter, Justice Scalia suggested that innovations in horse-training techniques, such as horse whispering, would be patentable. Yet, no such patents have issued for them. Scalia asked Jakes to explain why. When Jakes answered that the U.S. economy in the 19th century was based on industrial processes, Scalia derisively commented that the economy back then was based more on horses.

Scalia also asked Jakes if an improved method for winning friends and influencing people was patent-eligible, conveying by the tone of his voice that he thought the very idea was absurd.

The patentability of speed-dating methods was raised by Justice Sotomayor, who worried that without some sort of technology limitation patents would extend too far and impose too many costs on society.

That Bilski's theory would also allow patents on estate planning, tax avoidance, and jury selection methods was of concern to Justice Ginsburg who plainly regarded these methods as beyond the patent pale.

Justice Breyer asked Jakes if a professor could patent an improved method of teaching antitrust law. After Jakes affirmed this, Breyer asked him to suppose the Court was not willing to go that far; did Jakes have anything to offer as an alternative formulation of patent subject matter? Jakes did not.

The Court will likely leave questions about the patentability of software innovations to future cases.

What Test to Use?

That Bilski will lose his appeal is certain. But the Justices were plainly struggling during the oral argument about what test should be used to distinguish between patentable and unpatentable processes.

The test will certainly not be the patent subject matter rule that the Court of Appeals for the Federal Circuit (CAFC) used between 1998 and 2008. It focused on whether a claimed method produced a "useful, concrete, and tangible result."

In the decade after the CAFC announced this test, the PTO was flooded with applications for patents on a wide range of methods in many fields of human endeavor, including sports moves, business methods, arbitration procedures, charitable giving techniques, and dating methods.

After the Supreme Court in 2006 expressed dissatisfaction with the CAFC's views of patent subject matter (see my July 2008 column "Revisiting Patentable Subject Matter"), the CAFC decided to revisit patent subject matter. It heard Bilski's appeal en banc (with all 12 judges on the court, not just the usual three-judge panel) and articulated the machine-or-transformation test mentioned previously, under which Bilski's method was unpatentable.

As formulated by the CAFC, the machine-or-transformation test has been criticized for being too formalistic, failing to articulate a normative or policy-based grounding, and too easily subverted by a simple mention of technology (for example, a computer) in the claims.

Yet, the PTO has defended this test as practicable for conducting examinations. In its brief to the Court, the Solicitor General explained why the PTO believed this test was consistent with the Court's prior rulings and why it would be workable in making subject matter determinations.

During the oral argument, three other bases for resolving the patent subject matter question posed by Bilski's application came up.

Justice Alito wondered whether the Court should reject Bilski's claims on the ground that they were too abstract to warrant a patent. Malcolm Stewart, the government lawyer who defended the PTO's rejection of Bilski's claims, said such a ruling would undermine the "limited clarity" that the machine-or-transformation test had provided and would leave unresolved the question as to whether nontechnological processes, such as antitrust teaching methods, were or were not patentable.

Justice Sotomayor asked whether the Court should resolve the case by ruling that business methods were unpatentable. Stewart argued against this because the PTO thought that some technological implementations of business methods might qualify for patents.

Justice Ginsburg was attracted to the idea of saying that technological processes are patentable, but nontechnological processes aren't. Stewart characterized the machine-or-transformation" test as a "shorthand version" of that standard.

As the oral argument proceeded, the Justices became more comfortable with the machine-or-transformation test. Yet, they were plainly concerned about the risk that adoption of this test might foreclose patentability as to a new technology that did not satisfy this test.

To address this concern, Stewart recommended that the Court "acknowledge that there has never been a case up to this point that didn't involve a machine or transformation," but it "could leave open the possibility that some new and as yet unforeseen technology could require the creation of an exception." This seemed to satisfy the Court's concerns.

Difficult Questions Ahead Involving Computers

Bilski is an easy case under the machine-or-transformation test because Bilski didn't mention any technology in his application: no telephone, no fax machine, no computer.

Several Justices were skeptical of the view that merely mentioning a conventional technology in a patent claim could suffice to convert an unpatentable process into a patentable one. A method of calculating historical averages of prices, for instance, should not

The patentability of software-related inventions has been hotly debated since the mid-1960s.

become patentable just because the claim mentions the use of a calculator in carrying out the method.

Justice Roberts stated his view that "tangential and insignificant" uses of machines in a claimed process should not render the process patentable. Stewart agreed that the use of a conventional piece of technology for its conventional functionality should not change the patent calculus for claims mentioning them.

A much more difficult set of questions arises, however, with respect to computers. Arguing for the PTO, Stewart asserted that a programmed computer to carry out a claimed method would satisfy the machine-or-transformation test. Several Justices did not find this argument persuasive.

Justice Breyer, for instance, expressed concern that if the Court accepted this view, then business methods such as Bilski's could be easily become patentable by mentioning use of computers to carry out the methods. This would undermine the Court's clear intention that such methods not be patentable.

Justice Stevens contested the view that a programmed computer was a new machine, given that the only new thing about the computer was a software process being run on it.

Also unclear is what kinds of transformations will satisfy the test. Back in 1972, the Court called into question the patentability of processes that transform data in *Gottschalk v. Benson*. That case upheld the PTO's denial of a patent for an algorithm for converting binary coded decimals into pure binary form. The only software-related process that the Court has ever deemed patentable—and that only by a 5-4 decision—was *Diamond v. Diehr* in 1981. *Diehr* involved a rubber-curing process that transformed matter from one physical state to another, which utilized a computer program in conjunction with it.

By the end of the oral argument, Stewart seemed to have convinced the Court that *Bilski* was not the appropriate vehicle for addressing the complex issues that computers raise. They will likely be left for another day.

Conclusion

Normally I would wait until the Court published its decision before writing a "Legally Speaking" column about it and its implications for computing professionals. *Bilski* was a rare instance in which the oral argument illuminated the Court's views on the merits and clearly signaled the direction of the Court's thinking about the reasoning it would use to justify its ruling.

(I should confess, however, that one reason I decided to write about *Bilski* now is because it was a case in which I submitted an amicus brief in support of the PTO, and it was the first oral argument before the U.S. Supreme Court I ever attended. It was such a thrill.)

The *Bilski* ruling will likely be unanimous. The only question is whether there will be one opinion or two or three. In some recent intellectual property cases, the unanimous opinion for the Court has been fairly short and straightforward, supplemented by concurring opinions that express some Justices' views about issues not addressed in the main opinion for the Court.

It would not surprise me if the Justices did a little (unpatentable) horse trading in their post-argument conference on *Bilski* under which they agreed to issue only one opinion in this case and to take a software-related patent subject matter case when the opportunity arose, as it almost certainly will very soon.

The patentability of software-related inventions has been hotly debated since the mid-1960s. There is still no resolution in sight. But the Court is focused on software-related patent issues again. So we can expect some significant developments in the next two or three years. C

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

Calendar of Events

March 15–19
Eighth International Conference on Aspect-Oriented Software Development
Rennes and Saint Malo France,
Contact: Jean-Marc Jezequel,
Phone: 33299847192,
Email: jeqzequel@irisa.fr

March 16–18
3rd International Conference on Simulation Tools and Techniques
Malaga, Spain,
Contact: Luiz Felipe Perrone,
Phone: 570-577-1687,
Email: perrone@bucknell.edu

March 18–19
ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems
TBA, CA,
Sponsored: SIGDA,
Contact: Peng Li,
Email: pli@tamu.edu

March 22–24
Eye Tracking Research and Applications
Austin, TX,
Sponsored: SIGCHI and SIGGRAPH,
Contact: Carlos Hitoshi Morimoto,
Phone: 55-11-3091-6499,
Email: chmorimoto@gmail.com

March 22–26
The 2010 ACM Symposium on Applied Computing
Sierre, Switzerland,
Sponsored: SIGAPP,
Contact: Sung Y. Shin,
Phone: 605-688-6235,
Email: sung.shin@sdstate.edu

March 26–27
Consortium for Computing Sciences in Colleges (CCSC) Midsouth
Searcy, AR,
Contact: Dr William M Mitchell,
Phone: 317-392-3038,
Email: willmitchell@lightbound.com

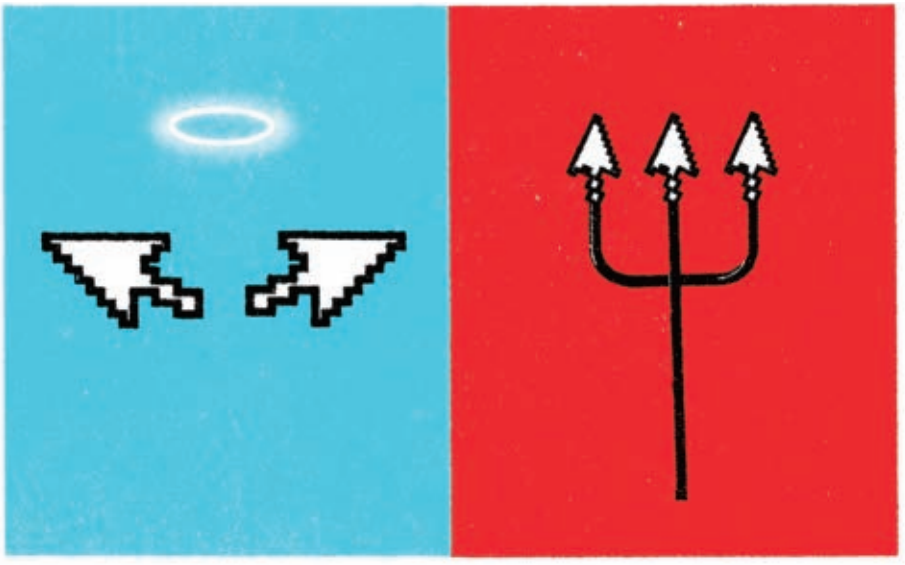
March 29–31
International Conference on Multimedia Information Retrieval
Philadelphia, PA,
Sponsored: SIGMM,
Contact: James Ze Wang,
Phone: 814-865-7889,
Email: jwang@ist.psu.edu

DOI:10.1145/1666420.1666433

Rachelle Hollander

Computing Ethics
The Ethics Beat

Surveying the increasing variety and nature of ethical challenges encountered by computing researchers and practitioners.



I HAVE ASSUMED responsibility for the ethics column in *Communications* Viewpoints section. The Computing Ethics column will appear occasionally and provide an editorial focus promoting understanding and resolution of ethical issues of concern to people in the computing profession. This inaugural column has two goals: to provide information about the center I run and to briefly highlight the work of two people who have contributed to ethics in the areas of computer science and engineering.

I direct the Center for Engineering, Ethics, and Society (CEES, http://www.nae.edu/ethicscenter) at the U.S. National Academy of Engineering (NAE). The NAE is part of The National Academies, a federally chartered membership organization that advises the U.S. on science, engineering, and medicine. The NAE launched CEES in 2007 with support from member Harry E. Bovay, Jr.; the Bovay grant provides core funding through 2011. CEES examines and helps to resolve societal and ethical issues through workshops, conferences, research, and education.

CEES manages the Online Ethics Center (OEC) at the NAE (www.onlineethics.org). OEC provides professionals and students in science and engineering with resources for understanding and addressing ethically significant problems that arise in their work. It promotes learning and advances the understanding of responsible research and practice. CEES is currently revamping the OEC with support from the U.S. National Science Foundation (NSF) that began in May 2009. Site improvements will provide assistance to principal investigators and academic institutions regarding new requirements for ethics instruction in the American COMPETES Act of 2007. The People section explains how you can contribute, and you can use the Ask-Us link in that section to ask questions.

Workshops

The NSF recently provided support for CEES to hold a workshop on Ethics Education in Scientific and Engineering Research: What's Been Learned? What Should Be Done? (See the report by that title from National Academies Press, 2009; http://books.nap.edu/catalog.php?record_id=12695). CEES worked with its advisory group and the National Research Council's Division on Policy and Global Affairs and the Academies-wide Committee on Science, Engineering, and Public Policy to develop the project.

At the meeting, participants articulated the following assumptions. Competitive and complex research environments pose increasing ethical challenges for research scientists and engineers. Interdisciplinary and international participation require crossing of cultural boundaries, and the close coupling of commerce and academia can create difficulty in recognizing the right thing to do. Gaps remain in ethics education, and it is difficult to measure the effectiveness of existing programs.

Charles Huff, Department of Psychology, St. Olaf College, has a longstanding research interest in identifying and evaluating ethical professional behavior. At the ethics education meeting he reported results of research involving numerous collaborators and sources of support. Huff's team used

ILLUSTRATION BY YAREK WASZUL

interviews and documentary materials to study two types of morally exemplary individuals in computing: those oriented toward craft (for example, computer accessibility for disabled users); and those oriented toward reform (for example, computing and privacy). These types represent different moral ecologies, which are environments in which individuals can develop ethically exemplary careers. Characteristics in a "model" of ethical performance over time include "moral ecologies, individual personality, relevant skills and knowledge, and the integration of morality into the individual self."

By understanding such complexities it is possible to assess the limitations in approaches to ethics education that focus only on individual decision points. Training in the skills and knowledge necessary to address particular ethical issues in research provides important guidance for analysis of particular situations, but it cannot inoculate individuals against questionable practices. A performance approach requires the evaluation of professional ethical behavior over the course of a career, and encourages an ethics perspective that goes beyond compliance toward the development of ethical ideals. For more information see http://www.stolaf.edu/people/huff/.

Ideas emerging from the workshop include:

- *Context*: Academic institutions should show they have established wide-ranging programs to stimulate and reward ethically appropriate behavior.
- *Learning*: Student participation should be mandatory and a repository of information about best practices should be created with a plan for dissemination of these materials to colleges and universities.
- *Criteria for programs and activities*: Successful programs involve research faculty using case studies and interactive formats supplemented with appropriate online materials.
- *Interactivity*: Students have a facility for accessible and interactive online resources. Ethics-focused instructional materials must reflect this.
- *Mentoring*: Science and engineering faculty and faculty with ethics education responsibilities should work together on mentoring postdoctoral fellows and graduate students at the dissertation level.
- *Evaluation*: Appropriate agencies should fund a workshop to develop evaluation criteria and measures for ethics education in science and engineering curricula.
- *Social responsibility and responsible conduct of research*: Support should be given to programs that creatively teach ethics and the social responsibilities of science and engineering, as well as the responsible conduct of research.

CEES is planning panels at several professional society meetings on this topic (see www.nae.edu/ethicscenter for upcoming events). Copies of the workshop report will be available at the meetings and free online at the Web site of the National Academies Press.

Another important topic examined by CEES is the relationship between engineering and social and environmental justice, and sustainability. Engineers sometimes get caught in conflicts that arise between different positive goals; for instance, when humanitarian efforts reinforce status inequalities or environmental degradation. In 2008, with partial support from NSF, CEES and the Association for Practical and Professional Ethics (APPE) sponsored a workshop titled "Engineering, Social Justice, and Sustainable Community Development." This workshop brought together engineers and scholars from Science and Technology Studies (STS) to consider improvements in engineering ethics, engineering practice, and engineering education. Engineering and social justice were hotly contested at the meeting, while engineering and humanitarianism, engineering and social responsibility, and engineering and environmental justice were less controversial.

Kevin Passino of the Department of Electrical Engineering at The Ohio State University participated in this workshop. He argued that educating engineers who take on volunteer work is a responsibility for engineering educators, and that fulfilling that responsibility requires the following:

- Putting more emphasis on ethics and professionalism in the curriculum;
- Encouraging hands-on volunteerism via student organizations; and
- Promoting service learning through community-oriented design projects.

Infrastructure Development

Developing the academic infrastructures that can encourage and support engineering volunteerism is a significant challenge. Passino noted that the definition of a profession has always included public service. Applied to the engineering disciplines, this definition implies that some portion of the engineering community must focus on serving society. Not every engineer must satisfy this criterion, but the profession as a whole must.

Passano provided examples of class assignments for teaching ethics and professionalism in the design of projects that meet community design constraints or address global issues, and research papers on subjects such as assessment of corporate citizenship programs and engineering volunteerism projects, evaluating codes of ethics, and so on.

To accomplish such goals, Passino argued for an infrastructure that goes beyond academia to involve professional organizations, government, and industry. He discussed as an example ECOS (Engineers for Community Service), a student-run organization at The Ohio State University that links students with sponsors of local and international service projects that promote professionalism.

For more about the ECOS-sponsored activities, see www.ecos.osu.edu for project descriptions; for more about his activities, see www.ece.osu.edu/~passino/.

Participants in the 2008 workshop on engineering and social and environmental justice, and sustainability agreed the discussion should continue at the 2010 APPE Annual Meeting, through a mini-conference titled "Engineering Towards a More Just and Sustainable World." Those interested in attending can learn more by checking the CEES or APPE Web sites.

I intend to explore ethics from many perspectives in future installments of this column and encourage and welcome any suggestions readers wish to provide. C

Rachelle Hollander (rhollander@nae.edu) is the director of the Center for Engineering, Ethics, and Society at the U.S. National Academy of Engineering in Washington, D.C.

DOI:10.1145/1666420.1666434

Peter J. Denning, Fernando Flores, and Peter Luzmore

The Profession of IT
Orchestrating Coordination in Pluralistic Networks

Learning to build virtual teams of people of diverse backgrounds is an urgent challenge.

Long the bane of organizations and teams, coordination breakdowns can be expensive, wasteful, mission killing, and sometimes life threatening. They manifest as miscommunication, misunderstandings, ill-timed actions, wasted motion and resources, and performance-killing bad moods. A plethora of coordination technologies seeks to overcome these problems and enable virtual teams, but coordination breakdowns have become more common and more severe in virtual teams. Exquisite coordination, which separates high performance teams from the rest, is an ever more elusive goal.

The core of the challenge is that the team members are drawn from pluralistic networks—people from different countries, cultures, backgrounds, worldviews, and practices. This diversity of value sets makes coordination all the more difficult.

Recent disasters have made the pluralism issue publicly visible. Despite all the good they did, the groups gathered for humanitarian assistance encountered systemic inabilities of government and non-government organizations to coordinate well, leading to delayed responses, wasted resources, and additional lost lives. Examples appeared during the 9/11 attack in New York City, the 2004 tsunami in the Indian Ocean, and the 2005 Hurricane Katrina in the U.S.[2]

Disaster relief teams have an additional problem: they are often under overwhelming stress. The tendency of teams to move toward dysfunction under stress regularly deepens disasters, loses wars, and sinks companies. Pluralistic worldviews exacerbate the stress because they add obstacles to coordination when there is no time to deal with them.

Interestingly, it appears that computing people have a great deal to contribute to the solution of this problem. They know how to design and build computational tools that facilitate conversational protocols, and collect, analyze, and present complex data in a form that facilitates decision-making. Prototypes of these tools appear in MMOGs (massively multiplayer online games). The challenge for computing people is to help understand the coordination skills for pluralistic networks and then design tools to enable diverse communities to quickly form effective teams. We will discuss the latest in a series of experiments we conducted with the *World of Warcraft* (*WOW*) game that leads us to be optimistic about this possibility.

***World of Warcraft* screen depicting avatars.**

PHOTOGRAPH BY BRIAN J. MATIS

The Changing Context

Most of us have enjoyed a tradition of working in organizations with clear chains of command in fairly homog-

enous communities. This tradition, which might be called "hierarchical uniformity," is no longer valid for many groups. Instead, many groups are confronted with what might be called "diversified nonuniformity." In this context, teams are multicultural, deadlines are short, actions are automatic (nonreflective), decision making is distributed, leadership is earned, performance assessment is purely merit based, in-person meetings are infrequent, resources are insufficient, information is overwhelming, and sensory data is conflicting.

It is no surprise that hastily formed networks for disaster relief are fertile grounds for miscoordination: they violate the tradition dramatically.[2] Participants from hierarchical uniform organizations have little need to practice coordination in pluralistic networks. When they convene in such a network, they are unprepared to work together.

The hierarchical uniform tradition goes hand in hand with three other beliefs about effective teams. One is the notion of "best practices": the leadership finds a "best" way to do something and requires everyone to do it that way. In our experience, this notion is incompatible with pluralistic networks. There is no one "best way" for a diversified team to accomplish its mission. It must adapt and flow with a constant stream of new possibilities.

Second is relativism, the notion that all team member worldviews are equally valid and, hence, the common ground must be found in the absence of universal values. We believe, to the contrary, that there are universal values. Seven of them motivate the practices we recommend below. For example, asking for and receiving binding commitments is universal, although the style of making requests and promises varies among cultures. Another example is that everyone believes in "do not kill any person," although many do not hesitate to kill those whom their culture defines as "non-persons."

Third is team stages of development, the notion that teams move through the stages that Bruce Tuckerman called "forming, storming, norming, and performing."[9] This is useful guidance for leaders of relatively homogenous teams. In pluralistic networks, the formation of leadership itself becomes a central concern. There is no externally appointed leader who can guide the team through those four stages. The team's emergent leadership must do this by itself. The possibilities of miscommunication and dramatic mood shifts are constant threats.

The main issue of pluralistic networks is that the members bring different values and do not see the world the same way.

Practices for Diversified Coordination

We have been conducting experiments to understand a small but important piece of the problem: What practices do small teams need to function well in a pluralistic network? Answering this question is the first step toward building helpful computational tools.

The main issue of pluralistic networks is that the members bring different values and do not see the world the same way. We have investigated whether there are universal values that would bridge the diversity, generate mutual respect, and support everyone's dignity. We have found seven universal values and associated practices that realize them in the team:

1. Proficiency in a practice essential to the team;
2. Capacity to articulate a vision of the team's value in the world that others embrace and commit to;
3. Capacity to enter into binding commitments and fulfill them;
4. Capacity to spot and eliminate waste;
5. Capacity to share on the spot, real-time assessments of performance, for the sake of building and maintaining trust, including disclosures of moods and emotions inspired by the environment and action of the team;
6. Capacity to observe one's own history and how it interacts with the histories of the others on the team; and
7. Capacity to blend, meaning to dynamically align one's intentions, movements, and actions with those of others.

Research and experience support the hypothesis that these practices constitute the essential core for coordination in pluralistic networks. For example, Womack and Jones[11] promote "lean thinking," a practice of seeing and eliminating waste. Gladwell[4] reports on how airlines discovered that most accidents could be traced to cross-culture miscommunication in the cockpit; accidents dropped significantly after the airlines put pilots through multicultural communication training. Multicultural group processes such as the Barrett-Fry Appreciative Inquiry[1] and the Straus-Layton method[7] have been very successful at developing shared interpretation and solidarity in pluralistic communities. Strozzi-Heckler[8] reports that Leadership practices for making assessments and blending have been very effective for teams and groups. Tuomi[10] concluded that loosely formed volunteer networks of collaboration frequently fall into practices like these.

An Experiment

We recently completed a four-month experiment to examine whether an MMOG could be used as a learning environment for the core practices listed here. The diversified group consisted of 28 people who did not know each other. They came from about half a dozen countries and varied professional backgrounds. The MMOG was the *WOW* game. We chose *WOW* because it is an amazingly complex synthetic world created by a social machine from the interactions of millions of players. John Seely Brown and Douglas Thomas have already brought *WOW* to the attention of the business community as a possible training ground for leadership.[5,6]

Within the *WOW* context, it is possible to define precisely what it means for a small team to be proficient by extending the Dreyfus definitions[3] from individuals to teams. The definitions enable us to measure the progress of teams toward proficiency. The game

guides players gradually up a hierarchy of 80 levels, starting from the novice level 1. Every quest (exercise) in the game is rated for the level of players allowed to undertake it.

Players who reach a sufficient level may team with others in groups for raids into "dungeons" that house powerful denizens (called "bosses") that cannot be defeated by individuals. Successful raids are a measure of a team's coordination proficiency under pressure. We measured team learning proficiency by the number of successful raids at each level of difficulty, and by the new actions team members were applying to their daily lives.

Each player satisfied the first practice on the list above by attaining a sufficient game level. We set up general team practices for the remainder of the list. Observers accompanied the teams in-game to monitor their coordination and coach them on their use of the general practices. The observer made sure that the team paused periodically to share their moods and honest performance assessments (practice 5 on the list); this enabled them to regenerate their shared interpretation of what they were doing.

On completion of each in-game assignment, the teams debriefed in a standard after-action assessment exercise to critique each other's performances, reflect on their overall effectiveness, and plan new strategies for their next assignment. They also reflected on how the coordination practices they were learning would apply in their real-life worlds.

Some in-game assignments were team raids to defeat high-level bosses.

Avatar used in team-building experiment.

The inability to achieve proficient coordination in pluralistic networks is a real problem.

One of the bosses was so tough that there was no hope for any team to survive; the purpose was to see how the teams handled their moods when faced with an impossible situation.

We observed that the general coordination practices were initially unfamiliar to most team members. Even after the first month of working together, many members had difficulties voicing assessments of their teammates. Slowly they learned that sharing performance assessments was progressively easier with practice and they overcame their aversions. Over time, the regular practice of making these assessments ceased to embarrass or to generate hard feelings. Because acting on these assessments significantly improved their team success the teams came to value them. Their mutual respect, solidarity, and team effectiveness improved markedly. By the end of the four months, teams openly wondered why they had not been using these practices at work.

In the first two months, only one of the six teams achieved solidarity and clear proficiency. We then shuffled the team members into new teams for the next two months. This time, all teams achieved solidarity and proficiency.

The experiment validated our intuition that the general practices foster proficient diversified coordination.

Conclusion

The inability to achieve proficient coordination in pluralistic networks is a real problem. It is becoming worse as the global Internet creates more connections and more opportunities for people to work together across international and organization boundaries. Disaster relief experiences have called wide attention to the problem, and have stimulated research into what is needed for coordination in pluralistic networks.

The universal values of articulating visions, making and fulfilling commitments, eliminating waste, sharing performance assessments, disclosing moods, observing histories, and blending, underlie an enabling core of general team practices that lead to proficiency at pluralistic coordination. The MMOG game environment is a means of engaging teams in complex tasks requiring sophisticated use of these practices in a synthetic world.

Preliminary examples of computational tools to facilitate these practices can be seen already in the *WOW* game environment. Numerous interface add-ons present situational information in easy-to-interpret formats. Group forming tools make the process of creating diversified teams ridiculously easy. Voice-over-IP tools facilitate group conversations for coordination.

Despite the preliminary nature of these conclusions, the results are sufficiently intriguing to warrant a wider discussion of how computing professionals can help with this important problem. C

References

1. Barrett, F. and Fry, R. *Appreciative Inquiry.* Taos Institute, 2005.
2. Denning, P. Hastily formed networks. *Commun. ACM 49*, 4 (Apr. 2006), 15–20.
3. Dreyfus, H. *On The Internet.* Routledge, 2004.
4. Gladwell, M. *Outliers: The Story of Success.* Little Brown, 2008.
5. Seely Brown, J., and Thomas, D. You play World of Warcraft? You're hired! *Wired* (Apr. 2006).
6. Seely Brown, J. and Thomas, D. The gamer disposition. *Harvard Business Review* (Feb. 2008).
7. Straus, D., and T. Layton. *How to Make Collaboration Work.* Berrett-Koehler publishers, 2002.
8. Strozzi-Heckler, R. *The Leadership Dojo.* Frog, 2007.
9. Tuckerman, B. Developmental sequence in small groups. *Psychological Bulletin 63* (6), 384–399; http://findarticles.com/p/articles/mi_qa3954/is_200104/ai_n8943663/
10. Tuomi, I. *Networks of Innovation.* Oxford Press, 2003.
11. Womack, J. and Jones, D. *Lean Thinking.* Simon & Schuster, 1996.

Peter J. Denning (pjd@nps.edu) is the director of the Cebrowski Institute for Innovation and Information Superiority at the Naval Postgraduate School in Monterey, CA, and is a past president of ACM.

Fernando Flores (contactoflores@gmail.com) is an author, entrepreneur, and current Senator of Chile. He is the founder of multiple companies, including Action Technologies, Business Design Associates, and Pluralistic Networks.

Peter Luzmore (peter@Luzmore.com) is consultant, entrepreneur, and a leadership development trainer.

DOI:10.1145/1666420.1666435

Richard Tapia

Broadening Participation
Hiring and Developing Minority Faculty at Research Universities

Emphasizing the importance of creating more programs and investing more funding toward the goal of developing minority faculty at research universities.

No one was surprised, certainly not those of us who sit in science and engineering faculty meetings as the only underrepresented minority in the room, by Donna Nelson's recent study results—the second edition of which was released this January[1]—of tenured and tenure track faculty in the top science and engineering departments (as ranked by the U.S. National Science Foundation according to research funds expended.). Nelson concludes "There are relatively few tenured and tenure-track underrepresented minority (URM) faculty in these research university departments, even though a growing number and percentage of minorities are completing their Ph.D.s. Qualified minorities are not going to faculties of many science and engineering disciplines." While computer science had the lowest percentage of URM professors in 2002, other disciplines, noticeably math and physics, grew increasingly worse in the ensuing five years to equal this distinction (see http://chem.ou.edu/~djn/diversity/Faculty_Tables_FY07/07Report.pdf for the complete data set from the second edition of the report).

Richard Tapia at the Tapia Celebration of Diversity in Computing, April 2009.

PHOTOGRAPH BY JOANNA PAPAKONSTANTINOU

Importance of Minority Faculty at Research Universities

Nelson makes a strong point in her report on the importance to the university and to the discipline of having minority faculty. She says, "Dearth of minority faculty at a university or in a discipline discourages minority students from selecting that university or discipline, since most students are comfortable in environments that include people with backgrounds and characteristics similar to theirs."[1] Students who do choose the discipline need role models and mentors to inspire, motivate, and encourage them.

Over the years at Rice University, I have directed or co-directed 23 URMs or women Ph.D. doctoral recipients in Computational and Applied mathematics and lead an NSF Alliance for Graduate Education and the Professoriate (AGEP) with approximately 65 URM students from across science and engineering. Each year I teach an advanced-level class in optimization theory in the engineering division. Minority students from the various engineering disciplines are invariably drawn to my class. They seem to be motivated to perform well, and usually do. Often a minority student is at the top of the class even though there are many excellent non-minority students in the class. A few years ago, I had 24 students in class and 12 were minority. Just think: 50% of the students in an advanced level class at a Tier 1 Research School were minorities.

As minority faculty we serve as role models in two directions. We demon-

strate feasibility to the minority students and show the non-minorities that we as minorities can be excellent teachers and faculty. We promote understanding in components that non-minority faculty members cannot.

I want to make what I believe is an often-overlooked critical point about the importance of minority faculty at our best research universities. Leadership in science and engineering comes from top research institutions. I believe that much of my national leadership has been possible because I am a faculty member at a respected university with respected research credentials. I am often asked to speak to research university presidents, faculty members, and national government leaders about representation. They listen to me because they know that I have been there. We must have strong faculty representation at the nation's leading universities in order to produce high quality URM scientists. Consequently, I strongly encourage us to create more programs and invest more funding with the goal of developing minority faculty at research universities.

What Won't Work

There is a growing movement for Minority Serving Institutions (MSIs) to develop Ph.D. programs, but Ph.D.s produced at MSIs will not become faculty at top research universities. Top research universities choose faculty from Ph.D.s produced at top research universities. I am extremely concerned that this will produce a permanent underclass. If we underrepresented minorities are ever to be an equitable presence as faculty at our top-level schools, then our students must be schooled at those same institutions. This is a hard statement for me to make. I have great friends at MSIs for whom I have great admiration. Their students speak warmly of how confident and supported they felt in their experiences there. Research universities should learn from them how to nurture that kind of confidence, but MSIs should not expect to produce graduate programs of the same caliber that more than a hundred years of investment has produced at the nation's top research universities. More about this topic can be found in my *Chronicle of Higher Education* article, "Minority Students and Research Universities: How to Overcome the 'Mismatch'."[2]

Also, filling faculty positions with foreign scholars—even those who are black, brown, or Spanish-speaking — does little to solve the problem of universities' lack of success with Mexican-American, Puerto Rican, and black youth from across the U.S. People from places like Africa, Spain, or Latin America cannot be effective role models or mentors for African-Americans and Latinos who grew up in the U.S. In fact, it is not unusual for those scholars to view their domestic-minority counterparts negatively and to strongly resist being identified with them. Many international students were admitted to graduate school in the U.S. because they were highly competitive and the best students of their nations. Often the products of early academic tracking, they have had strong educational foundations and intense, specialized study in their fields.

Also, foreign scholars were not viewed as racially or ethnically different in their countries of origin and, from their formative years on, made to feel they were second-class citizens who did not belong in higher education or in leadership positions. So when we make those hires, we must understand that we are not doing our part to increase participation, or provide role models. A fuller development on this topic can be found in my *Chronicle of Higher Education* article, "True Diversity Doesn't Come From Abroad."[3]

Another mistake we often make is of exclusively working up the ladder rather than also starting at the top and working down: starting with K–12 to increase the pool of bachelor's degrees in science and engineering, for example. As a long-term solution, this is neces-

The post-doc position may be the most critical step in either making or breaking a successful future in the academy.

sary, but we can't wait for the next generation; we need to do something now that will have an *immediate* impact.

What Will Work

Universities have the responsibility to hire and promote minority faculty members, and if we take the role seriously, we could make a significant improvement over the next five years. Here are some steps that I think we need to take:

Put qualified people in strong post-doctorate positions. Graduate research advisors must take a role in finding a strong post-doc position for students with potential. After receiving my Ph.D. from UCLA, I was guided by David Sanchez, the only underrepresented minority faculty member at that time in the UCLA Mathematics Department, to a post-doctoral position at the University of Wisconsin. This intervention and guidance was probably the most important in my entire professional life. At Wisconsin, I was very fortunate that I got to work with some of the finest mathematicians in my area. I was fully integrated into the research program. Graduate advisors must elicit a commitment of that kind of relationship from the post-doc advisor and then check to see that it is happening. The post-doc position may be the most critical step in either making or breaking a successful future in the academy.

Reexamine hiring criteria. When top level departments hire new faculty, their number one criteria is the candidate's potential to be the next Gauss or Turing. What we assess when we hire is not what we expect or need of all faculty. I will illustrate this point with a story. A few years back I was invited to the University of California Berkeley as a Regents Lecturer. I gave five different talks in five days. In my university-wide talk on diversity, I included a segment entitled "Why the Berkeley Math Department Would Never Hire Me." The reason is that my potential for winning a Fields Medal in Mathematics is low, even though I have performed solid research that would get me tenure at essentially any university including Berkeley. As I went from talk to talk the minority graduate students followed me around like I was the Pied Piper of Hamelin. I told them that my next talk really was not for graduate students. They said they did not care and just wanted to interact with me. Simply stated, I would give Berkeley more than 99% of their faculty in the broad and complete sense. Of course, I would be promoted; I would give in so many components that the university values. At universities like Berkeley, the promotion criteria are much broader than the hiring criteria, and this is good for the university and the nation.

We can't wait for the next generation; we need to do something now that will have an *immediate* impact.

At Rice in 2005 I was appointed University Professor, an honor bestowed upon only six individuals, including two Nobel Laureates, in its 100-year history. However, I did not gain this distinction for my research alone, but primarily for contributions in the other dimensions I have discussed. When I gave my acceptance speech I thanked Rice for being sufficiently progressive to allow me to do it my way. I stated that I hope this example serves to show young faculty members that there are various paths to the same place, not just one, and the other more non-traditional paths are important. The Berkeley Math Department would greatly benefit from hiring someone like me, but they are unwilling to break their traditional hiring culture. And Berkeley is of course, representative of other universities that follow the same course of action in hiring.

Mentor young faculty. The Nelson data shows a loss as members go through the tenure process, a heartbreaking failure. It is wrong to assume that beginning faculty members will understand faculty culture and what is expected of faculty members. They need someone who will be forthright with them about departmental expectations. Someone must warn them of the danger of being enticed away from research by too much leadership or outreach too soon. This mentoring must be proactive. Young minority faculty members frequently will not ask for help or express concern that there is any problem with their progress. A few years ago, the Rice Sociology Department denied tenure to a young minority woman claiming that her as yet unpublished book on minority K–12 education was not up to par with their standards. Yet this book when published was extremely well received and allowed her to be hired with tenure at an excellent Tier 1 University. In talking to this woman, she told me she was shocked by the decision and thought the department was most happy with her research. She had not had sufficient communication with her chair. The loss to Rice was huge; this young woman was the primary mentor of Rice minority women undergraduates across campus. Many a tear was shed and much anger felt when she left. In another case, a minority faculty member was denied tenure because he had extremely poor teaching evaluations. He was hired from industry, and his research was solid, but he had not been sufficiently well mentored on the need for good teaching. Rice lost a valuable faculty member who could have been saved with proper mentoring. Just as industry has for new executives, many departments are now making new faculty mentoring a formal responsibility of caring senior faculty members, and more need to do so.

We often lament the condition of representation without providing suggestions for making changes. I hope the suggestions I've made here might be the impetus for discussions in departments across the U.S. I am keenly interested in this process and welcome participation in a national effort to improve representation of university science and engineering faculty. C

References

1. Nelson, J., and Brammer, C.N. *A National Analysis of Minorities in Science and Engineering Faculties at Research Universities*," Norman, OK, October, 2007, January 2010; http://chem.ou.edu/~djn/diversity/Faculty_Tables_FY07/07Report.pdf
2. Tapia, R. Minority students and research universities: How to overcome the 'mismatch'. *The Chronicle of Higher Education 55*, 29 (Mar. 27, 2009).
3. Tapia, R. True diversity doesn't come from abroad. *The Chronicle of Higher Education 54*, 5 (Sept. 28, 2007).

Richard Tapia (rat@rice.edu) is the Maxfield-Oshman Professor in Engineering Department of Computational and Applied Mathematics at Rice University and Principal Investigator and Director of the Empowering Leadership Alliance, an NSF Broadening Participation in Computing Alliance.

DOI:10.1145/1666420.1666436

Cameron Wilson and Peter Harsha

IT Policy
Making the Case for Computing

Seeking funding for current and future computing initiatives requires both a strong argument and a broad community of supporters.

When it comes to distributing trillions in U.S. taxpayer dollars, funding for science joins a crowded field of special interests where competition for federal funding is fierce. Policymakers are ultimately stewards of taxpayer dollars and must make judgments about the areas in which government has a legitimate reason to invest. And because tax dollars are not limitless, policymakers must prioritize federal investments, deciding which programs or which agencies have the most compelling need for funding.

Consequently, every special interest—from researchers to roadbuilders, health care professionals to hovercraft manufacturers—has an advocacy group urging policymakers to focus federal investment in their particular area. What ties all of these groups together is the need to have a story—a case to make to Congress, the Administration and the American people—that justifies the expenditure of those tax dollars on the things they care about.

Funding Decisions

The stakes are high. Last year (fiscal year 2009), the U.S. discretionary budget—that is, the amount not automatically committed to federal programs like Social Security or Medicare—was just over $1 trillion. Congress spent that money, as it does every year, by parceling it out to federal agencies and programs in 12 separate pieces of legislation. This is quite literally a zero-sum game. Aggregate spending by Congress is capped, and each of these 12 appropriations bills has its own spending cap. This means that once the spending caps are reached—and they always are—any additional increase in spending for one program must be offset by an equal reduction in another program.

As a result, policymakers find the need to invest in fundamental research in competition with the need to fund agricultural subsidies, or the

VISUALIZATION BY JULIANA FREIRE AND CLAUDIO SILVA. UNIVERSITY OF UTAH

Government funding for computing research is tight and the competition plentiful. A new infrastructure for computational oceanography incorporating the VisTrails system created by the University of Utah was among the scientific projects receiving support from The National Science Foundation's Cluster Exploratory (CluE) program in 2009.

need to support ongoing military efforts in Afghanistan and Iraq, or the need to fund sewer projects in their own districts. In fact, it is more stark than that, because Congressional rules stipulate that any increase to a program in one of the 12 appropriations bills must be offset by a decrease to a program in that same bill. So, additional increases in spending for federal science agencies like the National Science Foundation or the National Institutes of Standards and Technology may result in cuts to another science agency like the National Oceanic and Atmospheric Administration, or to a program to subsidize bulletproof vests for local law enforcement, or to the Census Bureau, because they all reside in the same bill.

So just like any other special interest group, advocates for science—advocates for a greater federal investment in fundamental research, and in particular, for computing research—have had to learn to make a case compelling enough to survive in this competition for funding. But unlike other special interest groups, science advocacy groups like the Computing Research Association or ACM's U.S. Public Policy Committee compete at a disadvantage because we lack (due to legal restrictions and organizational cultures) political action committees (PACs) to contribute to the campaigns of members of Congress or vast resources to fly congressional delegations out to exotic locales on fact-finding trips. Our success is based solely on the strength of our arguments and an active community making them.

While we are limited in the tools of influence, we have a powerful case. Fundamental research in information technology has led to tangible breakthroughs that have created entire new industries, driven economic growth, and developed deep and productive relationships between industry and universities.

Computing Advances

Advances in computing have changed all aspects of our lives: how we conduct commerce, how we learn, our employment, our health care, how we manufacture goods, how government functions, how we preserve our national security, how we communicate, and how we're entertained.

Computing facilitates innovation because a vital IT R&D ecosystem enables innovation within IT itself.

Advances in computing drive our economy—not just through the growth of the IT industry, but also through productivity gains across the entire economy. Recent analysis suggests that the remarkable economic growth the U.S. experienced between 1995 and 2002 was spurred by an increase in productivity enabled almost completely by factors related to IT.[2] The processes by which advances in information technology enable productivity growth, enable the economy to run at full capacity, enable goods and services to be allocated more efficiently, and enable the production of higher quality goods and services are now well understood.[1]

Advances in computing enable innovation in all other fields. In business, advances in IT are giving researchers powerful new tools, enabling small firms to significantly expand R&D, boosting innovation by giving users more of a role, and letting organizations better manage the existing knowledge of its employees.[2] In science and engineering, advances in IT are enabling discovery across every discipline—from mapping the human brain to modeling climatic change. Researchers, faced with research problems that are ever more complex and interdisciplinary in nature, are using IT to collaborate across the globe, and to collect, manage, and explore massive amounts of data. Computer modeling, visualization, and data analysis have joined observation, theory and experiment as the drivers of scientific discovery.

Advances in computing continue unabated. Worldwide, there has been no slowdown in the pace of innovation, the production of new ideas, the discovery of additional opportunities to advance products and services for society.

Thus, leadership in computing is essential to the U.S., economically and socially.

Future Opportunities

While the history of computing-related contributions to shaping our world is a compelling topic, future opportunities in computing—where the field might go and what problems it might tackle—are perhaps even more compelling. Whether it's creating the future of networking, revolutionizing transportation, delivering personalized education, enabling the smart grid, empowering the developing world, improving health care, or driving advances in all fields of science and engineering—all national priorities—computing has key contributions to make and key roles to play. In March 2009, the National Academy of Engineering unveiled 14 "Grand Challenges for Engineering" for the 21st century (see http://www.engineeringchallenges.org/). The majority of these—the majority of the "Grand Challenges" for *all of engineering*—have either substantial or predominant information technology content:

- Secure cyberspace
- Enhance virtual reality
- Advance health information systems
- Advance personalized learning
- Engineer better medicines
- Engineer the tools of scientific discovery
- Reverse engineer the brain
- Prevent nuclear terror (to a great extent a sensor network and data mining problem)

And there are many more information technology challenges of equally high impact:

- Create the future of networking
- Empower the developing world through appropriate information and communication technology
- Revolutionize transportation safety and efficiency
- Build truly scalable computing systems, and devise algorithms for extracting knowledge from massive volumes of data
- Engineer advanced "robotic pros-

thetics" and, more broadly, enhance people's quality of life

- Instrument your body as thoroughly as your automobile
- Engineer biology (synthetic biology)
- Revolutionize our electrical energy infrastructure: generation, storage, transmission, and consumption
- Achieve quantum computing

It is impossible to imagine a field with greater opportunity to change the world.

Computing facilitates innovation because a vital IT R&D ecosystem enables innovation within IT itself. At the heart of this ecosystem is federally sponsored research. A 1995 study by the National Research Council (NRC) describes the "extraordinarily productive interplay of federally funded university research, federally and privately funded industrial research, and entrepreneurial companies founded and staffed by people who moved back and forth between universities and industry." That study, and a subsequent 1999 report by the President's Information Technology Advisory Committee, emphasized the "spectacular" return on the federal investment in long-term IT research and development. Indeed, a 2003 NRC study identified 19 multibillion-dollar IT industries—industries that are transforming our lives and driving our economy—that were enabled by federally sponsored research (see http://books.nap.edu/openbook.php?record_id=10795&page=5).

Academia and Industry

Beyond transforming society and bolstering economic growth, funding for computing research and the subsequent development of the U.S. IT sector has created particularly strong relationships between universities and industry. Robust funding for research has allowed university research to assume the role of focusing on fundamental questions and long-term problems, without supplanting industrial research and development. While industry research, geared primarily toward short-term development, does not supplant university research.

In fact, industry generally avoids long-term research because it entails risk in a couple of unappealing ways. First, it is difficult to predict the outcome of fundamental research. The value of the research may surface in unanticipated areas. Second, fundamental research, because it is published openly, provides broad value to all players in the marketplace. It is difficult for any one company to "protect" the fundamental knowledge gleaned from long-term research and capitalize on it without everyone in the marketplace having a chance to incorporate the new knowledge into their thinking.

Industry generally avoids long-term research because it entails risk in a couple of unappealing ways.

A sustained, robust commitment to long-term, fundamental research is also necessary because the innovations that drive the new economy today are the fruits of investments the federal government made in fundamental research 10, 15, or even 30 years ago. Essentially every aspect of information technology upon which we rely today—the Internet, Web browsers, public key cryptography for secure credit card transactions, parallel database systems, high-performance computer graphics, portable communications...essentially every billion-dollar sub-market—is a product of this commitment and bears the stamp of federally supported research.

Computing has a compelling story, and fortunately one that finds a lot of support in Congress and in the Administration. The federal government currently invests more than $3 billion per year in information technology R&D across 13 different agencies, and that figure could increase significantly if the Obama administration follows its plan to increase funding at key science agencies and Congress concurs. However, looking forward, making our case will be more important than ever.

Not only is society faced with grand challenges that will require fundamental breakthroughs in computing, but competition for scarce federal dollars is going to be more intense than ever. The competitive environment we've described was largely in the era of U.S. federal deficits of billions of dollars; today the federal deficit is over a trillion dollars with major spending proposals—such as health care reform—currently winding through Congress. The budget politics driving these issues are the same politics that can affect spending for fundamental research. Without a strong case and support from a broad community (industry, higher education, and scientific societies) in making it, research funding and the innovations it enables will face a chilly reception among policymakers.

With your help, we'll continue to make the case for computing research wherever we can. We encourage you to take advantage of any opportunities you might have in your own community to do the same.

Authors' Note: The inspiration for this column, and indeed some of the text, came from a white paper prepared by Peter Harsha along with Edward Lazowska (University of Washington) and Peter Lee (Carnegie Mellon University). The white paper ("Information Technology R&D and U.S. Innovation") was one of a series prepared in December 2008 at the request of the Obama Administration by the Computing Community Consortium, to aid in the transition of Presidential Administrations. The collected series of white papers, entitled *Computing Research Initiatives for the 21st Century*, is available at http://www.cra.org/ccc/initiatives.

References

1. Atkinson, R.D. and McKay, A.S. *Digital Prosperity: Understanding the Economic Benefits of the Information Technology Revolution.* Information Technology and Innovation Foundation. 2007; http://www.itif.org/files/digital_prosperity.pdf
2. Jorgenson, D.W., Ho, M.S., and Stiroh, K.J. *Productivity, Volume 3: Information Technology and the American Growth Resurgence.* MIT Press. 2005.

Cameron Wilson (wilson_c@hq.acm.org) is the director of the ACM U.S. Public Policy Office in Washington, D.C.

Peter Harsha (harsha@cra.org) is the director of government affairs at the Computing Research Association (CRA) in Washington, D.C.

DOI:10.1145/1666420.1666437

Kieron O'Hara and Nigel Shadbolt

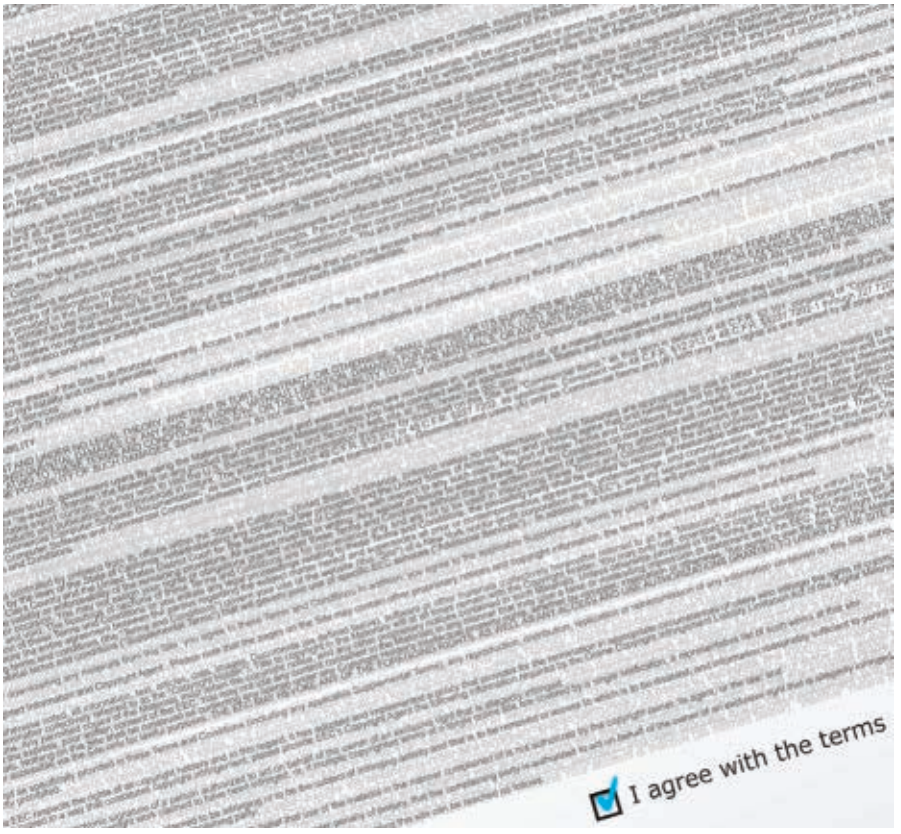
Viewpoint
Privacy on the Data Web

Considering the nebulous question of ownership in the virtual realm.

The World Wide Web in its current form, linking documents with hyperlinks in an associative network, has led to a number of concerns about issues related to privacy, copyright, and intellectual property.[6] But the movement away from the linking of *documents* to the linking of data, a much more powerful paradigm allowing automation of a greater number of information processing tasks, will test legal and technical regimes still further.

The linked data Web, in which heterogeneous data is brought together from distributed sources relatively seamlessly with user-provided ontologies, allows information about individuals or organizations to be queried despite being collected at different times for different purposes, with different provenances and different formats. The benefits of such a Web are manifest[6,9] but threats to personal privacy will also increase as boundaries blur between personal information published intentionally, that published conditionally (for example, to specific social networking sites for a specific audience) and information over which the subject has no control.

One way of expressing the dilemmas that will face us is to ask the question "who owns all this data?" When it is personal data, surely we do? Perhaps surprisingly, the answer is no. Even if you enter the data yourself, for example onto some Internet service, you do not own it—the service generally does.



You will have signed up for something in the small print—that is, you will tacitly have consented to handing over the data. Given the highly interactive nature of the Web where one creates data consciously and unconsciously all the time, this consent model will be increasingly stretched over the next few years.

It has always been somewhat flawed, with few limits to the uses to which data is put when consent to process has been lawfully obtained (and privacy policies may change after one has consented[4]). Naïve users and minors often treat policies, or terms and conditions, as a tedious box necessary to check to get onto a site, rather than as signing away their rights.[8] But even when there are no problems of asymmetric information or proportionality, there are social issues to be considered—privacy is not a private matter. It impacts on a series of wider communities.

A social network includes lots of information not directly about you. The information is implicit, but network analysis makes it explicit. The evidence of a network is circumstantial, but an important basis for profiling. For example, if you have a high percentage of gay friends does that mean you are gay? Many people—gay or straight—would find that inference embarrassing.

We do not own our networks. In January 2008, blogger Robert Scoble automatically harvested the names and email addresses of his several thousand Facebook friends, and exported them to another account. The row was resolved amicably in the end—but the outcome was that Scoble's network was not his to harvest.

Given the benefits of wide access to data, it is appropriate to ask whether "ownership" is the concept needed. In the first place, legal frameworks that define a type of data ownership for the subject are absent—these are facts *about* a person, not copyright material, intellectual property, or trade secrets.

Second, the most important power of ownership is denial of access: if I own something, I can stop you using it. But this undermines the potential of the Web of linked data. In the old days of paper and practical obscurity, the value of information was in its scarcity, but on the Data Web value comes from abundance, the ability to place information in new and unexpected contexts, facilitating what Tim Berners-Lee calls "serendipitous reuse."[9] Ensuring data is correct is more valuable than preventing its use. We should also not ignore the opposite pull from rights of access to information, as a corollary to rights of freedom of expression,[7] while many people and organizations have legitimate interests in access to data.

This is the rationale for data protection, whose aim is not exclusively to protect individuals' privacy, but rather to balance privacy with the maintenance of the free flow of information, as well as other desirable things for individuals like quality and accessibility.[10] Under a data protection regime, individuals have the right to inspect and correct information being held about them, in theory allowing them to address issues of incorrectness, inappropriateness, excessiveness, and so on.

It also has the effect of bringing rules into the area directly—data protection provides controls administered by a regulatory body over how data should be handled. On the other hand, one's privacy can only be addressed under an ownership regime in court *after* a tort or legal injury had occurred as a result of misuse.

In Europe, the 1981 Council of Europe Convention on data protection was required to reconcile the right of privacy in Article 8 of the European Convention on Human Rights with the right of freedom of expression given in Article 10. The Convention led directly to the EU's directive on data protection in 1995 (95/46/EC), and to national legislation such as the U.K.'s Data Protection Act of 1998.[10] Most industrialized nations have some sort of data protection legislation in place, although European laws are probably the most comprehensive.

There are differences between jurisdictions, of which some extend protection to legal entities like companies, others include non-digital information under the remit, others have restricted data protection to public sector data, while still more have argued that information affecting national sovereignty or sociocultural interests should also fall under the banner, with states having rights as well as individuals.

This variation is often cultural; some nations value privacy more than others, Continental Europeans worry about corporations' access to data, while Anglophone nations tend to be more suspicious of governments, and so on. Yet it also matters economically—some senior business people suspect that such is the value of data that businesses in those states with strong data protection laws, such as Germany, could well lose out to those in jurisdictions with less protection, such as the U.K.

Given the benefits of wide access to data, it is appropriate to ask whether "ownership" is the concept needed.

Different regimes offer different levels of protection. Consider for instance the definition of personal data. Belgium has incorporated the wording of Directive 95/46/EC directly into law, covering anyone who can be identified directly or indirectly from the data, while the U.K. has altered the wording to cover only those who could be identified *by the data controller* from the data. Data that can be used to identify one (such as an IP address) can be collected without data protection legislation in the U.K. as long as the controller has no way of going from IP address to an individual.[8]

Nevertheless, the Web is an opaque place, especially to non-expert users. Putting the onus on the data *subject* to ask for details of how personal data is being used ensures that much will be missed—how many know the right questions to ask about cookies, ISPs, search engines, or browsers? Will it pay regulators to take a stronger stance?

Regulation of the Web is a complex matter, crossing jurisdictions and posing problems for the W3C's consensus-based standards approach. Regulation generally leverages normality, and is premised on common behavior and shared interpretations of a situation.[4,11] It is more effective if it goes with the grain of a society's norms, but online there is no "normal" behavior, as work on the scale-free aspects of the Web has repeatedly demonstrated (recently in Meiss et al.[5]), while user understanding of online situations is highly heterogeneous.

The Web moves so quickly that regulation is risky. It takes time and coordination across borders; by the time rules are in place, behavioral patterns may likely have changed, and all that is left is unintended consequences.[6] Directive 95/46/EC dates back to 1995, with key updates to cover traffic and location data introduced in 2002. The scale and speed of the Web's evolution means that carefully considered regulation is rarely timely; the whole privacy-threatening phenomenon of Web 2.0 has arisen since those directives. For example, in social network sites friends sometimes take information that a user had originally character-

ized as private to them and republish it to their immediate friends. The discipline of Web Science covered recently in these pages[3] is an attempt to harness transdisciplinary endeavor to try to understand the complex feedback cycles between the Web and society.

If ownership and regulation are problematic, what to do? We have two proposals, one modest, one a little deeper.

As things stand, privacy is a game for the rich and well informed, creating a digital divide to which one response is to redress the balance by exploring ways in which people can perceive advantage from protecting their privacy. In particular, if we can shift the emphasis from concealment to transparency—from the *concealment* of data from potential users, to *transparency* of how data is being used—we can begin to provide answers to questions like "who is looking at you?" and "what is being said about you?" Data will continue to be gathered, aggregated and graphed, but its use should be clear and traceable. We are of course gesturing toward the work of Daniel Weitzner and colleagues on information accountability, reported in this magazine.[11]

With a proper infrastructure in place, it should be possible to construct legal/technical/economic models where people can be recompensed for the use of their data—you could be paid for your clickthroughs. Or perhaps you would require a donation to a cause of your choice in return for your clickthroughs. If others are making money from observing your activity, it doesn't seem outrageous that you or your nominees should be compensated.

It may be that the commercial thirst for consumer data is about to wane as the global financial crisis undermines advertising, and therefore the business models of many Internet companies. But this idea is just one instance of a more general principle of reciprocity between technology developers and information subjects. If a technology makes public service more efficient, or a business process more profitable, then it should also be used reciprocally to aid the citizen or consumer.

If government officials have better access to data as a result of technology, then citizens should too—improved data for government implying more freedom of information. Indeed, this is the thrust of the Making Public Data Public project on which Nigel Shadbolt and Tim Berners-Lee are advising the U.K. government.[2] Although the project is focusing on non-personal public sector information the premise is that more data increases transparency and can drive public sector improvement and reform. In the context of personal information a consumer should be able to get improvements in data protection, for example by being able to use technology to enforce access to information in the many jurisdictions where such enforcement is currently problematic.[7]

Perhaps we should be talking of the *responsibilities* of privacy too.

As our rights as citizens and as consumers seem to be coming together, markets could be redefined to change the incentives to protect one's own privacy and respect that of others, for example, as with principles such as 'the polluter pays.' The analogy with pollution is suggestive for our more fundamental idea—an invasion of privacy has things in common with pollution, in particular that the individual benefits and costs do not capture the full social costs.

In many jurisdictions, particularly common law ones, the complexities of privacy are dealt with by exploiting collective wisdom, referring individual cases to a reasonable expectation of privacy. In other words, if one behaves in such a way that one could not reasonably expect to be private, then others are not liable for invading one's privacy. Reasonable expectations change through time and space, making law sensitive to context.

Online, reasonable expectations are diminishing all the time, as our clicks are logged and people generously give information about themselves and others away to their social networks. Surveillance is becoming the norm, with the complicity of many data subjects. But might this be a social harm?

Privacy is essential for the proper functioning of a liberal, democratic society. Some benefits may accrue to the individual (who gains autonomy, a space of intimacy, freedom of speech, and so forth). But equally benefits accrue to society—a free, liberal polity of autonomous individuals is a public good, in the same way that clean air is. Everyone benefits, even if not everyone contributes.

If privacy is a public, not a private, good, then talking exclusively of rights is not the right way to go. Perhaps we should be talking of the *responsibilities* of privacy too. This would involve something of a culture change, especially in our voyeuristic society.[1] But this would not be unprecedented: it was privacy activists, not the law, which pressured Web sites in the 1990s to respect privacy rather than promiscuously gathering and selling consumers' data.[4]

Perhaps it is our duty to ensure that reasonable expectations of privacy are kept high. C

References

1. Anderson, D. The failure of American privacy law. In B.S. Markesinis, Ed., *Protecting Privacy*, Oxford University Press, Oxford, 1999.
2. Berners-Lee, T. and Shadbolt, N. Put your postcode in, out comes the data. *The Times* (Nov. 18, 2009).
3. Hendler, J. et al. Web Science: An interdisciplinary approach to understanding the Web. *Commun. ACM 51*, 7 (July 2008), 60–69.
4. Hetcher, S.A. *Norms in a Wired World*, Cambridge University Press, Cambridge, 2004.
5. Meiss, M.R., Menczer, F., and Vespignani, A. Structural analysis of behavioural networks from the Internet. *Journal of Physics A: Mathematical and Theoretical 41*, 22 (June 2008); doi: 10.1088/1751-8113/41/22/224022.
6. O'Hara, K. and Shadbolt, N., *The Spy in the Coffee Machine: The End of Privacy As We Know It*. Oneworld, Oxford, 2008.
7. Pitt-Payne, T. Access to electronic information. In C. Reed, and J. Angel, Eds., *Computer Law: The Law and Regulation of Information Technology*, 6th ed., Oxford University Press, Oxford, 2007.
8. Poullet, Y. and Dinant, J.M. The Internet and private life in Europe: Risks and aspirations, In A.T. Kenyon and M. Richardson, Eds., *New Dimensions in Privacy Law*, Cambridge University Press, Cambridge, 2006.
9. Shadbolt, N., Hall, W., and Berners-Lee, T. The Semantic Web revisited. *IEEE Intelligent Systems 23*, 3 (May/June 2006), 96–101.
10. Walden, I., Privacy and data protection. In C. Reed and J. Angel. *Computer Law: The Law and Regulation of Information Technology*, 6th ed., Oxford University Press, Oxford, 2007.
11. Weitzner, D. et al. Information accountability. *Commun. ACM 51*, 6 (June 2008), 82–87.

Kieron O'Hara (kmo@ecs.soton.ac.uk) is a senior research fellow in Electronics and Computer Science at the University of Southampton, and a research fellow of the Web Science Trust at the University of Southampton, U.K.

Nigel Shadbolt (nrs@ecs.soton.ac.uk) is Professor of Artificial Intelligence and Deputy Head (Research) of the School of Electronics and Computer Science at the University of Southampton, and information advisor to the U.K. government.

DOI:10.1145/1666420.1666439

Article development led by acmqueue
queue.acm.org

Kirk McKusick and Sean Quinlan discuss the origin and evolution of the Google File System.

GFS: Evolution on Fast-Forward

DURING THE EARLY stages of development at Google, the initial thinking did not include plans for building a new file system. While work was under way on one of the earliest versions of the company's crawl and indexing system, however, it became quite clear to the core engineers that they really had no other choice—thus, the Google File System (GFS) was born.

Given that Google's goal was to build a vast storage network out of inexpensive commodity hardware, it had to be assumed that component failures would be the norm—meaning that constant monitoring, error detection, fault tolerance, and automatic recovery must be an integral part of the file system. Also, even by Google's earliest estimates, the system's throughput requirements were going to be daunting by anybody's standards—featuring multi-gigabyte files and data sets containing terabytes of information and millions of objects. Clearly, this meant traditional assumptions about I/O operations and block sizes would have to be revisited. There was also the matter of scalability. This was a file system that would surely need to scale like no other. Of course, back in those earliest days, no one could have possibly imagined just how much scalability would be required. They would learn about that soon enough.

Still, nearly a decade later, most of Google's mind-boggling store of data and its ever-growing array of applications continue to rely upon GFS. Many adjustments have been made to the file system along the way, and—together with a fair number of accommodations implemented within the applications that use GFS—they have made the journey possible.

To explore the reasoning behind a few of the more crucial initial design decisions as well as some of the incremental adaptations that have been made since then, Sean Quinlan was asked to pull back the covers on the changing file-system requirements and the evolving thinking at Google. Since Quinlan served as the GFS tech leader for a couple of years and continues now as a principal engineer at Google, he's in a good position to offer that perspective. As a grounding point beyond the Googleplex, Kirk McKusick was asked to lead the discussion. He is best known for his work on BSD (Berkeley Software Distribution) Unix, including the original design of the Berkeley FFS (Fast File System).

The discussion starts at the beginning—with the unorthodox decision to base the initial GFS implementation on a single-master design. At first blush, the risk of a single centralized master becoming a bandwidth bottleneck—or worse, a single point of failure—seems fairly obvious, but it turns out Google's engineers had their reasons for making this choice.

MCKUSICK: One of the more interesting—and significant—aspects of the original GFS architecture was the decision to base it on a single master. Can you walk us through what led to that decision?

QUINLAN: The decision to go with a

PHOTOGRAPH BY STEPHEN AUSTIN WELCH


Google

single master was actually one of the very first decisions, mostly just to simplify the overall design problem. That is, building a distributed master right from the outset was deemed too difficult and would take too much time. Also, by going with the single-master approach, the engineers were able to simplify a lot of problems. Having a central place to control replication and garbage collection and many other activities was definitely simpler than handling it all on a distributed basis. So the decision was made to centralize that in one machine.

MCKUSICK: Was this mostly about being able to roll out something within a reasonably short time frame?

QUINLAN: Yes. In fact, some of the engineers who were involved in that early effort later went on to build BigTable, a distributed storage system, but that effort took many years. The decision to build the original GFS around the single master really helped get something out into the hands of users much more rapidly than would have otherwise been possible.

Also, in sketching out the use cases they anticipated, it didn't seem the single-master design would cause much of a problem. The scale they were thinking about back then was framed in terms of hundreds of terabytes and a few million files. In fact, the system worked just fine to start with.

MCKUSICK: But then what?

QUINLAN: Problems started to occur once the size of the underlying storage increased. Going from a few hundred terabytes up to petabytes, and then up to tens of petabytes...that really required a proportionate increase in the amount of metadata the master had to maintain. Also, operations such as scanning the metadata to look for recoveries all scaled linearly with the volume of data. So the amount of work required of the master grew substantially. The amount of storage needed to retain all that information grew as well.

In addition, this proved to be a bottleneck for the clients, even though the clients issue few metadata operations themselves—for example, a client talks to the master whenever it does an open. When you have thousands of clients all talking to the master at the same time, given that the master is capable of doing only a few thousand operations a second, the average client isn't able to command all that many operations per second. Also bear in mind that there are applications such as MapReduce, where you might suddenly have a thousand tasks, each wanting to open a number of files. Obviously, it would take a long time to handle all those requests, and the master would be under a fair amount of duress.

MCKUSICK: Now, under the current schema for GFS, you have one master per cell, right?

QUINLAN: That's correct.

MCKUSICK: And historically you've had one cell per data center, right?

QUINLAN: That was initially the goal, but it didn't work out like that to a large extent—partly because of the limitations of the single-master design and partly because isolation proved to be difficult. As a consequence, people generally ended up with more than one cell per data center. We also ended up doing what we call a *multi-cell* approach, which basically made it possible to put multiple GFS masters on top of a pool of chunkservers. That way, the chunkservers could be configured to have, say, eight GFS masters assigned to them, and that would give you at least one pool of underlying storage—with multiple master heads on it, if you will. Then the application was responsible for partitioning data across those different cells.

MCKUSICK: Presumably each application would then essentially have its own master that would be responsible for managing its own little file system. Was that basically the idea?

QUINLAN: Well, yes and no. Applications would tend to use either one master or a small set of the masters. We also have something we called Name Spaces, which are just a very static way of partitioning a namespace that people can use to hide all of this from the actual application. The Logs Processing System offers an example of this approach: once logs exhaust their ability to use just one cell, they move to multiple GFS cells; a namespace file describes how the log data is partitioned across those different cells and basically serves to hide the exact partitioning from the application. But this is all fairly static.

MCKUSICK: What is the performance like, in light of all that?

QUINLAN: We ended up putting a fair amount of effort into tuning master performance, and it's atypical of Google to put a lot of work into tuning any one particular binary. Generally, our approach is just to get things working reasonably well and then turn our focus to scalability—which usually works well in that you can generally get your performance back by scaling things. Because in this instance we had a single bottleneck that was starting to have an impact on operations, however, we felt that investing a bit of additional effort into making the master lighter weight would be really worthwhile. In the course of scaling from thousands of operations to tens of thousands and beyond, the single master had become somewhat less of a bottleneck. That was a case where paying more attention to the efficiency of that one binary definitely helped keep GFS going for quite a bit longer than would have otherwise been possible.

It could be argued that managing to get GFS ready for production in record time constituted a victory in its own right and that, by speeding Google to market, this ultimately contributed mightily to the company's success. A team of three was responsible for all of that—for the core of GFS—and for the system being readied for deployment in less than a year.

But then came the price that so often befalls any successful system—that is, once the scale and use cases have had time to expand far beyond what anyone could have possibly imagined. In Google's case, those pressures proved to be particularly intense.

Although organizations don't make a habit of exchanging file-system statistics, it's safe to assume that GFS is the largest file system in operation (in fact, that was probably true even before Google's acquisition of YouTube). Hence, even though the original architects of GFS felt they had provided adequately for at least a couple of orders of magnitude of growth, Google quickly zoomed right past that.

In addition, the number of applications GFS was called upon to support soon ballooned. In an interview with one of the original GFS architects, Howard Gobioff (conducted just prior to his untimely death in early 2008), he recalled, "The original consumer of all our earliest GFS versions was basically this tremendously large crawling

and indexing system. The second wave came when our quality team and research groups started using GFS rather aggressively—and basically, they were all looking to use GFS to store large data sets. And then, before long, we had 50 users, all of whom required a little support from time to time so they'd all keep playing nicely with each other."

One thing that helped tremendously was that Google built not only the file system but also all of the applications running on top of it. While adjustments were continually made in GFS to make it more accommodating to all the new use cases, the applications themselves were also developed with the various strengths and weaknesses of GFS in mind. "Because we built everything, we were free to cheat whenever we wanted to," Gobioff neatly summarized. "We could push problems back and forth between the application space and the file-system space, and then work out accommodations between the two."

The matter of sheer scale, however, called for some more substantial adjustments. One coping strategy had to do with the use of multiple "cells" across the network, functioning essentially as related but distinct file systems. Besides helping to deal with the immediate problem of scale, this proved to be a more efficient arrangement for the operations of widely dispersed data centers.

Rapid growth also put pressure on another key parameter of the original GFS design: the choice to establish 64MB as the standard chunk size. That, of course, was much larger than the typical file-system block size, but only because the files generated by Google's crawling and indexing system were unusually large. As the application mix changed over time, however, ways had to be found to let the system deal efficiently with large numbers of files requiring far less than 64MB (think in terms of Gmail, for example). The problem was not so much with the number of files itself, but rather with the memory demands all of those files made on the centralized master, thus exposing one of the bottleneck risks inherent in the original GFS design.

It could be argued that managing to get GFS ready for production in record time constituted a victory in its own right and that, by speeding Google to market, this ultimately contributed mightily to the company's success.

MCKUSICK: I gather from the original GFS paper [in *Proceedings of the 2003 ACM Symposium on Operating Systems Principles*] that file counts have been a significant issue for you right along. Can you go into that a little bit?

QUINLAN: The file-count issue came up fairly early because of the way people ended up designing their systems around GFS. Let me cite a specific example. Early in my time at Google, I was involved in the design of the Logs Processing system. We initially had a model where a front-end server would write a log, which we would then basically copy into GFS for processing and archival. That was fine to start with, but then the number of front-end servers increased, each rolling logs every day. At the same time, the number of log types was going up, and then you'd have front-end servers that would go through crash loops and generate lots more logs. So we ended up with a lot more files than we had anticipated based on our initial back-of-the-envelope estimates.

This became an area we really had to keep an eye on. Finally, we just had to concede there was no way we were going to survive a continuation of the sort of file-count growth we had been experiencing.

MCKUSICK: Let me make sure I'm following this correctly: your issue with file-count growth is a result of your needing to have a piece of metadata on the master for each file, and that metadata has to fit in the master's memory.

QUINLAN: That's correct.

MCKUSICK: And there are only a finite number of files you can accommodate before the master runs out of memory?

QUINLAN: Exactly. And there are two bits of metadata. One identifies the file, and the other points out the chunks that back that file. If you had a chunk that contained only 1MB, it would take up only 1MB of disk space, but it still would require those two bits of metadata on the master. If your average file size ends up dipping below 64MB, the ratio of the number of objects on your master to what you have in storage starts to go down. That's where you run into problems.

Going back to that logs example, it quickly became apparent that the natural mapping we had thought of—and which seemed to make perfect sense back when we were doing our back-of-the-envelope estimates—turned out not to be acceptable at all. We needed to find a way to work around this by fig-

uring out how we could combine some number of underlying objects into larger files. In the case of the logs, that wasn't exactly rocket science, but it did require a lot of effort.

MCKUSICK: That sounds like the old days when IBM had only a minimum disk allocation, so it provided you with a utility that let you pack a bunch of files together and then create a table of contents for that.

QUINLAN: Exactly. For us, each application essentially ended up doing that to varying degrees. That proved to be less burdensome for some applications than for others. In the case of our logs, we hadn't really been planning to delete individual log files. It was more likely that we would end up rewriting the logs to anonymize them or do something else along those lines. That way, you don't get the garbage-collection problems that can come up if you delete only some of the files within a bundle.

For some other applications, however, the file-count problem was more acute. Many times, the most natural design for some application just wouldn't fit into GFS—even though at first glance you would think the file count would be perfectly acceptable, it would turn out to be a problem. When we started using more shared cells, we put quotas on both file counts and storage space. The limit that people have ended up running into most has been, by far, the file-count quota. In comparison, the underlying storage quota rarely proves to be a problem.

MCKUSICK: What longer-term strategy have you come up with for dealing with the file-count issue? Certainly, it doesn't seem that a distributed master is really going to help with that—not if the master still has to keep all the metadata in memory, that is.

QUINLAN: The distributed master certainly allows you to grow file counts, in line with the number of machines you're willing to throw at it. That certainly helps.

One of the appeals of the distributed multimaster model is that if you scale everything up by two orders of magnitude, then getting down to a 1MB average file size is going to be a lot different from having a 64MB average file size. If you end up going below 1MB, then you're also going to run into other issues that you really need to be careful about. For example, if you end up having to read 10,000 10KB files, you're going to be doing a lot more seeking than if you're just reading 100 1MB files.

My gut feeling is that if you design for an average 1MB file size, then that should provide for a much larger class of things than does a design that assumes a 64MB average file size. Ideally, you would like to imagine a system that goes all the way down to much smaller file sizes, but 1MB seems a reasonable compromise in our environment.

MCKUSICK: What have you been doing to design GFS to work with 1MB files?

QUINLAN: We haven't been doing anything with the existing GFS design. Our distributed master system that will provide for 1MB files is essentially a whole new design. That way, we can aim for something on the order of 100 million files per master. You can also have hundreds of masters.

MCKUSICK: So, essentially no single master would have all this data on it?

QUINLAN: That's the idea.

With the recent emergence within Google of BigTable, a distributed storage system for managing structured data, one potential remedy for the file-count problem—albeit perhaps not the very best one—is now available.

The significance of BigTable goes far beyond file counts, however. Specifically, it was designed to scale into the petabyte range across hundreds or thousands of machines, as well as to make it easy to add more machines to the system and automatically start taking advantage of those resources without reconfiguration. For a company predicated on the notion of employing the collective power, potential redundancy, and economies of scale inherent in a massive deployment of commodity hardware, these rate as significant advantages indeed.

Accordingly, BigTable is now used in conjunction with a growing number of Google applications. Although it represents a departure of sorts from the past, it also must be said that BigTable was built on GFS, runs on GFS, and was consciously designed to remain consistent with most GFS principles. Consider it, therefore, as one of the major adaptations made along the way to help keep GFS viable in the face of rapid and widespread change.

MCKUSICK: You now have this thing called BigTable. Do you view that as an application in its own right?

QUINLAN: From the GFS point of view, it's an application, but it's clearly more of an infrastructure piece.

MCKUSICK: If I understand this correctly, BigTable is essentially a lightweight relational database.

QUINLAN: It's not really a relational database. I mean, we're not doing SQL and it doesn't really support joins and such. But BigTable is a structured storage system that lets you have lots of key-value pairs and a schema.

MCKUSICK: Who are the real clients of BigTable?

QUINLAN: BigTable is increasingly being used within Google for crawling and indexing systems, and we use it a lot within many of our client-facing applications. The truth of the matter is that there are tons of BigTable clients. Basically, any app with lots of small data items tends to use BigTable. That's especially true wherever there's fairly structured data.

MCKUSICK: I guess the question I'm really trying to pose here is: Did BigTable just get stuck into a lot of these applications as an attempt to deal with the small-file problem, basically by taking a whole bunch of small things and then aggregating them together?

QUINLAN: That has certainly been one use case for BigTable, but it was actually intended for a much more general sort of problem. If you're using BigTable in that way—that is, as a way of fighting the file-count problem where you might have otherwise used a file system to handle that—then you would not end up employing all of BigTable's functionality by any means. BigTable isn't really ideal for that purpose in that it requires resources for its own operations that are nontrivial. Also, it has a garbage-collection policy that's not super-aggressive, so that might not be the most efficient way to use your space. I'd say that the people who have been using BigTable purely to deal with the file-count problem probably haven't been terribly happy, but there's no question that it is one way for people to handle that problem.

MCKUSICK: What I've read about GFS seems to suggest that the idea was to have only two basic data structures: logs and SSTables (Sorted String Tables). Since I'm guessing the SSTables must

be used to handle key-value pairs and that sort of thing, how is that different from BigTable?

QUINLAN: The main difference is that SSTables are immutable, while BigTable provides mutable key value storage, and a whole lot more. BigTable itself is actually built on top of logs and SSTables. Initially, it stores incoming data into transaction log files. Then it gets *compacted*—as we call it—into a series of SSTables, which in turn get compacted together over time. In some respects, it's reminiscent of a log-structure file system. Anyway, as you've observed, logs and SSTables do seem to be the two data structures underlying the way we structure most of our data. We have log files for mutable stuff as it's being recorded. Then, once you have enough of that, you sort it and put it into this structure that has an index.

Even though GFS does not provide a Posix interface, it still has a pretty generic file-system interface, so people are essentially free to write any sort of data they like. It's just that, over time, the majority of our users have ended up using these two data structures. We also have something called *protocol buffers*, which is our data description language. The majority of data ends up being protocol buffers in these two structures.

Both provide for compression and checksums. Even though there are some people internally who end up reinventing these things, most people are content just to use those two basic building blocks.

Because GFS was designed initially to enable a crawling and indexing system, throughput was everything. In fact, the original paper written about the system makes this quite explicit: "High sustained bandwidth is more important than low latency. Most of our target applications place a premium on processing data in bulk at a high rate, while few have stringent response-time requirements for an individual read and write."

But then Google either developed or embraced many user-facing Internet services for which this is most definitely not the case.

One GFS shortcoming that this immediately exposed had to do with the original single-master design. A single point of failure may not have been a disaster for batch-oriented applications, but it was certainly unacceptable for latency-sensitive applications, such as video serving. The later addition of automated failover capabilities helped, but even then service could be out for up to a minute.

The other major challenge for GFS, of course, has revolved around finding ways to build latency-sensitive applications on top of a file system designed around an entirely different set of priorities.

MCKUSICK: It's well documented that the initial emphasis in designing GFS was on batch efficiency as opposed to low latency. Now that has come back to cause you trouble, particularly in terms of handling things such as videos. How are you handling that?

QUINLAN: The GFS design model from the get-go was all about achieving throughput, not about the latency at which that might be achieved. To give you a concrete example, if you're writing a file, it will typically be written in triplicate—meaning you'll actually be writing to three chunkservers. Should one of those chunkservers die or hiccup for a long period of time, the GFS master will notice the problem and schedule what we call a *pullchunk*, which means it will basically replicate one of those chunks. That will get you back up to three copies, and then the system will pass control back to the client, which will continue writing.

When we do a pullchunk we limit it to something on the order of 5MB–10MB a second. So, for 64MB, you're talking about 10 seconds for this recovery to take place. There are lots of other things like this that might take 10 seconds to a minute, which works just fine for batch-type operations. If you're doing a large MapReduce operation, you're OK just so long as one of the items is not a real straggler, in which case you've got yourself a different sort of problem. Still, generally speaking, a hiccup on the order of a minute over the course of an hour-long batch job doesn't really show up. If you are working on Gmail, however, and you're trying to write a mutation that represents some user action, then getting stuck for a minute is really going to mess you up.

We've had similar issues with our master failover. Initially, GFS had no provision for automatic master failover. It was a manual process. Although it didn't happen a lot, whenever it did, the cell might be down for an hour. Even our initial master-failover implementation required on the order of minutes. Over the past year, however, we've taken that down to something on the order of tens of seconds.

MCKUSICK: Still, for user-facing applications, that's not acceptable.

QUINLAN: Right. While these instances—where you have to provide for failover and error recovery—may have been acceptable in the batch situation, they're definitely not OK from a latency point of view for a user-facing application. Another issue here is that there are places in the design where we've tried to optimize for throughput by dumping thousands of operations into a queue and then just processing through them. That leads to fine throughput, but it's not great for latency. You can easily get into situations where you might be stuck for seconds at a time in a queue just waiting to get to the head of the queue.

Our user base has definitely migrated from being a MapReduce-based world to more of an interactive world that relies on things such as BigTable. Gmail is an obvious example of that. Videos aren't quite as bad where GFS is concerned because you get to stream data, meaning you can buffer. Still, trying to build an interactive database on top of a file system that was designed from the start to support more batch-oriented operations has certainly proved to be a pain point.

MCKUSICK: How exactly have you managed to deal with that?

QUINLAN: Within GFS, we've managed to improve things to a certain degree, mostly by designing the applications to deal with the problems that come up. Take BigTable as a good concrete example. The BigTable transaction log is actually the biggest bottleneck for getting a transaction logged. In effect, we decided, "Well, we're going to see hiccups in these writes, so what we'll do is to have two logs open at any one time. Then we'll just basically merge the two. We'll write to one and if that gets stuck, we'll write to the other. We'll merge those logs once we do a replay—if we need to do a replay, that is." We tended to design our applications to function

like that—which is to say they basically try to hide that latency since they know the system underneath isn't really all that great.

The guys who built Gmail went to a multihomed model, so if one instance of your Gmail account got stuck, you would basically just get moved to another data center. Actually, that capability was needed anyway just to ensure availability. Still, part of the motivation was that they wanted to hide the GFS problems.

MCKUSICK: I think it's fair to say that, by moving to a distributed-master file system, you're definitely going to be able to attack some of those latency issues.

QUINLAN: That was certainly one of our design goals. Also, BigTable itself is a very failure-aware system that tries to respond to failures far more rapidly than we were able to before. Using that as our metadata storage helps with some of those latency issues as well.

The engineers who worked on the earliest versions of GFS weren't particularly shy about departing from traditional choices in file-system design whenever they felt the need to do so. It just so happens that the approach taken to consistency is one of the aspects of the system where this is particularly evident.

Part of this, of course, was driven by necessity. Since Google's plans rested largely on massive deployments of commodity hardware, failures and hardware-related faults were a given. Beyond that, according to the original GFS paper, there were a few compatibility issues. "Many of our disks claimed to the Linux driver that they supported a range of IDE protocol versions but in fact responded reliably only to the more recent ones. Since the protocol versions are very similar, these drives mostly worked but occasionally the mismatches would cause the drive and the kernel to disagree about the drive's state. This would corrupt data silently due to problems in the kernel. This problem motivated our use of checksums to detect data corruption."

That didn't mean just any checksumming, however, but instead rigorous end-to-end checksumming, with an eye to everything from disk corruption to TCP/IP corruption to machine backplane corruption.

Interestingly, for all that checksumming vigilance, the GFS engineering team also opted for an approach to consistency that's relatively loose by file-system standards. Basically, GFS simply accepts that there will be times when people will end up reading slightly stale data. Since GFS is used mostly as an append-only system as opposed to an overwriting system, this generally means those people might end up missing something that was appended to the end of the file after they'd already opened it. To the GFS designers, this seemed an acceptable cost (although it turns out that there are applications for which this proves problematic).

Also, as Gobioff explained, "The risk of stale data in certain circumstances is just inherent to a highly distributed architecture that doesn't ask the master to maintain all that much information. We definitely could have made things a lot tighter if we were willing to dump a lot more data into the master and then have it maintain more state. But that just really wasn't all that critical to us."

Perhaps an even more important issue here is that the engineers making this decision owned not just the file system but also the applications intended to run on the file system. According to Gobioff, "The thing is that we controlled both the horizontal and the vertical—the file system and the application. So we could be sure our applications would know what to expect from the file system. And we just decided to push some of the complexity out to the applications to let them deal with it."

Still, there are some at Google who wonder whether that was the right call if only because people can sometimes obtain different data in the course of reading a given file multiple times, which tends to be so strongly at odds with their whole notion of how data storage is supposed to work.

MCKUSICK: Let's talk about consistency. The issue seems to be that it presumably takes some amount of time to get everything fully written to all the replicas. I think you said something earlier to the effect that GFS essentially requires that this all be fully written before you can continue.

QUINLAN: That's correct.

MCKUSICK: If that's the case, then how can you possibly end up with things that aren't consistent?

QUINLAN: Client failures have a way of fouling things up. Basically, the model in GFS is that the client just continues to push the write until it succeeds. If the client ends up crashing in the middle of an operation, things are left in a bit of an indeterminate state.

Early on, that was sort of considered to be OK, but over time, we tightened the window for how long that inconsistency could be tolerated, and then we slowly continued to reduce that. Otherwise, whenever the data is in that inconsistent state, you may get different lengths for the file. That can lead to some confusion. We had to have some backdoor interfaces for checking the consistency of the file data in those instances. We also have something called RecordAppend, which is an interface designed for multiple writers to append to a log concurrently. There the consistency was designed to be very loose. In retrospect, that turned out to be a lot more painful than anyone expected.

MCKUSICK: What exactly was loose? If the primary replica picks what the offset is for each write and then makes sure that actually occurs; I don't see where the inconsistencies are going to come up.

QUINLAN: What happens is that the primary will try. It will pick an offset, it will do the writes, but then one of them won't actually get written. Then the primary might change, at which point it can pick a different offset. RecordAppend does not offer any replay protection either. You could end up getting the data multiple times in the file.

There were even situations where you could get the data in a different order. It might appear multiple times in one chunk replica, but not necessarily in all of them. If you were reading the file, you could discover the data in different ways at different times. At the record level, you could discover the records in different orders depending on which chunks you happened to be reading.

MCKUSICK: Was this done by design?

QUINLAN: At the time, it must have seemed like a good idea, but in retrospect I think the consensus is that it proved to be more painful than it was worth. It just doesn't meet the expectations people have of a file system, so they end up getting surprised. Then they had to figure out work-arounds.

MCKUSICK: In retrospect, how would you handle this differently?

QUINLAN: I think it makes more sense to have a single writer per file.

MCKUSICK: All right, but what happens when you have multiple people wanting to append to a log?

QUINLAN: You serialize the writes through a single process that can ensure the replicas are consistent.

MCKUSICK: There's also this business where you essentially snapshot a chunk. Presumably, that's something you use when you're essentially replacing a replica, or whenever some chunkserver goes down and you need to replace some of its files.

QUINLAN: Actually, two things are going on there. One, as you suggest, is the recovery mechanism, which definitely involves copying around replicas of the file. The way that works in GFS is we basically revoke the lock so the client can't write it anymore, and this is part of that latency issue we were talking about.

There's also a separate issue, which is to support the snapshot feature of GFS. GFS has the most general-purpose snapshot capability you can imagine. You could snapshot any directory somewhere, and then both copies would be entirely equivalent. They would share the unchanged data. You could change either one and you could further snapshot either one. So it was really more of a clone than what most people think of as a snapshot. It's an interesting thing, but it makes for difficulties—especially as you try to build more distributed systems and you want potentially to snapshot larger chunks of the file tree.

I also think it's interesting that the snapshot feature hasn't been used more since it's actually a very powerful feature. That is, from a file-system point of view, it really offers a pretty nice piece of functionality. But putting snapshots into file systems, as I'm sure you know, is a real pain.

MCKUSICK: I know. I've done it. It's excruciating—especially in an overwriting file system.

QUINLAN: Exactly. This is a case where we didn't cheat, but from an implementation perspective, it's hard to create true snapshots. Still, it seems that in this case, going the full deal was the right decision. Just the same, it's an interesting contrast to some of the other decisions that were made early on in terms of the semantics.

All in all, the report card on GFS nearly 10 years later seems positive. There have been problems and shortcomings, to be sure, but there's surely no arguing with Google's success and GFS has without a doubt played an important role in that. What's more, its staying power has been nothing short of remarkable given that Google's operations have scaled orders of magnitude beyond anything the system had been designed to handle, while the application mix Google currently supports is not one that anyone could have possibly imagined back in the late 1990s.

Still, there's no question that GFS faces many challenges now. For one thing, the awkwardness of supporting an ever-growing fleet of user-facing, latency-sensitive applications on top of a system initially designed for batch-system throughput is something that's obvious to all.

The advent of BigTable has helped somewhat in this regard. As it turns out, however, BigTable isn't actually all that great a fit for GFS. In fact, it just makes the bottleneck limitations of the system's single-master design more apparent than would otherwise be the case.

For these and other reasons, engineers at Google have been working for much of the past two years on a new distributed master system designed to take full advantage of BigTable to attack some of those problems that have proved particularly difficult for GFS.

Accordingly, it now seems that beyond all the adjustments made to ensure the continued survival of GFS, the newest branch on the evolutionary tree will continue to grow in significance over the years to come. C

Q Related articles on queue.acm.org

A Conversation with Jeff Bonwick and Bill Moore
http://queue.acm.org/detail.cfm?id=1317400

The Five-Minute Rule 20 Years Later: and How Flash Memory Changes the Rules
Goetz Graefe
http://queue.acm.org/detail.cfm?id=1413264

Standardizing Storage Clusters
Garth Goodson, Sai Susharla, Rahul Iyer
http://queue.acm.org/detail.cfm?id=1317402

DOI:10.1145/1666420.1666438

Article development led by acmqueue
queue.acm.org

What will it take to make server-side computing more energy efficient?

BY DAVID J. BROWN AND CHARLES REAMS

Toward Energy-Efficient Computing

BY NOW, MOST everyone is aware of the energy problem at its highest level—our primary sources of energy are running out, while the demand for energy in both commercial and domestic environments is increasing. Moreover, the side effects of energy use have important global environmental considerations. The emission of greenhouse gases such as CO_2, now seen by most climatologists to be linked to global warming, is only one issue.

The world's preeminent scientists and thought leaders are perhaps most focused on a strategic solution: the need to develop new sources of clean and renewable energy if we are ultimately to overcome the energy problem. Lord Rees, president of the Royal Society, emphasized its urgency in an annual address delivered in 2008.[13]

The practical expectation of new sources of sustainable energy is at least three decades away, however. Steve Chu, who was the director of the Lawrence Berkeley National Laboratory prior to his recent appointment as U.S. Secretary of Energy, placed this situation in context:[3]

"A dual strategy is needed to solve the energy problem: (1) maximize energy efficiency and decrease energy use; (2) develop new sources of clean energy. No. 1 will remain the lowest-hanging fruit for the next few decades."

What part does computer equipment play in the demand for energy, and where must we focus to reduce consumption and improve energy efficiency?

In August 2007, the Environmental Protection Agency (EPA) issued a report to Congress on energy efficiency of servers and data centers.[5] Some key findings from the report include:

- Servers and data centers consumed 61 billion kWh (kilowatt hours) in 2006. This was 1.5% of total U.S. electricity consumption that year, amounting to $4.5 billion in electricity costs—equivalent to 5.8 million average U.S. households.
- Electricity use in this sector doubled between 2000 and 2006, a trend that is expected to continue.
- Infrastructure systems necessary to support the operation of IT equipment (for example, power delivery and cooling systems) also consumed a significant amount of energy, comprising 50% of annual electricity use.

Excerpts from the EPA report are shown in the accompanying figure and table. There are two particularly notable points in the data. The first is that as much energy is being consumed by site infrastructure as by the computing equipment itself. This infrastructure primarily represents heating, ventilation, and air-conditioning (HVAC) equipment, as well as that used to convert and transmit power and to maintain its continuity (the latter includes transformers

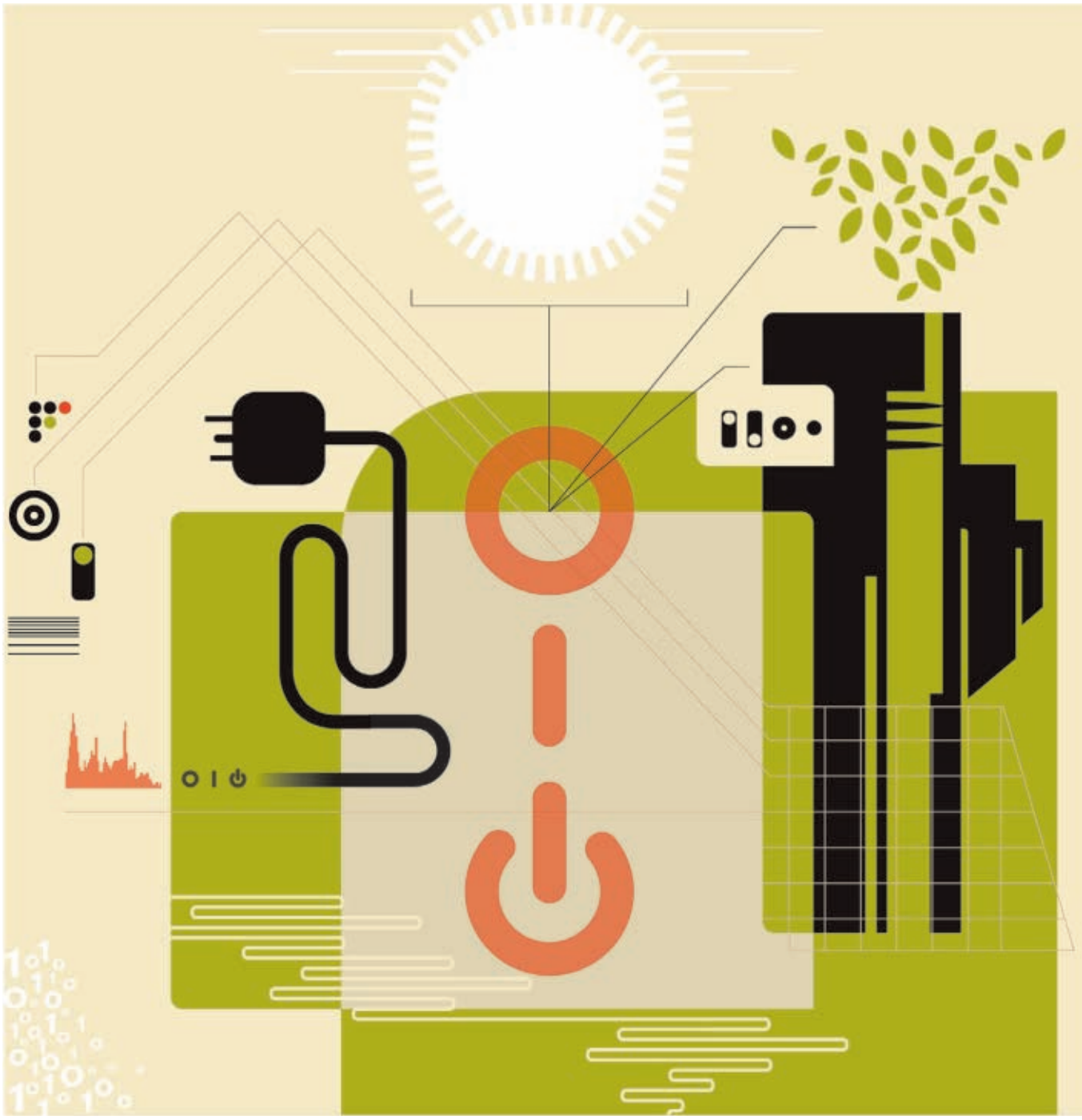


ILLUSTRATION BY CELIA JOHNSON

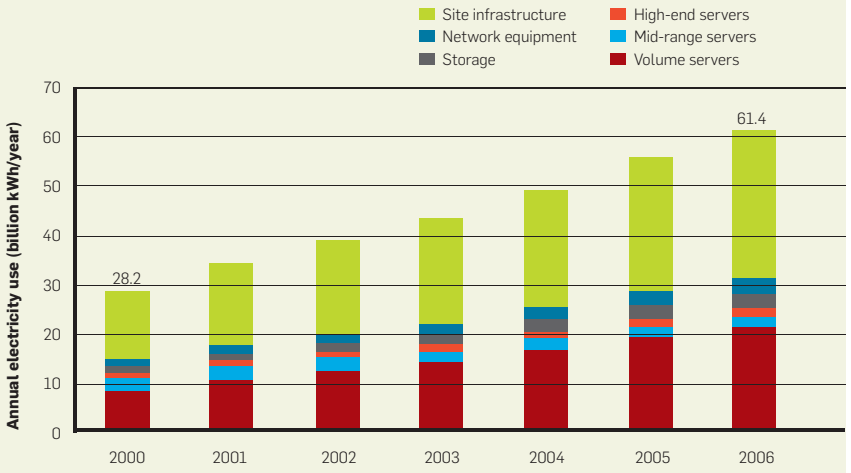
and in-building power-switching and transmission equipment, as well as power-conditioning and sustaining equipment such as uninterruptible power supplies). This factor is of great consequence, but may not be the most obvious domain for computing professionals to address.

Within the computing equipment itself, however, is the second point of interest. Of the five types of IT equipment studied, volume servers alone were responsible for the majority (68%) of the electricity used. Assuming that the 17% CAGR (combined annual growth rate) of volume servers continues, this suggests that they are the prime targets for energy reduction in the server space. The 20% growth rate of storage devices shown here—a rate that more recent data suggests is accelerating—indicates another significant trend.

If the exponential growth of data-center computing equipment revealed by this study continues, roughly double the demand for electricity seen in 2006 is expected in data centers by 2011. This poses challenges beyond the obvious economic ones. For example, peak instantaneous demand is expected to rise from 7GW (gigawatts) in 2006 to 12GW in 2011, and 10 new base-level power plants would be needed to meet such a demand.

Physical limitations on power availability are already a constraint for data centers in some areas; a managing director of IT for Morgan Stanley recently observed that the company is no longer able physically to get the power needed to run a new data

Electricity use by end-use component—2000 to 2006.
Source: EPA report to Congress on server and data center energy efficiency[5]



Electricity use by end-use component—2000 to 2006.
Source: EPA report to Congress on server and data center energy efficiency[5]

	2000		2006		2000–2006
End use component	Electricity use (billion kWh)	% Total	Electricity use (billion kWH)	% Total	Electricity use CAGR
Site infrastructure	14.1	50%	30.7	50%	14%
Network equipment	1.4	5%	3.0	5%	14%
Storage	1.1	4%	3.2	5%	20%
High-end servers	1.1	4%	1.5	2%	5%
Mid-range servers	2.5	9%	2.2	4%	–2%
Volume Servers	8.0	29%	20.9	34%	17%
Total	28.2		61.4		14%

center in Manhattan. The situation is serious. Corporations such as eBay, Google, Amazon, Microsoft, and Yahoo have pursued suitable locations where the data centers required to run their contemporary Web applications and services can be constructed.[9] A number of these companies have already negotiated with certain states in the U.S., as well as internationally, to construct these facilities, along with the power plants necessary to supply them. A few years ago Google touched off what some journalists deemed "a modern-day arms race" when it situated a new facility along the Columbia River in Washington. The combined benefits of lower land cost, lower external ambient temperature, and the availability of running water for cooling and hydroelectric power generation could provide an antidote both to Google's acute energy-availability problems and its cost.

There is some evidence[a] that the amount of energy consumed by mobile and desktop computing equipment is of roughly the same magnitude as that used by servers in data centers, although we do not have a correspondingly comprehensive and authoritative current study to refer to. The EPA data presented here provides some detailed perspective on where the energy goes in the important and growing server segment of the computing landscape. Also, some foundation has already been laid in the mobile and desktop computing space as a result of the earlier focus of the EPA's EnergyStar program on consumer electronics, which includes computer systems.

a The U.S. Energy Information Administration (www.eia.doe.gov) showed a figure of 23.1 terawatt hours per year consumed by PCs and printers within U.S. households in 2001.[4] The figures were similar in 2006.[14]

Power and its Management in Computer Systems Today

Perhaps the key factor to consider with today's computer systems is that the amount of power they consume does not adjust gracefully according to the amount of work the system is doing. The principal design objective for most general-purpose computer systems to date has been to maximize performance—or, perhaps, performance at a given price point—with very little consideration given to energy use. This is changing rapidly as we near the point where the capital cost to acquire computing equipment will be exceeded by the cost of energy to operate it—even over its relatively short (3- to 5-year) amortization period—unless we pay some attention to energy-conscious system design.

Although the case has been made for energy-proportional computing[2]—meaning the amount of power required corresponds directly to a system's (or component's) degree of utilization—this is far from the current situation. Many components of computer systems today exhibit particularly poor efficiencies at low levels of utilization, and most systems spend a great proportion of their time operating at relatively low-usage levels. Power supplies have been notorious for their inefficiency, especially when at low load, and fans can waste much energy when operated carelessly. In just four years, however, the efficiency of power supplies has improved.[1] Indeed, algorithms that adjust fan speeds more continuously in relation to thermal need, rather than using just a few discrete speed points, are emerging. The majority of hardware components in today's computer systems must still be managed explicitly, however, and the current widely deployed conceptions and facilities for power management in computer systems remain rudimentary.

Power Management

There are two basic modalities for power management: a running vs. suspended (not-running) aspect in which a component (or whole system)

can be powered off when it is not being used (that is, once it has become idle), but turned on again when it is needed; and a performance-adjustment aspect (while running) in which the performance level of a component can be lowered or raised, based on either the observed level of its utilization or other needs of the workload.

The running versus not-running choices is often called the component's (or system's) *power states*. While there is a single state to represent running, there may be more than one suspended state. The latter allows power to be removed from progressively more of the hardware associated with the component (or system) if there is some important power-relevant structure to its implementation. CPUs, for example, may have their execution suspended simply by stopping the issuance of instructions or by turning off their clock circuitry. "Deeper" power states, however, might successively remove power from the processor's caches, TLBs (translation lookaside buffers), memory controllers, and so on. While more energy is saved as more of a component's hardware has its power removed, there is then either a greater latency to recommence its operation, or extra energy is required to save and restore the hardware's contents and restart it, or both.

The performance-adjustment choices while running are most naturally called the component's *performance states*. A widely applied technique for adjusting performance is to change the component's operating frequency. When clock speed is slowed, operating voltage levels can also be reduced, and these two factors together—normally called DVFS (dynamic voltage and frequency scaling)—result in a compound power savings. Performance states were first introduced for CPUs, since processors are among the most consequential consumers of power on the hardware platform (something in the range of 35W (watts) to 165W is typical of a contemporary multicore CPU). Performance states might also be used to control the active cache size, the number and/or operating rates of memory and I/O interconnects, and the like.

ACPI

The most widely implemented architecture for power management is the Advanced Configuration and Power Interface (ACPI). It has evolved together with Intel architecture, the hardware platforms based on the most widely available commodity CPUs and related components. Although there are many detailed aspects to the specification, ACPI principally offers the controls needed to implement the two power-management modalities just described. It defines power states: seven at the whole-system level, called S-states (S0-S6); and four at the per-device level called D-states (D0-D3).[b] The zero-numbered state (S0 for the system, or D0 for each device) indicates the running (or active) state, while the higher-numbered ones are nonrunning (inactive) states with successively lower power—and correspondingly decreasing levels of availability (runreadiness). ACPI also defines performance states, called P-states (P0-P15, allowing a maximum of 16 per device), which affect the component's operational performance while running. Both affect power consumption.

Energy Efficiency in Computing

Although ACPI is an important de facto standard with reasonably broad support from manufacturers, it provides only a mechanism by which aspects of the system can be controlled to affect their power consumption. This enables but does not explicitly provide energy efficiency. Higher-level aspects of the overall system architecture are needed to exploit this or any similar mechanism.

How does energy-efficient computing differ from power management, and how would you know you had solved the energy-efficiency problem for a computer system? Here is a simple vision: *"The system consumes the minimum amount of energy[c] required to perform any task."*

In other words, energy efficiency is an optimization problem. Such a system must adjust the system's hardware resources dynamically, so that only what is needed to perform those tasks (whether to complete them on time, or analogously, to provide the throughput required to maintain a stated service level) is made available, and that the total energy used is minimized as a result.

Traditionally, systems have been designed to achieve maximum performance for the workload. On energy-efficient systems, maximum performance for some tasks (or the whole workload) will still be desired in some cases, but the system must now also minimize energy use. It is important to understand that performance and energy efficiency are not mutually exclusive. For example, even when achieving maximum performance, any resources that can be deactivated, or whose individual performance can be reduced without affecting the workload's best possible completion time or throughput, constitute energy optimization.

Indeed, there are few (if any) situations in which the full capacity of the hardware resources (that is, all operating at their peak performance levels) on any system is exploited. Systems that strive to achieve maximum performance at all times are notoriously over-provisioned (and correspondingly underutilized). People involved in practical computer system design may note that our science is weak in this area, however. (This area might be called "dynamic capacity planning and dynamic provisioning.")

Energy optimization is obviously subject to certain constraints. Some examples follow.

Required Performance Levels Must be Maintained

Tasks with deadlines must be completed on time. In the general case, a deadline is specified for a task or the workload. When any deadline is specified that is less than or equal to the optimum that the system can achieve with any or all of its hardware resources, this implies maximum performance. This is effectively the degenerate case.

Maximum performance for a task or the workload provides an implicit stipulation of the optimal deadline (to), or

b Idiosyncratically, the power states for CPUs are called C-states (C0-C3). In any case, the semantics of each nonrunning power state is specific to the device (or device class) in question.

c Energy is the time integral of power, so that for constant power, energy = power × time. Power and energy are different concepts and should not be confused.

"as soon as possible."[d] In this case, energy optimization is restricted to those resources that can be deactivated, or whose individual performance can be reduced, without affecting the workload's best possible completion time.

If a deadline later than the best achievable deadline is specified, the computation may take any length of time up to this deadline, and the system can seek a more global energy minimum for the task (or workload). Deadlines might be considered "hard," in which case the system's energy-optimizing resource allocator must somehow guarantee to meet them (raising difficult implementation issues), or "soft," in which case only a best effort can be tolerated.

Services must operate at required throughput. For online services, the notion of throughput, in order to characterize the required performance level, may be more suitable than that of a completion deadline. Since services, in their implementation, can ultimately be decomposed into individual tasks that do complete, we expect there to be a technical analog (although the most suitable means of specifying its performance constraint might be different).

The System Must be Responsive to Changing Demand

Real workloads are not static: the amount of work provided and the resources required to achieve a given performance level will vary as they run. Dynamic response is an important practical consideration related to service level.

Throughput (T) must be achievable within latency (L). Specification of the maximum latency within which reserved hardware capacity can be activated or its performance level increased seems a clear requirement, but this must also be related to the performance needs of the task or workload in question.

Throughput is dependent on the type of task. A metric such as TPS (transactions per second) might be relevant for database system operation, triangles per second for the rendering component of an image-generation subsystem, or corresponding measures for a filing service, I/O interconnect, or network interface. Interactive use imposes real-time responsiveness criteria, as does media delivery: computational, storage, and I/O capacity required to meet required audio and video delivery rates. A means by which such diverse throughput requirements might be handled in practice is suggested here.

Instantaneous power must never exceed power limit (P). A maximum power limit may be specified to respect practical limits on power availability (whether to an individual system or to a data center as a whole). In some cases, exceeding this limit briefly may be permissible.

Combinations of such constraints mean that over-constraint must be expected in some circumstances, and therefore a policy for constraint relaxation will also be required. A strict precedence of the constraints might be chosen or a more complex trade-off made between them.

Approaching a Solution

Given this concept for energy-efficient computing, how might such a system be constructed? How would you expect an energy-efficient system to operate?

A system has three principal aspects that could solve this problem:

- It must be able to construct a power model that allows the system to know how and where power is consumed, and how it can manipulate that power (this component is the basis for enacting any form of power management).
- The system must have a means for determining the performance requirements of tasks or the workload—whether by observation or by some more explicit means of communication. This is the constraints-determination and performance-assessment component.
- Finally, the system must implement an *energy optimizer*—a means of deciding an energy-efficient configuration of the hardware at all times while operating. That optimization may be relative (heuristically decided) or absolute (based on analytical techniques). This is the capacity-planning and dynamic-provisioning component.

The first aspect is relatively straightforward to construct. The third is certainly immediately approachable, especially where the optimization technique(s) are based on heuristic methods. The second consideration is the most daunting. It represents an important disruptive consequence of energy-efficient computing and could demand a more formal (programmatic) basis for communicating requirements of the workload to the system. A description of the workload's basic provisioning needs, along with a way to indicate both its performance requirements and present performance, seem basic to this.

A way of indicating a priori its expected sensitivity to changes in provisioning of various system resources could also be useful. Fortunately, there are a number of practical approaches to energy efficiency to pursue prior to the refinements enabled by the hoped-for developments in category 2.

Power Model

In order to manage the system's hardware for energy efficiency, the system[e] must know the specific power details of the physical devices under its control. Power-manageable components must expose the controls that they offer, such as their power and performance states (D-states and P-states, respectively, in the ACPI architectural model). To allow modeling of power relative to performance and availability (that is, relative to its activation responsiveness), however, the component interface must also describe at least the following:

- The per-state power consumption (for each inactive state) or power range (for each active state).
- State-transition latency (time required to make each state transition).
- State-change energy (energy expended to change state).

Once the system has such a power model, consisting of all its power-manageable hardware, it has the basic foun-

d All values of deadline: $D = t_i$ less than the shortest achievable deadline: to is equivalent to setting $D = t_o$ (that is: $\{\forall t\, t_i < t_o, [D = t_i] \approx [D = t_o]\}$). We can therefore denote maximum performance by $D = 0$.

e "The system" here most naturally suggests the operating system, although it is clear that this must include the hypervisor for virtualized systems. One can reasonably expect that this concept will need to be broadened to include some aspects of the firmware and even hardware components (on the low end) and important runtimes, such as the Java Virtual Machine, which have responsibility for, and/or particular knowledge of, resource allocation.

dation for operating to optimize energy. Importantly, it has the knowledge of those components that consume the most power and those that have the most highly responsive controls that can be used to affect power use.

Workload Constraints and Performance Assessment

In its desire to limit the amount of active hardware and reduce its performance so as to minimize energy consumption, how is a system to know whether the tasks being run are still achieving enough throughput to maintain appropriate service levels or realize their deadlines?

The assessment of throughput is subject to the task or application in question. The operating system can observe the degree to which its various resources have been and are currently being used, and it might use these observations as its best basis for prediction of future resource needs—thus shrinking or enlarging what is available. This is a relatively weak basis to determine what the workload will need, especially to anticipate its dynamic responsiveness sensitivities. As a result, the system will have to be much more conservative about its reduction of available resources or their performance levels. It seems clear that the best result will be realized if applications assess their own throughput relative to their service-level requirements or completion deadlines, and can convey that information to the operating system through an interface. The system can then use this information to make potentially much more aggressive resource adjustments and realize an improved overall energy-optimization solution accordingly.

Here is the crucial dichotomy: the system is responsible for solving the energy-optimization problem subject to the resources it allocates, while the application is responsible for monitoring its own performance level and informing the system so that appropriate resources can be provided to meet them.

Energy Optimization by the System

Once provided with the hardware's power characteristics, and possibly descriptive information from application-level software about its constraints, the operating system must begin the dynamic process of adjusting the hardware's performance and availability levels to control power consumption and improve systemwide energy use. How can the operating system make such decisions?

The system consumes the minimum amount of energy required to perform any task.

Heuristic methods. Provisioning for maximum throughput may, in some cases, optimize energy. This is the conjecture that "[maximum] performance is green," reflected in the ideas of race-to-idle or race-to-sleep.[8] Although there is some evidence that this approach has merit in client-side computing when the system becomes idle—especially for embedded and mobile systems where 95% of the energy may be saved if the entire system can be put in a suspended state—it is not clear how applicable this may be for server-side computing. A nonlinear increase in the power required to get linear speed-up (throughput) exists in some cases—Intel's Turbo mode on contemporary CPUs is one example—and hence, the energy optimum will not be found at a provisioning and performance point commensurate with maximum throughput in all cases.

A widely used heuristic for energy improvement on active systems is to adjust the hardware's performance level dynamically, based on its current utilization: downward with low utilization or upward with high utilization (utilization below or above some threshold for some duration). This can be an effective technique but is restricted to situations in which both the latency and energy to make the state change are so low as to be inconsequential.

Constraints-based optimization as an approach. In some cases, it may be possible to simplify the problem to such a degree to provide a complete analytical solution. For example, if we consider only a single task on a single CPU with a well-understood power/performance trade-off, it is relatively straightforward to specify completely a schedule in which the task will meet its deadline with the minimum total energy; more general formal results are also possible.[12] This relies, however, on a number of assumptions, such as good estimates of the total work required by a process, which frequently do not hold up in practice. Weaker assumptions require online optimization algorithms

to perform energy-aware scheduling. There is some existing work in this area but not yet enough to underpin a general-purpose operating system.[17]

For an optimization-based approach to be generally applicable, a range of techniques will be necessary. In the simplest cases, autonomous device-level operation is possible; for example, at the hardware level, a GPU can power down unused hardware pipelines aggressively, based solely on instantaneous assessment of their utilization levels, because the latency to bring those pipelines back up as they become necessary is inconsequential. Similar practices appear to be applicable in the use of CPU P-states (CPU performance and energy-cost adjustment based on voltage and frequency scaling), since both the state-transition energy and latency are very low.

Hardware state changes that affect power but exhibit a much greater latency and/or a much greater amount of energy to make the state change require a different treatment. An obvious example is spinning down a hard disk, considering the long latency to return it to running, but reactivation latency is not the only concern. Semiconductor memory systems in which part of the total physical memory could be powered off if not required, and where power-on latency may be near zero, will still have a consequential transition energy, since a great many in-memory transactions may be required to gather the working set into those physical pages that will remain active.[f] Resources of this class require greater knowledge of the task or workload behavior, as well as an anticipatory treatment of the required hardware resources, to ensure that the activation latency can be tolerated or managed and that the state-change energy will be exceeded by the energy that will be saved while in that state.

Some common optimization techniques may be based on state-change latency, their energy demands, and so on, and a taxonomy of such techniques might arise from this—some formal or analytical, some based on more numerical or heuristic methods.

Systems must be revised to pay attention to their use of energy; the operating system itself, which is always running, has not yet been optimized in its own use of energy.

Although we expect the specific techniques for energy optimization appropriate to different hardware resources or subsystems to be somewhat different, subject to the properties of the hardware resources in question, the hope is that the composition of energy-efficiency optimizers for all such resources will accumulate to form an efficiency scheme for the whole system.[g]

Getting There

The vision of systemwide concessions to energy efficiency cannot be accomplished in a single swift step. Today's systems software is not equipped in the ways described, nor are applications written in a way that could exploit that capability. In pragmatic terms, how do we expect this outcome to be achieved, and what steps are already under way?

As a first consideration, systems must be revised to pay attention to their use of energy; the operating system itself, which is always running, has not yet been optimized in its own use of energy. To date, almost all software, including systems software, has been optimized for performance, robustness, and scalability with no consideration of energy. An initial step, therefore, is the redesign and implementation of the operating system so that its operation is energy efficient. This is a significant undertaking, and its full implications are not yet well understood.

It is not clear whether modifying existing operating systems to consider energy as a first-class constraint is feasible, although this would certainly be preferable. Experience with system security shows that attempts to introduce such fundamental considerations after the fact are fraught with complications. We can certainly anticipate fundamental new structures within systems software, and perhaps even that new operating systems will emerge as a result of the energy-efficiency pressure.

At the very least, resource-management facilities within the operating sys-

f It is interesting to consider whether traditional heuristics such as the Five-minute Rule, designed to optimize the memory hierarchy for performance, might have analogues in energy optimization.

g We recognize that such reductionism may be overly optimistic if there are interactions between the resources allocated by different subsystems, and that a more holistic approach (e.g., a large dynamic-programming approach) may then be necessary in systems where "every joule counts."

tem must be adapted for energy awareness, and then for energy optimization.

Processors. Given the significant fraction of power on contemporary computing platforms attributed to CPUs (and the early introduction of power-management features on them as a result), much progress has already been made with operating-system schedulers/thread dispatchers. Careless activation of hardware when there is no useful work to be done must be eliminated. Polling within the operating system (or within applications) is an obvious example, but the use of a high-frequency clock-tick interrupt as the basis for timer events, timekeeping, and thread-scheduling can be equally problematic. The objective is to keep hardware quiescent until needed. The "tickless" kernel project[16] in Linux introduced an initial implementation of a dynamic tick. By reprogramming the per-CPU periodic timer interrupt to eliminate clock ticks during idle, the average amount of time that a CPU stays in its idle state after each idle state entry can be improved by a factor of 10 or more. Beyond the very good ideas that dynamic ticks and deferrable timers in Linux represent, the Tesla project in OpenSolaris is also considering what the transition to a more broadly event-based scheme for software development within the operating system might imply.

The confluence of features on modern processors—CMT (chip multithreading), CMP (chip multiprocessor), and NUMA (non-uniform memory access) for multiprocessor systems with multiple sockets—invites a great deal of new work to implement optimal-placement thread schedulers.[6] Given the ability to alter performance levels, energy efficiency and the expected introduction of heterogeneous multicore CPUs[h] will only add intrigue to this.[7,15]

Storage. Compared with CPUs, the power consumed by a disk drive does not seem especially large. A typical 3.5-in., 7200RPM commodity disk consumes about 7W to 8W—only about 10% of what a typical multicore CPU consumes. Although higher-performance 10,000RPM spindles consume about 14W, and 15,000RPM drives perhaps use around 20W, what is the worry? The alarming relative rate of growth in storage, mentioned earlier, could quickly change the percentage of total power that storage devices account for. Performance and reliability factors have already resulted in the common application of multiple spindles, even on desktop systems (to implement a simple RAID solution). In the data center, storage solutions are scaling up much faster.

Low-end volume server boxes now routinely house a dozen or more drives, and one example 4U rack-mount storage array product from Sun accommodates 46 3.5-in. drives. A single instance of the latter unit, if it used 10,000RPM- or 15,000RPM industrial drives, might therefore account for 1,088W to 1.6kW, rather a more significant energy-use picture.

Storage subsystems are now obviously on the radar of the energy attentive. There are at least two immediate steps that can be taken to help improve energy consumption by storage devices. The first is direct attention to energy use in traditional disk-based storage. Some of this work has been started by the disk hardware vendors, who are beginning to introduce disk-drive power states, and some have been started by operating-system developers working on contemporary file systems (such as ZFS) and storage resource management. The second, particularly derived from the recent introduction of large inexpensive Flash memory devices, is a more holistic look at the memory/storage hierarchy. Flash memory fills an important performance/capacity gap between main memory devices and disks,[10,11] but also has tremendous energy-efficiency advantages over rotating mechanical media.

Memory. Main memory, because of its relatively low power requirement (say, 2W per DIMM), seems at first glance to be of even less concern than disks. Its average size on contemporary hardware platforms, however, may be poised to grow more rapidly. With hardware system manufacturers' focus primarily on performance levels (to keep up with the corresponding performance demands of multicore CPUs), maintaining full CPU-to-memory bandwidth is critical. The consequence has been an evolution from single- to dual-channel and now triple-channel DIMMs along with the corresponding DDR, DDR2, and DDR3 SDRAM technologies. Although reductions in the process feature size (DDR3 is now on 50-nanometer technology) have enabled clock frequency to go up and power per DIMM to go down somewhat, the desire for even greater performance via an increase in DIMMs per memory channel is still increasing the total power consumed by the memory system.

For example, a current four-socket server system (based on the eight-core Sun Niagara2 CPU) with 16 DIMMs per socket using DDR2 dual-channel memory technology, has 64 DIMMs total. This would increase to 24 DIMMs per socket (96 total) if its faster successor used DDR3 triple-channel memory instead. A representative DDR2 DIMM consumes 1.65W (or 3.3W per pair), whereas the lowest-power edition of the current DDR3 DIMMs consume 1.3W (or 3.9W per trio). The result appears to be an increase of only 20% power consumption—from about 100W to 120W total in our example.

Given that the next-generation CPU will also have twice as many cores per socket, however, a possible scenario is also to desire twice the number of memory sets per socket (for a possible 192 total DIMMs) to balance overall memory system performance. The result, therefore, could be an increase from 100W to 240W (a 140% increase in power consumption for the whole memory system)! This trend is even being observed on desktop-class machines, admittedly at a much smaller scale, as systems containing quad-core hyperthreaded CPUs (such as Intel's Nehalem) have appeared.

If available physical memory is to be enabled and disabled, and perhaps correspondingly reconfigured as a system's processing capacity is dynamically adjusted, some new functionality will be required of the operating system's memory-management subsystem. The design of a future-looking virtual memory system that is energy aware and able to adjust physical memory resources while running is an open problem.

I/O. Energy aspects of the I/O system on hardware platforms will likely

h Heterogeneous here means a multicore CPU in which cores of different performance levels (different CPU microarchitectures) are put in the same multicore package, and whose power-consumption consequences are therefore very different.

become more important as well. As a simple example, present-day local-area networking interconnect and subsystems have evolved in two important respects: link-aggregation is increasingly used to bolster network bandwidth and reliability; and individual interconnect speed has advanced from 1GB to 10GB, with 40GB on the horizon. A transceiver for a 10GB network interface card may now require as much as 14W when operating at full speed, with a consequential power reduction when its link speed is reduced to 1GB or lower (about 3W at 1GB, 1W at 100MB). Other high-speed interconnects such as InfiniBand can be expected to have similar energy considerations for the overall system. Little attention has been given to the energy implications of communication interconnects in any of their various architectural manifestations, from on-chip to wide area networking.

The Evolution of Application Software

The most strategic aspect of energy-efficient computing will be the evolution of application software to facilitate systemwide energy efficiency. Although we can certainly expect new application interfaces to the system software supporting the development of new energy-efficient applications, the transition of historical and present-day applications represents a long-term evolution. How will we address the problem of greater energy efficiency for the remainder of the installed base in the interim? Obviously, it will not be brought about as the result of a unique epoch in the implementation of all existing applications.

One possibility for addressing the energy agnosticism of existing applications is to perform extrinsic analysis of their runtime behavior. Empirical data can be gathered about the degree to which application performance[i] is sensitive to varying levels and types of resource provisioning. For example, one can observe the degree to which performance is increased by the addition of CPU resources, or the allotment of a CPU with higher-performance microarchitecture, and so on.[15] The application might then be labeled, in its binary form, with its measured degree of sensitivity, without requiring the alteration of its existing implementation. The operating system could then use the data to assign resources that pursue a certain specified performance level or to locate an appropriate performance-versus-energy consumption trade-off.

Inevitably, we expect that a combination of techniques will be needed: both explicit, in which the application itself informs the system of its throughput and resource provisioning needs; and implicit, in which static and dynamic analysis is used to model resource needs relative to performance and energy consumption.

Conclusion

We are still at the debut of energy-conscious computing, with a great deal of the industry's attention being given to the introduction and use of power-management mechanisms and controls in individual hardware components rather than to the broader problem of energy efficiency: the minimization of total energy required to run computational workloads on a system. This article suggests an overall approach to energy efficiency in computing systems. It proposes the implementation of energy-optimization mechanisms within systems software, equipped with a power model for the system's hardware and informed by applications that suggest resource-provisioning adjustments so that they can achieve their required throughput levels and/or completion deadlines.

In the near term, a number of heuristic techniques designed to reduce the most obvious energy waste associated with the highest-power components, such as CPUs, are likely to remain practical. In the longer term, and for more effective total energy optimization, we believe that techniques able to model performance relative to the system's hardware configuration (and hence its energy consumption), along with an improved understanding and some predictive knowledge of workloads, will become increasingly important. C

i This assumes one can define some objective external metric of performance, which may be problematic.

Related articles on queue.acm.org

Power-Efficient Software
Eric Saxe
http://queue.acm.org/detail.cfm?id=1698225

Maximizing Power Efficiency with Asymmetric Multicore Systems
Alexandra Fedorova, Juan Carlos Saez, Daniel Shelepov, Manuel Prieto
http://queue.acm.org/detail.cfm?id=1658422

Powering Down
Matthew Garrett
http://queue.acm.org/detail.cfm?id=1331293

References

1. 80plus.org. Recent standards for power supply efficiency; http://www.80plus.org.
2. Barroso, L. and Holzle, U. The case for energy-proportional computing. *IEEE Computer* (Dec. 2007), 33-37.
3. Chu, S. The energy problem and Lawrence Berkeley National Laboratory. Talk given to the California Air Resources Board (Feb. 2008).
4. Energy Information Administration, U.S. Department of Energy. Residential Energy Consumption Surveys (2001); http://www.eia.doe.gov/emeu/recs/recs2001/enduse2001/enduse2001.html.
5. Environmental Protection Agency. EPA report to Congress on server and data center energy efficiency (Aug. 2007); http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf.
6. Fedorova, A. Operating system scheduling for chip multithreaded processors. Ph.D. dissertation. Harvard University (Sept. 2006).
7. Fedorova, A., Saez, J. C., Shelepov, D., and Prieto, M. Maximizing performance per watt with asymmetric multicore systems. *ACM Queue* (Nov./Dec. 2009); http://queue.acm.org/detail.cfm?id=1658422.
8. Garrett, M. Powering down. *ACM Queue* (Nov./Dec. 2007).
9. Katz, R. H. Tech titans building boom. *IEEE Spectrum* (Feb. 2009); http://www.spectrum.ieee.org/green-tech/buildings/.
10. Leventhal, A. Flash storage today. *ACM Queue 6*, 4 (July/Aug. 2008), 25-30, 25-30; http://queue.acm.org.
11. Mogul, J., Argollo, E., Shah, M., and Faraboschi, P. Operating system support for NVM+DRAM hybrid main memory. In *Proceedings of Usenix HotOS XII* (May 2009).
12. Reams, C. Energy-conscious computing—formal techniques for energy cost minimization (forthcoming paper).
13. Rees, M. Anniversary address to the Royal Society, London, 2008.
14. Roth, K. W. and McKenney, K. Energy consumption by consumer electronics in U.S. residences. TIAX LLC, Cambridge, MA (Jan. 2007).
15. Shelepov, D., Saez, J. C., Jeffery, S., Fedorova, A., Perez, N., Huang, Z. F., Blagodurov, S., and Kumar, V. 2009. HASS: a scheduler for heterogeneous multicore systems. *ACM Operating System Review 43*, 2 (2009), 66-75.
16. Siddha, S. Getting maximum mileage out of tickles. In *Proceedings of the Linux Symposium* (Ottawa, Ontario, June 2007), 201-208.
17. Yao, F., Demers, A., and Shenker, S. A scheduling model for reduced CPU energy. In *Proceedings of 36th Annual Symposium on Foundations of Computer Science* (1995), 374–382.

David Brown is currently working on the Solaris operating system's core power management facilities, with particular attention to Sun's x64 hardware platforms. Earlier at Sun he led the Solaris ABI program: a campaign to develop and deliver a practical approach to binary compatibility for applications built on Solaris.

Charles Reams is a Ph.D. student in computer science at the University of Cambridge. His research interests are focused on quantitative and emergent aspects of programming languages; he hopes to move his academic work on energy- and cost-efficient scheduling to the commercial world.

DOI:10.1145/1666420.1666449

To succeed on a global scale, businesses should focus on a trio of key elements.

BY SIEW KIEN SIA, CHRISTINA SOH, AND PETER WEILL

Global IT Management

Structuring for Scale, Responsiveness, and Innovation

GLOBALIZATION IS A significant factor in today's business strategies,[8] as companies in mature markets seek growth by expanding their operations in the emerging markets of Asia, Latin America, Eastern Europe, and the Middle East. These multinational companies (MNCs) have to extend their existing portfolio of IT applications, infrastructure, and services to support their global business strategies.

However, managing globally distributed IT resources is challenging. Visibility of such resources is often poor, as the local IT unit may not report back to central IT, and in many firms there is no enterprisewide IT budget management. For most firms there is also an inherent global-local tension to simultaneously achieve three strategic objectives: scale, responsiveness, and innovation. To balance the trade-offs, practice and research in the structural design of IT has moved away from the IT centralization-versus-decentralization debate to more nuanced forms of IT organizational design. These include the federal structure,[14] hybrid governance,[6] "centrally decentralized" governance,[17] and matrixed governance.[18]

These "hybrid" structures recognize that the various types of IT activities have different operating characteristics and economics and thus should be managed differently. Some researchers, for example, have found the management of IT infrastructure is usually centralized, while the management of IT use is often decentralized. The development of IT applications resides in the local units for some organizations, or at central IT for others, while a third group has applications development capabilities at both central and local units. Agarwal and Sambamuthy[1] noted three

variants—the partner, platform, and scalable model—where the decision rights for each of eight IT value processes (for example, infrastructure management, solutions delivery, and strategic planning) could be centralized, decentralized, or shared. Allocating decision rights differently for different IT activities was also at the heart of the matrix governance proposed by Weill and Ross[19]—who identified different configurations for making five key IT decisions—IT principles, IT architecture, IT infrastructure, business application needs, and IT investment and prioritization.

The features of hybrid structures remain under-studied, particularly as increasing globalization has resulted in continuing evolution of the structure of the IT function. Ineffective global IT structures result in the duplication of resources, proliferation of IT systems, increased complexity and risk, and the compromise of key business requirements such as agility. Here, we ask how are hybrid IT structures implemented in the global context to balance the global-local tensions while achieving scale, responsiveness, and innovation?

Structuring the Global IT Organizations

We examine this question through in-depth studies of four industry leading MNCs that have established a strong global presence, particularly in emerging markets such as Asia. The four companies represent a diverse set of industries. Microsoft develops, manufactures, licenses, and markets software in 90 countries. Intel is the world's largest producer of semiconductor chips and operates in 60 countries. Procter and Gamble is a leading manufacturer and marketer of consumer products in three sectors—beauty care, household care, and health and well being—across more than 180 countries. Underwood Financials (pseudonym), is among the top 10 investment banks globally, operating in 60 countries, and continues to perform relatively well even in the current economic downturn.

We interviewed between two and six executives (a few with multiple interviews) in each firm including the CIO, examined internal documents such as organization charts, and publicly available information such as annual reports, analyst reports, and news reports. Our interview questions were concerned with how these companies had set up and managed their global IT structures with a particular focus on the fast-growing Asian region.

Our findings showed that, despite the variation in industry, all the MNCs studied used three common structural elements to link the enterprisewide IT leadership (who design and oversee enterprisewide IT governance, the IT budget, and portfolio management, enterprise architecture, and enterprise risk management), and the more locally focused concerns of the business units. Although companies sometimes labeled these elements differently, such as, shared services, centers of excellence (CoEs), and value managers (VMs), the goals of each element were the same across the firms. The objective of shared services was to achieve scale economies; the objective of CoEs was to drive innovation; and the objective of value managers was to enable responsiveness. The three structural elements are described here in detail.

IT Shared Services are structural units that consolidate common IT functions (for example, helpdesk, operations, development) to achieve scale by providing standardized services. Such sharing eliminates unnecessary duplication of IT resources and improves utilization of IT assets. Global MNCs often have three shared service units located in the Americas, Europe, and Asia focused on delivery within their respective regions and serving as backups for the other regions. Microsoft, for example, created regional shared services at Richmond (corporate headquarters serving North America), Dublin (serving Europe, Middle East, and Latin America), and Singapore (serving Asia) to manage IT services across the globe.

Shared service units can offer a wide range of IT services, allowing the local business units to choose from

Table 1. Examples of P&G Global Business Services.

Employee Services and Solutions	
Employee Services	Pay, benefits, policies, career development, work plans
People Management	Compensation planning, relocation, employee management tools
Facilities	Office moves, conveniences: banking, dining, fitness centers, mail and documents
Computers and Communications	PCs, email, mobile phones, Intranet, service support
Meetings	Rooms, technology and scheduling, audio and video conferencing, events
Travel	Booking, expense accounting, credit cards, group meetings
Business Services and Solutions	
Strategic Sourcing and Procurement	Strategic sourcing, supplier relationship management, procurement service
Financial Services and Solutions	General ledger, affiliate accounting, product/fixed asset accounting, SRAP/MSA accounting, purchases-to-payment (include accounts payable), banking, financial reporting
Product Innovation	Bioinformatics systems, product imaging and modeling systems
Supply Network Solutions	Demand planning systems, total order management, physical distance systems
Consumer Solutions	Prime prospect research, CRM systems, advertising and media measurement
Customer Solutions	Shopper intelligence, in-store action planning, trade fund management systems
Initiative Management	Technical package and materials design, package artwork process, portfolio tracking, and reporting
Business Performance Solutions	Decision cockpits, market mix modeling, competitive intelligence, ad hoc business analyses

a catalog of IT services. The global-local tension here is to encourage local units to use more of the shared services while still meeting the diverse needs of the local units. For example, as a $90 billion global enterprise operating in more than 180 countries and marketing over 250 brands to nearly five billion consumers, P&G created the Global Business Services (GBS) unit in 1999. GBS provides a set of 70 IT services on a global scale with published IT unit costs and service-level agreements. To provide around-the-clock business support worldwide, three shared-services centers have been built: in San Jose, Costa Rica; in Newcastle, U.K.; and in Manila, Philippines. GBS strategy is to provide best-in-class business support services at the lowest possible costs.

P&G draws on its strong marketing culture to package and offer a catalog of services to its business units across the globe. The catalog embodies two principles of effective marketing—simplicity and choice (with transparent pricing). P&G filters the "best-in-class" service offerings down to a single-page catalog in two "shopping aisles"—Employee Services and Business Services (see Table 1). Brands who consume these services still have control and choice even though some of the solutions are mandated. Within the mandated solutions, there are several tiers of service with different prices. Brand units can influence their costs by choosing a tier of service and influencing the number of units of service consumed. Pricing is also dependent on the region. To encourage business units to adopt the shared solutions, GBS guarantees a 10%–30% cost reduction initially.

An annual "glide-path" of unit price reduction is also built in. Brand units are thus incentivized to phase out their local services increasing the shared service stack to achieve more global scale and allowing the local units to focus more on meeting the needs of the external customer. Another benefit of shared services is to make the cost of each IT service transparent so it can be managed. Previously these costs were often hidden or not managed. To achieve such flexible service delivery requires sophisticated IT financial management. IT service design, internal marketing, pricing, and service optimization and innovation are performed by P&G personnel while the delivery is outsourced. GBS's capability extends beyond IT including financial, sourcing, and HR services. P&G have identified over $600 million in savings from shared services and credits GBS in helping to absorb its large acquisition of Gillette in only 15 months.[4]

For the MNCs we studied, IT shared services achieved scale by brokering and incentivizing the use of standardized IT services across the firms, thus removing cost, duplication, and complexity. Some MNCs then outsourced the bulk of those shared services to external service providers who have even greater economies of scale.

IT Centers of Excellence (CoEs) are also known as competency centers or centers of expertise. CoEs are units that contain strategic IT capabilities identified by the firm as important sources of value creation and service innovation. CoEs are specialized units where the MNCs pool expertise physically or virtually across the globe. These units often do not have operational responsibilities but they serve as strategic resources that focus on designing and developing new solutions, such as, to innovate, and to develop depth in critical expertise. CoEs we encountered included those focused on application development, key business processes (for example, trade processing) and specific technologies or IT platforms (for example, EDI).

Underwood Financials has groups of IT experts who are co-located with the respective global product heads (foreign exchange, bonds, money market, equities, among others) in the HQ where new innovations in financial products typically occur. These IT specialists have in-depth IT and business domain expertise, and they work closely with the business to design and develop new IT solutions. The bank's ability for fast-to-market product launch globally is often dependent on their ability to respond with the necessary IT solutions. The day-to-day operations of the specific product platforms developed are handled by the shared services. These IT experts serve only as a third-level support for complex problems that cannot be resolved by first- and second-level technical support.

Microsoft, similarly, has created the Corporate Solution Deliveries (SD) group comprised of specialized IT application developers led by about 40 solution directors who are located with the businesses and work closely with senior VPs in each major line of business to translate their intimate business understanding into the designing and developing global solutions. In the case of Intel, such pools of IT experts are known as Capability Groups and they focus on enhancing four major IT application development capabilities, namely, the supply-net capability, customer capability, enterprise capability, and platform capability. The customer capability group even reports outside IT to Sales and Marketing for tighter business-IT alignment in developing innovative IT solutions.

As CoEs are designed to provide the firm expertise and innovation in critical areas, they are typically centrally coordinated with the head office identifying the areas of excellence and where they will be located. MNCs are beginning to locate some of their IT CoEs in Asia to take advantage of local talent and cost advantages. P&G located its CoE for mobile marketing in the Philippines to tap into the high usage of mobile phones in Asia. As part of the company's strategic innovation initiative the innovations from this CoE will be diffused to the global market.

Value Managers (VMs) are groups of IT managers that seek to maximize the value of IT for specific business units. VMs, sometimes called customer relationship managers, focus on the IT needs for business units, business functions, and large or fast growing geographical markets. Within the constraints laid out by central IT, the VMs must ensure key business requirements unique to these customers are not overlooked. They build deep relationships with these business customers and support their needs for responsive IT globally. VMs are organized so that the voices of its key customers can be heard, consolidated, and appropriately channeled for prioritization. Equally important, effective VMs also have responsibility to help implement enterprisewide IT initiatives within these customer units. Examples of

Table 2. General characteristics of the three structural elements.

Structural Elements	Objective	Organization	Approach
IT Shared Services	To achieve global/regional scale for cost efficiency while allowing some local choices via ▸ global scale/scope ▸ global sourcing of IT resources ▸ global common platform Heavily **resource**-intensive KPIs: service level agreement, unit cost, simplicity	By major IT functions, IT or business process services: ▸ catalog of services offered, for example, application and infrastructural services ▸ typically located in lower cost regions ▸ some services outsourced to external vendors.	Drive scale via: ▸ active service management and transparency ▸ standardization ▸ consolidation ▸ process improvement ▸ service quality ▸ sourcing
IT Centers of Excellence	To innovate and develop best practices via ▸ global coordination of capabilities ▸ global pooling of IT expertise Heavily **knowledge**-intensive KPIs: # of new global solutions developed, time to market for new application, reuse of best practice across firm, business process performance, and so on.	By innovative technologies or strategic capabilities: ▸ centrally coordinated ▸ may be located outside HQ ▸ can be virtual by pooling distributed experts	Drive innovation via: ▸ pooling deep internal knowledge and expertise ▸ investment into experimentation and innovation ▸ applying and sharing best practices enterprisewide
IT Value Managers	To maximize the value of IT for specific groups in the firm via ▸ being responsive to local needs through a single face of IT ▸ advocating for customer units to central IT ▸ helping implement enterprisewide initiatives locally Heavily **relationship**-intensive KPIs: customer satisfaction, business-IT alignment, partnership maturity, among others.	By major business dimensions: ▸ strategic lines of business ▸ important business functions ▸ large or fast growing geographical markets ▸ major external customers	Push for responsiveness via: ▸ proximity to customer units to capture voice of the customer ▸ simultaneous proximity to central IT ▸ constructive negotiation and facilitation of conflict resolution

centrally initiated enterprisewide programs are global ERP implementations, collaboration tools, and cost-cutting efforts. One CIO put it well: "Without the second objective of implementing enterprisewide initiatives those folks (VMs) go feral and have loyalty only to the local units."

Microsoft has an extended field IT structure that covers its geographical market across 106 countries. Field IT is overseen by an International IT VP reporting to the Global CIO. Below the International IT VP are the IT managers for three regions: North America, Europe/Middle East/Latin America, and Asia. The Asia region, for example, further cascades down to 13 regional clusters. These IT managers play a brokering role, such as in representing Central IT to influence and negotiate with the regional business owners, as well as the customer advocates in championing the interests of these business units and ensuring they derive adequate value from IT.

In one MNC, for example, when a new business in a major Indian city required an application for its fast-growing business, the local general manager wanted it delivered in six weeks, and was willing to pay for the required resources. Conformance with the global organization's IT approval, development, and quality processes, however, would require six months. The IT manager (VM) assessed that delay would impact the business growth, and negotiated a solution to put a program manager to work with the local GM's resources in meeting the local business' timeline. The VM ensured the new system met global guidelines on security and architecture. In another example, the global human resource application was unable to handle the high volume of recruitment in an Asian office. As the time required to change the global application would take too long, the IT manager (VM) negotiated for a short-term module to be created, while providing input to the global applications team. The short-term module would be used until the rollout of the next version of the global HR solution which included the new requirement to process the higher recruitment volume.

The "voice of the field" provided through the VMs in emerging markets can also be a source of global innovation. Through such feedback, P&G recognized the need for new IT applications to cater to the needs of Asian businesses. In one example, P&G noted a difference in the sales distribution model as Asian consumers tend to shop more frequently and in smaller quantities, and hence, began developing IT systems to support the fast growing "high frequency stores" segment. These systems are expected to be useful in other emerging regions as well. Another example is P&G's SKII beauty product, which originated in Japan and has grown to become one of the premium brands in the global cosmetic market. The product distribution for SKII operated on a different business model from P&G's mass market positioning, as it was sold in department stores with dedicated

counter sales consultants. To support the high-touch sales model, systems were built to automate counter operations, to track transactions for each customer, and to provide analysis of sales/marketing plans by customer segment. The systems significantly increased the efficiency for the thousands of sales consultants in Japan. The SKII line, together with the enabling systems, has been successfully deployed to the rest of the world.

Table 2 summarizes the general characteristics of these three structural IT elements, across the companies that we studied.

Configuring the Global-Local Balance in the Structural Elements

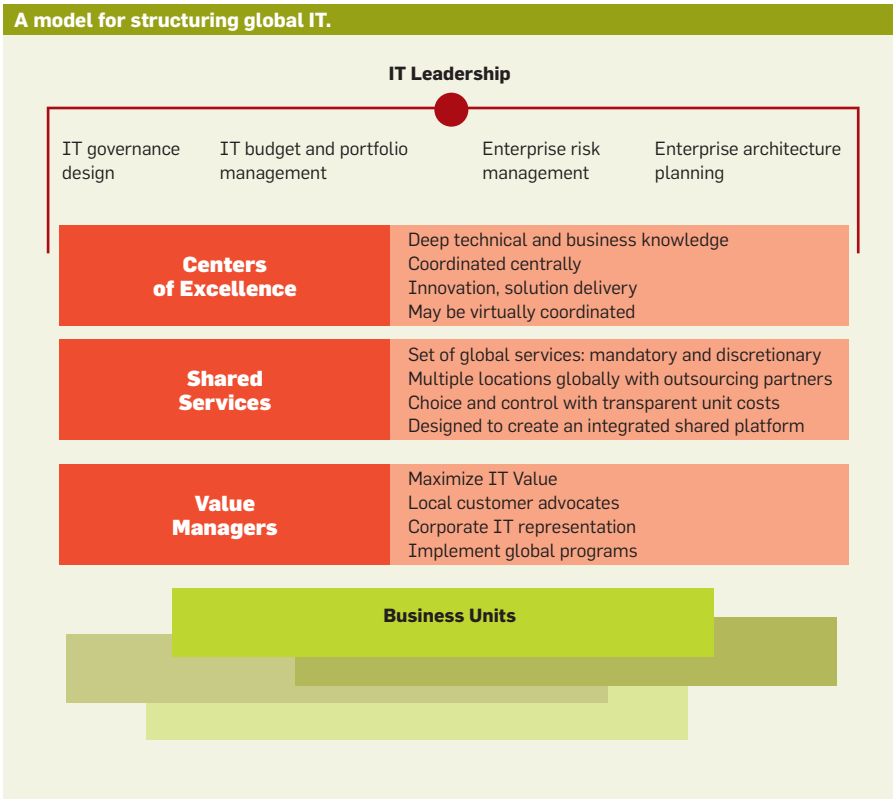
Although the four MNCs we studied are from different industries, they all had implemented similar structural elements of shared services, CoEs, and VMs. This observation suggests some convergence regarding the global structuring of IT resources, as they all seek to simultaneously achieve global scale, while providing local responsiveness and innovation. The accompanying figure summarizes the model for structuring global IT that emerges from our study.

However, multinationals still need to make trade-offs among these strategic objectives. Managers seek these trade-offs by varying configuration of each structural element and distributing resources among them. One of the most common trade-offs we observed was between achieving scale and responsiveness. Companies that sought greater scale tended to have a single global shared service unit. Underwood Financials, for example, has a single global shared service unit in Singapore that serves all business units worldwide over three work shifts. While first-line support was available 24×7, more sophisticated level 3 support was still centralized at headquarters. Responsiveness to complex problems that occurred in other time zones was therefore a challenge. At the time of this study, the head of shared services was lobbying for level 3 support in the Asian time zone as well. Other MNCs traded off global scale for greater regional responsiveness. Microsoft, for example, operates three regional shared services units, covering North America, Europe-Middle East-Africa and Latin America, and Asia respectively.

The configuration of CoEs also reflected trade-offs between local and more global innovation. While most CoEs tend to be global because such specialized expertise is usually costly and in tight supply, companies vary in whether they choose to locate the CoEs at HQ, or abroad, or to create virtual CoEs that pool expertise virtually across several geographies. Underwood Financials' application development CoE for its financial products resides with its business headquarters, which allows it to more tightly link its innovation activities to corporate strategy. P&G, on the other hand, has begun to experiment with locating some of its CoEs abroad, for example, its global mobile marketing CoE is in the Philippines. This is a response to the pervasiveness of mobile communications in Asia. Less commonly, MNCs attempt to achieve even greater responsiveness of local conditions by establishing CoEs at the regional level if there is significant disparity in institutional context, for example, having a separate regional SAP Competency Center in China to address the different language and its unique requirements. The trade-off is in the replication of resources, and greater coordination challenges of aligning local innovation with corporate direction.

MNCs, such as Underwood Financials that have prioritized scale through having a single global shared service center, and also global CoEs located at HQ, clearly are at risk of not responding adequately to legitimate regional or local concerns. In the case of Underwood Financials, they attempted to address this by creating a hierarchy of VMs. Within each region, there are VM roles at the intersection of product lines and region. For example, there would be VMs for bonds-Asia Pacific, bonds-Europe, and so on. These VMs had a matrix reporting structure to both the line of business, and to the regional CIOs. There were various forums that brought together VMs, with business and global IT services and CoEs, as a means to promote coordination and communication within this complex organization structure. Hence, while Underwood Financials reaped scale efficiencies from having global shared services and CoEs, it invested in its elaborate VM structure to be more responsive to local needs.

A model for structuring global IT.



We found VMs play a critical role in ensuring the inherent tensions between scale, responsiveness, and innovation are played out constructively in each business and region. The selection and training of VMs, as well as ongoing support, is critical. For example, Intel actively grooms IT managers who can appreciate both the global and local perspectives. Intel selects high-potential local individuals, exposes them to various "extracurricular activities" such as IT cost reduction initiatives, and sends them on yearlong postings in other roles. Intel also rotates some of its best people in other parts of the world through management stints in Asia to encourage a balanced global-local view so that more informed trade-offs can be made.

The VMs' role in constantly mediating between local demands and corporate policy can be wearing. In some MNCs, VMs who thrived did so by developing and drawing upon an informal network that comprised contacts in the business, corporate IT, and other VMs. The ability to quickly access the right people in the network appeared to enhance their ability to find solutions to global-local problems. Underwood Financials' various forums helped to develop such networks, as did Intel's approach to rotating its managers.

MNCs' trade-offs between scale, innovation, and responsiveness need to be made taking into account a complex mix of factors including: industry, size, desired levels of synergies, access to skilled people, and the roles of scale, innovation and responsiveness in the business model. Table 3 lists some of the questions we suggest CIOs consider in deciding the extent of scale, innovation, and responsiveness desired.

Globalization is an opportunity for CIOs to demonstrate business leadership. Shared services, CoEs, and VMs are structural elements that CIOs are increasingly using to re-bundle traditional IT resources to simultaneously deliver on scale, responsiveness, and innovation. We have observed that successful development of such IT managerial capabilities can deliver significant competitive advantage. C

Table 3. Discussion questions for the design of structural elements in global IT.

Structural Elements	Discussion Questions
IT Shared Services	What is the desired level of scale to be derived from IT shared services?
	Is your product or service global or commoditized? Is there significant value added from local variations?
	What are the factors that contribute to scale in your industry (for example, common customers, processes, resource, or information)?
	What are the common IT applications and infrastructure services that can be bundled to be offered through shared services?
	How are cost shared across the firm (for example, chargeback by service, overhead absorption depending on size, and so on)?
IT Centers of Excellence	Do you need to coordinate IT enabled innovation?
	Are your company's market offerings and competitive advantage driven by innovation in process, product, and/or technology?
	What are the strategic IT capabilities that can contribute to the future competitive advantage?
	What IT capabilities can benefit from regional or global pooling of expertise for continuous innovation?
IT Value Managers	What is the desired level of IT responsiveness to local needs?
	Who are the key user groups (for example, business units, business functions, fast-growing geographical markets) that IT must serve?
	How different are the IT needs of these user groups?
	What is the right balance of implementing enterprisewide IT initiatives and meeting local IT needs?

References

1. Agarwal, R. and Sambamurthy, V. Principles and models for organizing the IT functions. *MISQ Executive 1*, 1 (2002), 1–16.
2. Allen, B.R. and Boynton, A.C. Information architecture: In search of efficient flexibility. *MIS Quarterly 15*, 4 (1991), 435–445.
3. Bartlett, A.B. and Ghoshal, S. *Managing Across Borders: The Transnational Solution*. Harvard Business School Press, 2002.
4. Bloch, M. and Lempres, E. From internal service provider to strategic partner: An interview with the head of Global Business Services at P&G. *McKinsey Quarterly* (July 2008).
5. Brown, A.E. Framing the frameworks: A review of IT governance research. *Communications of the AIS 15* (2005), 696–712.
6. Brown, C.V. Examining the emergence of hybrid governance solutions: Evidence from a single case study. *Information Systems Research 8*, 1 (1997), 69–94.
7. Brown, C.V. and Magill, S.L. Reconceptualizing the context-design issue for Information Systems function. *Organization Science 9*, 2 (1998), 176–194.
8. Brown, J.S., and Hagel III, J. Innovation blowback: Disruptive management practices from Asia. *McKinsey Quarterly 1* (2005).
9. Ein-Dor, P. and Segev, E. Organizational context and MIS structure: Some empirical evidence. *MIS Quarterly 6*, 3 (1982), 55–68.
10. Fonstad, N.O., and Robertson, D. Transforming a company, project by project: The IT engagement model. *MISQ Executive 5*, 1 (2006), 1–13.
11. Gallagher, K.P., and Worrell, J.L. Organizing IT to promote agility. *Information Technology and Management 9*, 1 (2008), 71–88.
12. Ghemawat P. Distance still matters: The hard reality of global expansion. *Harvard Business Review*, Sept. 2001.
13. Santos, J., Doz, Y., and Williamson, P. Is your innovation process global? *Sloan Management Review*, Summer 2004.
14. Sambamurthy, V. and Zmud, R.W. Arrangements for information technology governance: A theory of multiple contingencies. *MIS Quarterly 23*, 2 (1999), 261–290.
15. Schwarz, A., and Villinger, R. Integrating Southeast Asia's economies. *McKinsey Quarterly 1*, 2004.
16. Tavakolian, H. Linking the Information Technology structure with organizational competitive strategy: A survey. *MIS Quarterly 13*, 3 (1989), 309–317.
17. Von Simson, E.M. The recentralization of IT. *Computerworld 29*, 51 (1995), 1–5.
18. Weill, P. Don't just lead, govern: How top-performing firms govern IT. *MISQ Executive 3*, 1 (2004), 1–17.
19. Weill, P. and Ross, J.W. A matrixed approach to designing IT governance. *Sloan Management Review 48*, 2 (2005), 26–34.
20. Weill, P. and Ross, J. *IT Governance*. Harvard Business School Press. Boston, MA, 2004
21. Yin, R.K. *Case Study Research: Design and Methods*. Sage Publications, Beverly Hills, CA, 1984.
22. Zmud, R.W. Design alternatives for organizing Information Systems activities. *MIS Quarterly 8*, 2 (1984), 79–93.
23. Zmud, R.W., Boynton, A.C., and Jacobs, G.C. The information economy: A new perspective for effective information systems management. *Data Base 18*, 1 (1986), 17–23.

Siew Kien Sia (asksia@ntu-edu.sg) is an associate professor and director of the Information Management Research Center at Nanyang Technological University, Singapore.

Christina Soh (acsoh@ntu-edu.sg) is an associate dean and professor at Nanyang Technological University, Singapore.

Peter Weill (pweill@mit.edu) is chair of the Center for Information Systems Research and Senior Research Scientist at MIT Sloan School of Management, Cambridge, MA.

ACM's Online Books & Courses Programs!

Helping Members Meet Today's Career Challenges

NEW! Over 2,500 Online Courses in Multiple Languages Plus 1,000 Virtual Labs from Element K!

element k®
your learning solutions partner

ACM's new Online Course Collection includes over **2,500 online courses in multiple languages, 1,000 virtual labs, e-reference tools, and offline capability**. Program highlights:

The ACM E-Learning Catalog - round-the-clock access to 2,500+ online courses on a wide range of computing and business topics, in multiple languages.

Exclusive vLab® Virtual Labs - 1,000 unique vLab® exercises place users on systems using real hardware and software allowing them to gain important job-related experience.

Reference Tools - an e-Reference Library extends technical knowledge outside of the classroom, plus online Executive Summaries and quick reference cards to answer on-the-job questions instantly.

Offline Player - members can access assessments and self-study courses offline, anywhere and anytime, without a live Internet connection.

A downloadable Quick Reference Guide and a 15-minute site orientation course for new users are also available to help members get started.

The ACM Online Course Program is open to ACM Professional and Student Members.

---

600 Online Books from Safari

ACM members are eligible for a **special 40% savings** offer to upgrade to a Premium or Full Library subscription.
For more details visit:
http://pd.acm.org/books/about_sel.cfm

The ACM Online Books Collection includes **full access to 600 online books** from Safari® Books Online, featuring leading publishers including O'Reilly. Safari puts a complete IT and business e-reference library right on your desktop. Available to ACM Professional Members, Safari will help you zero in on exactly the information you need, right when you need it.

Safari® Books Online

500 Online Books from Books24x7

All Professional and Student Members also have **full access to 500 online books** from Books24x7®, in ACM's rotating collection of complete unabridged books on the hottest computing topics. This virtual library puts information at your fingertips. Search, bookmark, or read cover-to-cover. Your bookshelf allows for quick retrieval and bookmarks let you easily return to specific places in a book.

books24x7® a SkillSoft Company

Association for Computing Machinery

Advancing Computing as a Science & Profession

pd.acm.org
www.acm.org/join

DOI:10.1145/1666420.1666440

With no HIV vaccine in sight, virologists need to know how the virus will react to a given combination drug therapy.

BY THOMAS LENGAUER, ANDRÉ ALTMANN, ALEXANDER THIELEN, AND ROLF KAISER

Chasing the AIDS Virus

THE MOST CHALLENGING problem for physicians treating AIDS patients with anti-HIV drugs is that the virus almost inevitably evolves toward resistance against any administered drug therapy. Once resistance is manifest, the physician must change the therapy regimen, which typically consists of a combination of anti-HIV drugs. Here, we describe bioinformatical methods supporting the choice of an effective follow-up therapy. Using underlying clinical-resistance databases and statistical-learning methods, we identify as-yet-undescribed resistance mutations, predict the level of resistance of a viral variant extracted from the blood of an AIDS patient against anti-HIV drugs, and estimate the expected mutational path of the virus toward resistance against specific combination drug therapies. This computational method enables us to rank possible therapies with respect to their expected effectiveness. We also offer a computational test for the expected effectiveness of a new drug capable of blocking viral cell entry.

Our analyses, which are freely available on the Internet via the server http://www.geno2pheno.org, are used routinely for treating about two-thirds of AIDS patients in Germany.

AIDS is a major scourge worldwide, causing millions of deaths annually. Whereas due to education and preventive measures, the number of new infections in the developed world is comparatively limited, other parts of the world (notably Sub-Saharan Africa) exhibit very high infection rates. The disease is on the rise globally.[20]

The AIDS pathogen—the Human Immunodeficiency Virus, or HIV—crossed over to humans from apes as recently as 100 years ago. The pathogen and its new host apparently have not yet adapted through co-evolution. Consequently, HIV is highly pathogenic in humans, unlike chimpanzees, which exhibit very high infection rates with the Simian Immunodeficiency Virus, or SIV, without presenting debilitating symptoms.

AIDS is especially lethal for a number of reasons. For the human population, one danger involves the fact that symptoms develop slowly, so hosts

» key insights

- **Clinical databases can be mined to help generate statistical models that predict HIV's viral resistance to administered drugs.**
- **These models incorporate interactions between drugs in combination with drug therapies, estimating future viral escape path toward resistance to an applied drug regimen.**
- **By continually incorporating new clinical insights and drugs, the software tool helps support therapy decisions in clinical routines.**

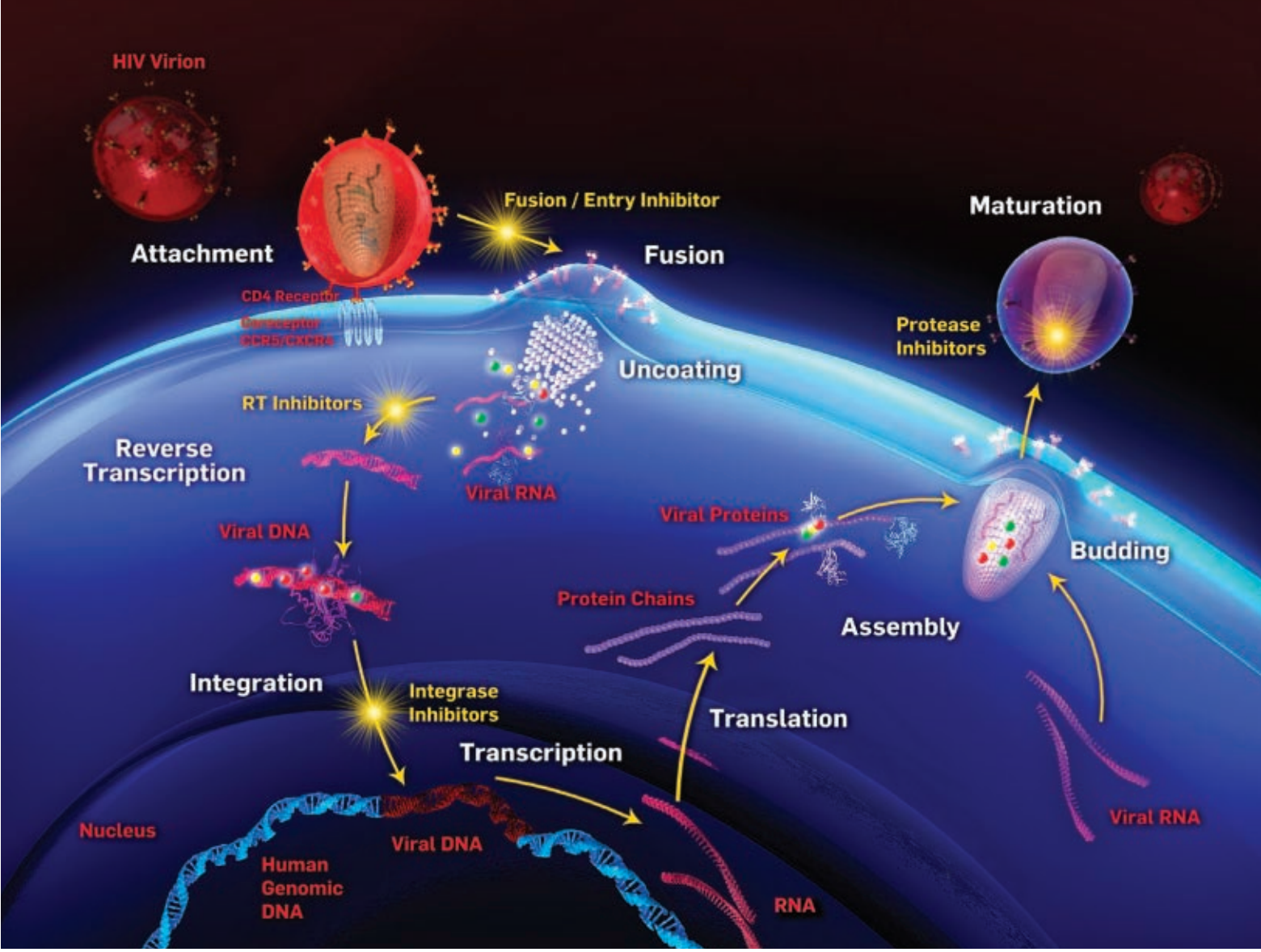


Figure 1. Replication cycle of HIV, yellow dots: RT molecules, green dots: IN molecules, red dots: PR molecules.

ILLUSTRATION BY STUDIO MACBETH

can be infectious for extended periods without their contacts knowing. For infected patients one problem involves the fact that the virus inserts its genome into the genome of the infected cell. These people cannot be cleared of the virus. As we describe here, the virus evolves dynamically. Thus it is difficult to produce vaccines against HIV, and no vaccine against HIV is in sight. Since there are major obstacles to curing AIDS, the objectives of drug therapy are to ease symptoms and delay progress of the disease by suppressing viral replication.

Since the virus continually changes in a patient, physicians are chasing a moving target. Given a particular drug therapy, the virus evolves toward resistance. The drug therapy then has to be changed to suppress what is now the prevalent viral variant in the patient. The underlying biological relationships between the viral genotype—the particular genome sequence of the virus—and the viral resistance phenotype—its ability to escape antiviral drugs—are complex and not well understood. Therefore, drug therapies are selected not so much on the basis of understanding the underlying biology as they are on the basis of clinical experience.

Clinical experience in treating AIDS patients with antiviral drugs has been collected for the past 20 years and assembled in sizeable resistance databases. The complexity of the relationship between viral genotype and resistance phenotype suggests using statistical-learning methods to support computational models for predicting the resistance phenotype from the viral genotype. For this purpose, we have developed the Web server geno2pheno (http://www.geno2pheno.org), offering such analysis for free on the Web.

Replication Cycle of HIV

HIV is not an autonomous organism but rather an enveloped piece of genome, roughly 10,000 letters of genomic text (bases) in protein packaging. This tiny genomic text (compared to three billion letters of the human genome) defines one of the most vicious biological killers. The structure of the HIV virus particle (virion) is known in detail.[9]

As with all viruses, to replicate, HIV uses the cells it infects, usually those of the human immune system (such as T-lymphocytes). Knowledge of the replication cycle of HIV (see Figure 1) is the basis for all drug therapies in use today. The genome of HIV does not consist of DNA (as in humans) but of the close relative RNA that in humans is used for translating genomic information and regulating cellular processes. The replication cycle of HIV begins with HIV using its surface

Statistical Learning Methods

Statistical learning comes in two versions: supervised and unsupervised. Unsupervised aims to elucidate patterns in unstructured (usually high-dimensional) data sets. Here, we exemplify unsupervised statistical learning methods with the mutagenetic trees model. Supervised methods use data sets of (usually high-dimensional) inputs x and associated (scalar or categorical) outputs y, to derive computational procedures for predicting (given a new input x_0) the associated output y_0. Here, we exemplify supervised statistical learning through the support-vector-machine model.

Mixtures of mutagenetic trees. A mutagenetic tree is a tree-shaped Bayesian model; two are included in Figure 6. The tree is rooted, and its root represents the viral wildtype, or the absence of mutation. Each other tree node represents a mutation. The edges of the tree are directed downward and labeled with conditional probabilities. Given the presence of all mutations along the path from the root of the tree to the source node of an edge, the label of the edge indicates the probability that the mutation at its target node takes place.

In principle, a mutagenetic tree can be used to generate a set of viral variants by performing a random experiment based on the probabilities at the edges of the tree. We are not interested in explicitly performing such an experiment. Rather, given a set of viral variants (such as the subset of viral genotypes in our resistance database that has seen a certain drug, like saquinavir) we are looking for the mutagenetic tree that generates that set with greatest probability (maximum likelihood model). This tree best represents the escape of the virus toward resistance against the drug saquinavir.

Desper et al.[8] presented a method for finding a mutagenetic tree that is optimal under restricted circumstances and good (only) in the general case; the result is derived not in a viral context but in the context of cancer research. We extended the method to be able to generate several trees,[4] because viral escape paths do not usually submit to a single tree model, as reflected in Figure 6. Again, this method is heuristic; it does not find the best model but just a reasonably good model. Rather than labeling the edges of a mutagenetic tree with conditional probabilities, we can also annotate them with expected times for the relevant mutation to occur. Labeling affords a route to analyzing the times the virus takes to escape toward resistance.[3] We use this model to assess therapy effectiveness.

Classifying therapy success with support-vector machines. Different versions of THEO have used different multivariate statistical-learning methods to come up with accurate classifiers. Among them are logistic model trees[11] and support-vector machines.[7] Support-vector machines are a recent, popular method for classifying data that regards data as points in a (usually high-dimensional) Euclidean space. In our case, each data point represents a therapy change episode, or event where physicians assign a new therapy based on a viral genotype seen in a patient. Some therapy selections are successful, others are not. This dichotomy represents our binary classification problem. The question of what is a successful therapy and what is a failure, both medically and methodically, is beyond our scope here.

A linear support-vector machine defines a hyperplane that best separates the set of points indicating therapy successes from the points indicating therapy failures. The hyperplane divides the Euclidean space into two half-spaces, one for therapy success, one for therapy failure. What is the "best" hyperplane (for minimizing risk of wrong predictions) is defined in terms of two criteria:

Discriminating between therapy successes and failures. As few therapy data points as possible should be located "on the wrong side" of the hyperplane, that is, we do not want to see therapy failures in the half-space for the successes and vice versa. The further a point is in the wrong half or away from the hyperplane on the wrong side, the more it reduces the quality of the model; and

Maximizing prediction reliability. The hyperplane should be as distant as possible from the closest correctly classified points. Since the hyperplane represents the "decision boundary," points lying close to it represent uncertain decisions, and small changes in the data or in the location of the hyperplane can reverse their classification.

Quadratic programming techniques are used to find the optimal hyperplane according to these criteria.

While we have taken state-of-the-art versions of support-vector machines developed by others, our main objective here is to define the Euclidean space to which we apply the support-vector machine. We must therefore address the following issues:

Representing viral genotypes. Should we use binary indicator variables? Which mutations should we consider? Considering all possible mutations leads to a high-dimensional space and is thus infeasible; and

Additional information for the method. The therapy we want to apply is a necessary input. Additional input includes predictions of resistance factors against single drugs, the probability that the virus will achieve resistance against a drug in a certain time interval (estimated via the mutagenetic trees), and previous antiretroviral drugs to which the patient was exposed.

Addressing them is difficult, as we must balance the amount of information we present to the method against the available data. The more information we present, the more complex are the resulting models. However, we must find the best model on the basis of limited data. If models are too complex we incur the risk of overtraining the model. An overtrained model incorporates not only patterns pertaining to the phenomenon or process we want to analyze and whose results we want to predict (here viral resistance) but also idiosyncrasies of the particular data set on which we derived the model. Such idiosyncrasies do not generalize to future data. Thus an overtrained model suffers from reduced predictive power. We have performed several studies and reported our choices.[1,2,18v]

protein gp120 to bind to surface proteins of the host cell. This binding event triggers a cascade of structural changes of the participating proteins that result in HIV entering the host cell. Once inside, HIV sheds its molecular envelope and uses a special viral protein—the reverse transcriptase (RT)—to copy its RNA genome to DNA. The DNA is then transported into the cell nucleus where it is spliced into the genome of the host cell with the help of a second viral protein—the integrase (IN). At this stage, the viral DNA is called a "provirus." Once the cell begins to divide, as it does within an immune response, it manufactures all components of the virus. These components assemble near the cell surface, and a new still-immature virion buds from the cell. In a final maturation step, strings of viral proteins in the immature virion (the so-called polyproteins) are cleaved to yield the functional viral proteins. This renders the virion infectious. The protein performing the cleavage is the viral protease (PR). Each host cell is able to produce thousands of virions for a long period before inevitably dying.

Drug Therapies Against HIV

More than two dozen drugs against HIV are in clinical use; see http://www.fda.gov/oashi/aids/virals.html for the current list of U.S. Food and Drug Agency-approved anti-HIV drugs. All are small molecules that block (inhibit) the function of a specific protein involved in the viral replication cycle, the so-called target protein. One way to block a protein is to bind to it in a place that deacti-

vates the protein, either by replacing its natural binding partner or by interfering with essential protein movements. Target proteins can be viral or human. The classical target proteins are viral, namely RT and PR. Originally, viral proteins were preferred because one does not want to interfere with unknown functions of human target proteins. However, viral target proteins have the disadvantage that the virus can quickly change them through mutation and thus evolve toward drug resistance. More recently, human proteins have also been targeted by antiviral drugs.

Toward Resistance

If the virus were not so variable, one or two AIDS drugs would suffice. But the virus changes its genome with practically every copy. The reason for such flexibility is that RT lacks a proofreading mechanism and does not repair copy errors. Mutations in the HIV genome can result in changes in the composition of its proteins. Most of these changes are detrimental or even lethal to the virus, but with many millions to even billions of virus copies produced daily in the same patient, chances are high that a viral variant will arise quickly whose target protein remains functional even in the presence of a drug. Such a virus is resistant to the drug.

Suppressing viral replication means reducing the number of experiments the virus can perform to produce a resistant variant. In order to increase the barrier of the virus to escape toward resistance, several drugs targeting different viral proteins are given simultaneously. This scheme, called highly active antiretroviral therapy, or HAART, renders therapies effective for much longer periods of time. The virus always wins. Most current therapies remain effective for only months to a few years.

Antiviral Therapies

Once the virus is resistant, the treating physician must select a new drug therapy that effectively suppresses the present viral variant. The standard of care today is to use diagnostic tools for selecting a new therapy regimen. There are two fundamental approaches toward this goal:

Phenotypic resistance testing. Phenotypic resistance testing basically provides a lab test, essentially exposing the virus taken from a patient's blood serum in cell culture to increasing drug concentrations and observing quantitatively how quickly the replication rate of the virus declines. The decline is compared with the decline of the replication rate of a nonresistant reference virus. The comparison yields a quantitative measure of viral resistance against individual drugs, the resistance factor. This measure is the drug concentration that cuts the replication rate of the patient's virus in half divided by the drug concentration that cuts the replication rate of the reference virus in half. Large resistance factors mean greater resistance.

Phenotypic resistance testing meets with major obstacles when used in clinical practice, mainly because such testing is restricted to labs with high security levels and is thus difficult to standardize and not sufficiently accessible. Cost is another issue.

Genotypic resistance testing. In contrast, genotypic resistance testing determines the genomic sequences of the relevant parts of the viral genome taken from a patient's blood serum. The relevant genome sequence can be obtained cheaply, quickly, and with standardized procedures by many laboratories. However, it is not easy to infer the resistance phenotype from the viral genotype. Virologists used to perform this interpretation by hand with the help of a so-called mutation table; mutation tables are offered and continually updated by such authorities as the International AIDS Society,[10] collecting the global knowledge on mutations observed to cause resistance against specific drugs. Figure 2 is an excerpt from a mutation table covering three protease inhibitors. The blue bar represents the protein sequence, here the protease with 99 amino-acid positions. Numbers inside the blue bar indicate protein-sequence positions. The amino acid of the reference virus at that position is given above the number. Resistance mutations at that position are indicated below the number. Each row pertains to a single drug named to the left of the row. Mutations enter the table as a result of committee consensus. More recently, the tables have been turned into expert systems that provide more complex rules. These systems can also express interactions between different mutations that result in resistance or susceptibility of the virus to a given drug.[16]

Computational Biology

One problem with mutation tables and expert systems is they are the result of a consensus among human experts, rather than being systematically derived from the underlying clinical data. This is where the contribution of computational biology comes in. If we can render the clinical resistance databases computer-readable, we can apply statistical-learning methods to systematically derive estimates of the resistance phenotype from the viral genotype. We can also assess not only the level of resistance of the virus present in the patient but also estimate the path the virus will take toward resistance in the future if presented with a specific drug therapy, along with the time the virus will take to get there.

Since 1988, we have been partners in a number of consortia collecting HIV-resistance data comprising viral genotypes, associated clinical markers (such as counts of virus and immune cells in the blood), and phenotypic-resistance data where available. We did this nationally in Germany through the Arevir database.[17] In 2004, we co-founded

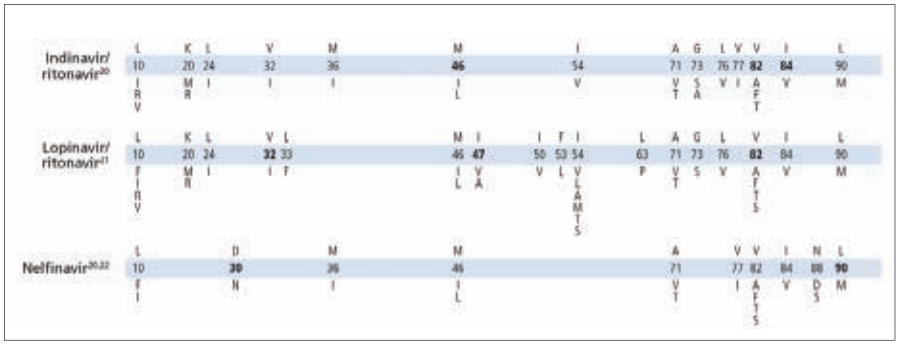
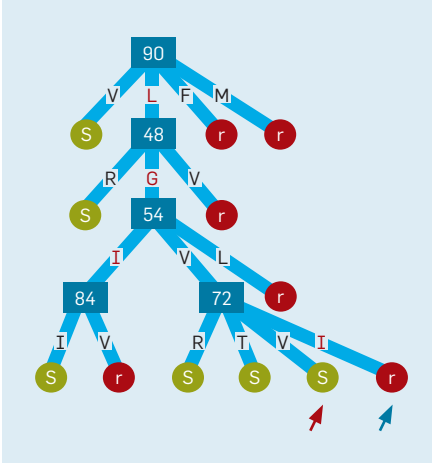


Figure 2. Excerpt from a mutation table.[10]

Figure 3. Decision tree for resistance against the AIDS drug saquinavir.



the EuResist consortium, whose database is the result of integrating several large resistance databases for all of Europe.[18] To our knowledge, the EuResist database is the largest HIV-resistance database worldwide, harboring data on just under 100,000 therapies for almost 34,000 patients. Paired data on viral-mutations and clinical response to treatment is available for more than 5,000 therapies.

Identifying new resistance mutations. Given an HIV-resistance database, we use statistical methods to systematically find resistance mutations. A resistance mutation is one, such that viruses resistant (against a given drug) are highly enriched among the viral variants with the mutation, unlike the ones without the mutation. The "information content" a mutation harbors on viral resistance against a given drug can be quantified in various ways, including mutual information and distance from the decision boundary in a discriminatory classifier. Using such methods, we have uncovered new, that is, as-yet-undescribed resistance mutations.[19] That study won a Best Presentation award at the Third European HIV Resistance Workshop, Athens, Greece, in 2005. This peer recognition reflects how much virologists and clinicians are interested in approaches to identifying resistance mutations beyond the classical mutation tables.

Resistance prediction based on complete viral genomes. The second class of models incorporates multivariate analysis to systematically deduce the kind of information offered less systematically by rule-based expert systems. We have produced many such models, including classifiers (into the resistance classes resistant and susceptible) and regression models that estimate the numerical value of the resistance factor. All models are trained on the data available in our resistance databases, notably genotype-phenotype pair data, that is, viral variants for which we have both the viral genotype and the resistance factor. We employed decision trees[6] and random forests to determine the classifications. For regression we found support-vector machines are most effective.[5] Our statistical-learning methods are state-of-the-art and adapted to the respective problem; the sidebar "Statistical Learning Methods" outlines two such methods: mutagenetic trees and support-vector machines. Modeling and feature selection are the focus of the effort. Appropriate statistical validation of the resulting models represents another major aspect of our research.

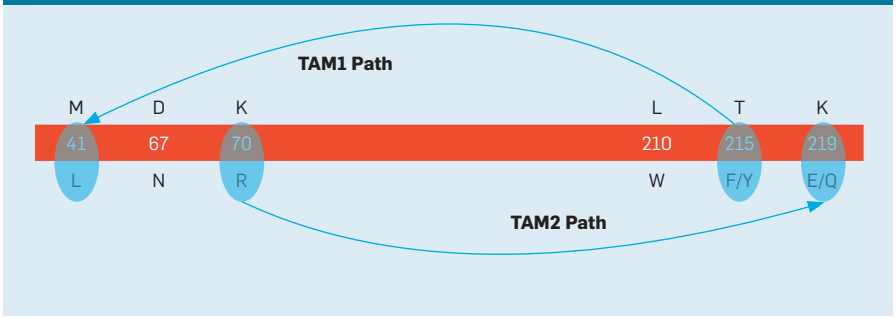
Figure 3 is a decision tree for the resistance of HIV against the PR inhibitor saquinavir. The branching nodes are labeled with amino-acid positions in the target protein PR. Terminal nodes are labeled with the classes "resistant" and "susceptible," respectively. Edges leaving a node are labeled with amino acids found at these positions. The amino acid of the reference virus (no mutation) is in red. The path leading from the root of the tree (top) to the blue arrow indicates a single mutation at position 54 from the reference Isoleucine (I) to Valine (V). (All other edges along the path represent the reference virus.) The resulting virus is resistant according to the model (red terminal node). However, if in position 72, there is also a mutation from the reference Isoleucine (I) to Valine (V) (red arrow), then the virus is susceptible (green terminal node) to treatment with the drug. Such resensitization events present interactions between different mutations and are derived systematically from the procedure of learning decision trees for drug resistance. Cross-validation helps us show that our decision trees make accurate predictions in approximately 90% of all cases.

Our resistance models are the basic service of the geno2pheno server. Practicing physicians and laboratory virologists paste in the nucleic acid sequence of the relevant genes of the viral variant extracted from a patient's blood. The analysis responds with the kind of output listed in Figure 4, where each row represents a drug. Column 1 names the drug. Column 2 gives the estimated resistance factor. Column 3

Figure 4. Sample output of geno2pheno resistance analysis.

	Drug	RF	Z-score	Scored Mutations						
Reverse transcriptase inhibitors	ZDV	0.995	−0.755	122k	123E	179D	103N	115F	74V	
	ddI	2.861	1.220	74V	184V	190A	177E			
	d4T	1.169	0.065	178V	184V	219T	122K	123E	190A	
	3TC	63.041	14.807	184V						
	FTC	19.200	3.607	184V	122K	115F				
	ABC	4.746	8.870	184V	74V	123E	214F	115F		
	TDF	0.655	−2.510	184V	74V	177E	214F	115F	200A	219T
	NVP	515.252	6.645	103N	179D	102N	211K			
	EFV	358.773	10.833	103N	179D	200A	214F	178V	177E	
Protease inhibitors	SQV	1.334	0.792	63P	71T	37N				
	IDV	1.332	−0.073	63P	71T					
	RTV	1.527	0.631	63P	71T	3I				
	NFV	1.503	0.377	63P	3I					
	APV	1.385	0.820	63P						
	LPV	1.522	1.256	63P	71T	35D				
	ATV	1.264	−0.069	71T	63P					
	TPV	1.521	0.137	3I						
	DRV	2.235	−0.327	35D						

Figure 5. Two favored mutational escape paths of HIV from the therapy with the RT inhibitor zidovudine.



gives a normalized value reflecting the significance of the resistance value. Column 4 lists mutations found in the input sequence, red if they strengthen the resistance of the virus and green if they weaken it. The data in the figure points to strong resistance against many inhibitors of RT and therapy options targeting PR. The Geno2pheno server is the basis for supporting treatment decisions in about two-thirds of HIV-infected patients treated in Germany.[13] This means at least 12,000 decisions for treatment selection per year in Germany involve geno2pheno or its findings.

Chasing the virus. This analysis treats each drug separately. Given the output in Figure 4, the physician assesses the resistance level of the virus against each individual drug and manually composes the combination drug therapy that is (hopefully) effective against the present virus. We also look into the future of the virus. Presented with a given combination drug therapy, how will it react? What are its mutational escape paths and how long will the therapy stay effective? The virus does not just randomly introduce mutations. Rather, it follows more-or-less established mutational escape paths; Figure 5 outlines two favored paths from a therapy with the single AIDS drug zidovudine (ZDV, AZT). (The notation is analogous to that of Figure 3.) We denote with K70R the mutation of K to R in position 70 (of RT). Hence, one escape path is K70R followed by K219E/Q.

The biological reasons for the virus following these paths are not well understood. But the paths show up in a clinical HIV-resistance database. Finding them is simple if we have longitudinal data. The data comprises sequences of viral genotypes and clinical parameters from the same patient over long periods of time. However, such data is difficult to come by. Our databases are dominated by cross-sectional data involving only a few or single data points for each patient. Nevertheless, we are still able to identify favored escape paths from cross-sectional data, as in Figure 5.

A database of cross-sectional data on therapies with zidovudine will not contain many viruses having mutation M41L but not the mutation T215F/Y. This mutational pattern indicates the direction of the escape path. We have developed statistical models that pinpoint the paths, so-called mixtures of mutagenetic trees, from the database[4]; Figure 6 outlines the trees derived from the database concerning zidovudine therapy.

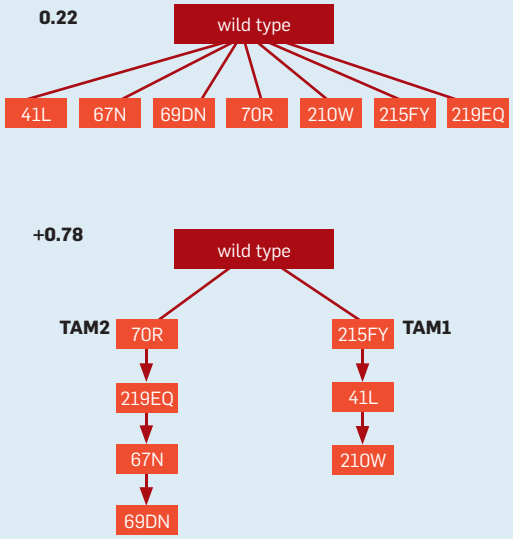
Figure 6 outlines a mixture model of two mutagenetic trees, the bottom one expressing the two thymidine analogue mutations (TAM) escape paths and the top one (noise tree) expressing an unstructured escape to resistance. The mixture model indicates that 78% of the data is explained by the escape via the TAM paths; 22% can be viewed as noise. The sidebar explores mixtures of the mutagenetic trees model.

The analysis of viral escape is available on the geno2pheno server via the applet known as THEO (therapy optimization), which ranks all reasonable therapies by the probability of their staying effective for six months or longer for the Web-server version of the software. The statistical method for doing this is discussed in the sidebar section on support-vector machines. Figure 7 outlines the results of THEO on the same data as in Figure 5.

Training the model requires data encompassing the viral genotype, the drugs involved in the therapy, and clinical follow-up data on the effectiveness of the therapy. How to characterize a successful therapy is complex. We do not, for example, need the resistance factor to be input for each query. We can supply it through our computational-resistance prediction method discussed earlier. Also, the expected future viral evolution can be estimated through mutagenetic trees.

THEO, which has been validated extensively, improves the accuracy

Figure 6. Two mutagenetic trees indicating the escape paths in Figure 5; "wild type" indicates the reference virus.



of therapy selection substantially[1,2]; for example, approximately 24% of the therapy selections reported in the 2006 version of our Arevir database turned out to be ineffective. Using THEO could have helped reduce the error rate in selecting effective therapies below 15%.

The EuResist project (http://www.euresists.org) adds two qualities to the research we discuss here: Data collection includes data from several European countries; and, on the EuResist prediction server, three independently developed prediction engines are executed and return individual results and a consensus prediction.[18]

New Drugs

Using sophisticated methods to administer antiviral combination drug therapies does not obviate demand for continually developing new drugs. For an individual patient, administering a drug provokes resistance mutations that accumulate within the virus genome. Eventually, only new drugs with new modes of action or even new target proteins will deliver additional effective drug therapies. Moreover, AIDS drugs age as resistance mutations accumulate in the global viral population, necessitating continuous development of new drugs. And clinical side effects enforce the development of new drugs with the same mode of action as existing "old" drugs. Such new drugs might replace the "old" drugs but might also provoke slightly different resistance mutations.

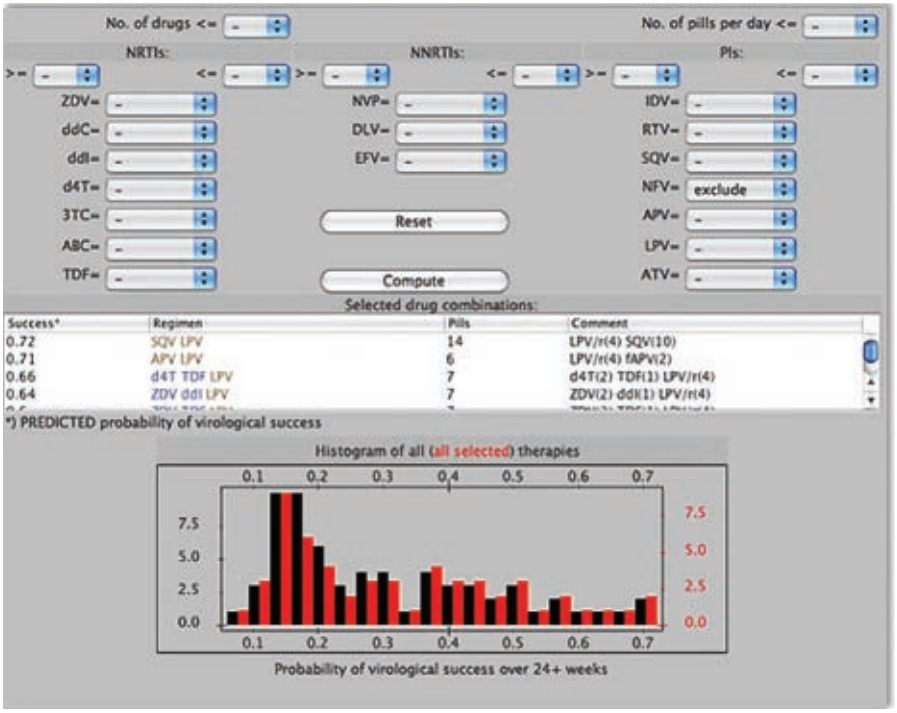


Figure 7. THEO applet.

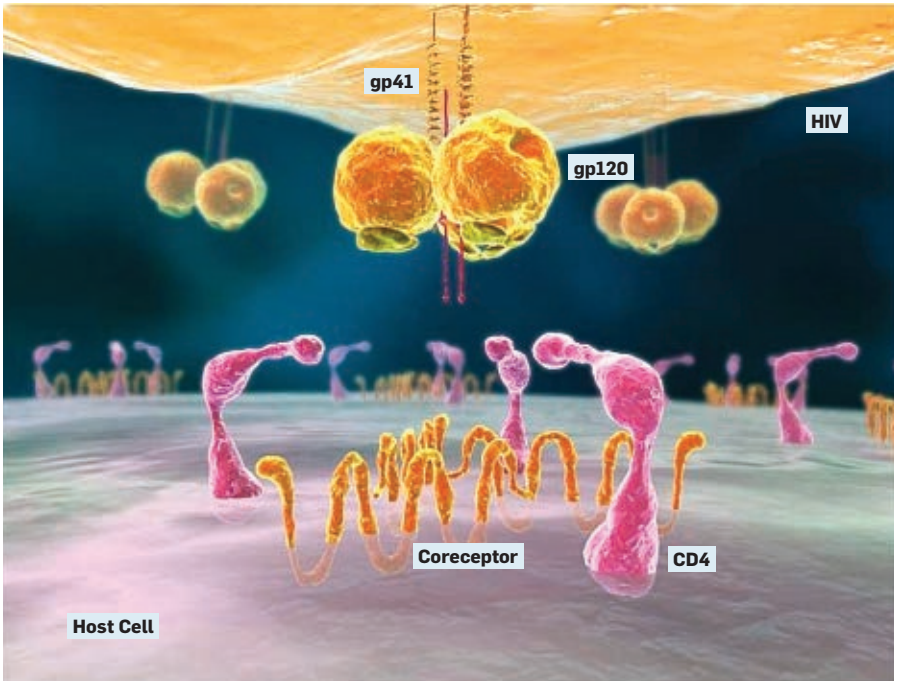


Figure 8. Proteins participating in HIV cell entry (courtesy Pfizer).

Drugs targeting RT and PR were the basis of AIDS therapy until the early 2000s. Since 2003, drugs targeting other proteins have come onto the market. Especially attractive targets for anti-HIV drugs are proteins facilitating cell entry of HIV. Such targets are chosen because blocking viral cell entry helps prevent integration of the viral DNA into the cellular genome. To understand how we block viral cell entry we must look at the process of HIV entering the cell in more detail (see Figure 8). First, the viral surface protein gp120 binds to the cellular receptor protein CD4. This leads to a conformational change in gp120 so it can then bind to an additional cellular protein, the so-called co-receptor. The binding of gp120 to the cellular co-receptor triggers the actual viral cell entry, during which the helical (corkscrew-like) viral surface protein gp41 penetrates the cellular membrane, and the hull of HIV fuses with the cellular membrane. HIV can use one of two cellular surface proteins—CCR5 or CXCR4—as a co-receptor; some viral strains use either. The co-receptor specificity of HIV is also called viral tropism. A virus using CCR5 is called R5-virus. Analogously, a virus using CXCR4 is called X4-virus. A virus using either co-receptor is called dual-tropic, or R5/X4-virus.

Viral tropism has important clinical consequences. For example, the initial infection results almost exclusively in an R5-virus population; we assume that X4-viruses may infect the patient but can be controlled initially by the immune system. Approximately 1% of the

Caucasian population worldwide lacks a functional gene for CCR5, has no apparent symptoms, and is resistant to being infected by HIV. As the disease progresses, a virus using CXCR4 can become dominant.

Targets for drugs that block cell entry are the viral surface protein gp41 and the cellular co-receptor CCR5. The latter is targeted by the drug Selzentry/Celsentri, which contains the active substance maraviroc (developed by pharmaceutical manufacturer Pfizer). Regulatory agencies in Europe and the U.S. require viral tropism testing before administration of this drug. As with resistance analysis, there are again two options for a viral tropism test: One is a lab-based phenotypic test, the other a genotypic test with computer-based interpretation. The advantages and disadvantages of each are similar to those in resistance testing; for example, phenotypic tests are accurate but take a long time and are expensive and not always easily accessible. Moreover, and in contrast to phenotypic resistance tests, phenotypic tropism tests provide only a classification into X4-capable or not-X4-capable and no quantification of the risk of using the wrong co-receptor.

The main problem with genotypic tests is the elucidation of the genotype-phenotype relationship. The geno2pheno server offers a prediction for viral tropism from genotype. As with resistance analysis it is based on careful modeling of the input and on the development of a multivariate statistical model trained on genotype-phenotype pair data.[12] In this instance, the phenotype is the viral tropism, not the resistance against a drug, though the co-receptor switch can be viewed as a way for HIV to evade drugs blocking CCR5. Three notable advantages of this genotypic approach are lower costs, wider availability, and a quantification of the risk of using the CXCR4 co-receptor.

Since there are major obstacles to curing AIDS, the objectives of drug therapy are to ease symptoms, suppress viral replication, and delay progress of the disease.

Measuring the Viral Quasi-Species

A problem with genotypic data that seems more relevant for predicting viral tropism than for predicting drug resistance is that the patient harbors not a single viral variant but rather a diverse viral population, or so-called quasi-species. Classical genotypic measurements reduce the quasi-species to a single viral variant (the dominant one) or to a sequence consensus of a few frequent viral variants. However, minorities of X4-virus present in the viral quasi-species (but not detected by the genotypic test) can accumulate in the patient under therapy with CCR5-blockers. Detecting such minorities may be clinically important, and phenotypic tests are able to detect them. To enable genotypic tests to also detect them, we use new deep sequencing technology called pyrosequencing[14] to generate data from which appropriate computational procedures reconstruct (with great accuracy) the profile of the whole quasi-species. One of our current research activities targets predicting viral tropism and its clinical consequences based on such data.

Outlook

The work described here can now be extended in several directions. For example, a multitude of questions pertain to the statistical-modeling procedure, including those involving the representativeness of the clinical databases, how to improve prediction accuracy when sufficient training data is unavailable, and how to follow different notions of therapy success.

More fundamental, the technology applies to other viral infections, the pathogens of which exhibit dynamic evolutionary development, a property shared by Hepatitis C (caused by HCV) and Hepatitis B (caused by HBV). In both cases, drug development and the collection of resistance data has not advanced as far as it has for HIV. We are involved in projects that collect such data, intending to transfer our technology to these diseases. We have gone beyond infectious diseases and applied the mutagenetic-trees technology to assessing the status of tumor progression in cancers from data on the evolutionary degeneration of the genomes of the related tumor cells.[15]

Thus far, our analysis is based mostly on pattern matching with limited concrete biology in the form of mechanistic models of the creation of the viral phenotype. Methods from experimental virology and systems biology can be used to generate data that facilitates development of such models. Incorporating them into the prediction of viral

resistance and therapy effectiveness should increase the accuracy of the relevant prediction procedures and help further our understanding of how the viral phenotype develops.

Finally, though not included in our present analysis, host factors, including a patient's immunotype, also play a role in disease development and the effectiveness of drug therapy. For instance, it is under debate whether the immune system initially suppresses the enrichment of preexisting X4-viruses in the viral quasi-species. If this is the case, solely detecting X4 minorities need not be clinically significant; such detection does not necessarily predict the breakthrough of the viral variants, as long as the immune system is intact. Indeed, we and others have observed that the risk of X4-virus emerging rises with decreasing immune-cell count, reflecting the decreased intensity of the patient's immune response. Such observations strongly encourage construction of a comprehensive model that includes information on all three players—pathogen, drug, and host.

Acknowledgments

The work reported here is the result of extensive interdisciplinary collaboration. We thank all involved scientists, past and present, especially the computational biologists Niko Beerenwinkel and Tobias Sing, the Arevir consortium, especially Eugen Schülter, Martin Däumer, and Hauke Walter, and the Euresist consortium, especially, Francesca Incardona, Maurizio Zazzi, and Anders Sönnerborg. The work has been partially funded by Deutsche Forschungsgemeinschaft, grant Ho 1582/1-3, KA 1569/1-3 (Arevir) and EU grants LSHG-CT-2003-503265 (BioSapiens) and IST-2004-027173 (EuResist) and is being partially funded by BMBF grant 0315480 C (HIV Cell Entry), BMG grant 310/4476 (RESINA), and EU grant HEALTH-F3-2009-223131 (CHAIN). C

References

1. Altmann, A. et al. Predicting the response to combination antiretroviral therapy: Retrospective validation of geno2pheno-THEO on a large clinical database. *Journal of Infectious Diseases 199*, 7 (Apr. 2009), 999–1006.
2. Altmann, A. et al. Improved prediction of response to antiretroviral combination therapy using the genetic barrier to drug resistance. *Antiviral Therapy 12*, 2 (2007), 169–178.
3. Beerenwinkel, N. et al. Learning multiple evolutionary pathways from cross-sectional data. *Journal of Computational Biology 12*, 6 (July/Aug. 2005), 584–598.
4. Beerenwinkel, N. et al. Estimating HIV evolutionary pathways and the genetic barrier to drug resistance. *Journal of Infectious Diseases 191*, 11 (June 2005), 1953–1960.
5. Beerenwinkel, N. et al. Geno2pheno: Estimating phenotypic drug resistance from HIV-1 genotypes. *Nucleic Acids Research 31*, 13 (July 2003), 3850–3855.
6. Beerenwinkel, N. et al. Diversity and complexity of HIV-1 drug resistance: A bioinformatics approach to predicting phenotype from genotype. *Proceedings of the National Academy of Science USA 99*, 12 (June 2002), 8271–8276.
7. Christianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, U.K., 2000.
8. Desper, R. et al. Inferring tree models for oncogenesis from comparative genome hybridization data. *Journal of Computational Biology 6*, 1 (Spring 1999), 37–51.
9. Fields, B.N., Knipe, D.M., and Howley, P.M. *Fields' Virology, Fifth Edition*. Wolters Kluwer Health/Lippincott Williams & Wilkins, Philadelphia, PA, 2007.
10. Johnson, V.A. et al. Update of the drug-resistance mutations in HIV-1. *Topics in HIV Medicine 16*, 5 (Dec. 2008), 138–145.
11. Landwehr, N., Hall, M., and Frank, E. Logistic model trees. *Machine Learning 59*, 1–2 (May 2005), 161–205.
12. Lengauer, T. et al. Bioinformatics prediction of HIV co-receptor usage. *Nature Biotechnology 25*, 12 (Dec. 2007), 1407–1410.
13. Lengauer, T. and Sing, T. Bioinformatics-assisted anti-HIV therapy. *Nature Reviews Microbiology 4*, 10 (Oct. 2006), 790–797.
14. Margulies, M. et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature 437*, 7057 (Sept. 15, 2005), 376–380.
15. Rahnenführer, J. et al. Estimating cancer survival and clinical outcome based on genetic tumor progression scores. *Bioinformatics 21*, 10 (May 2005), 2438–2446.
16. Rhee, S.Y. et al. Human immunodeficiency virus reverse transcriptase and protease sequence database. *Nucleic Acids Research 31*, 1 (Jan. 2003), 298–303.
17. Roomp, K. et al. Arevir: A secure platform for designing personalized antiretroviral therapies against HIV. In *Proceedings of the Third International Workshop on Data Integration in the Life Sciences* (Hinxton, U.K. July 20–22). Springer Verlag, Berlin, Heidelberg, 2006, 185–194.
18. Rosen-Zvi, M. et al. Selecting anti-HIV therapies based on a variety of genomic and clinical factors. *Bioinformatics 24*, 13 (July 2008), 399–406.
19. Svicher, V. et al. Involvement of novel human immunodeficiency virus type 1 reverse transcriptase mutations in the regulation of resistance to nucleoside inhibitors. *Journal of Virology 80*, 14 (July 2006), 7186–7198.
20. UNAIDS. *2008 Report on the Global AIDS Epidemic*. UNAIDS, Geneva, Switzerland, 2008; http://www.unaids.org/en/KnowledgeCentre/HIVData/GlobalReport/2008/2008_Global_report.asp

Thomas Lengauer (lengauer@mpi-inf.mpg.de) is Director of the Department of Computational Biology and Applied Algorithmics at the Max Planck Institute for Informatics, Saarbrücken, Germany.

André Altmann (altmann@mpi-inf.mpg.de) is a staff scientist in the Department of Computational Biology and Applied Algorithmics at the Max Planck Institute for Informatics, Saarbrücken, Germany.

Alexander Thielen (athielen@mpi-inf.mpg.de) is a staff scientist in the Department of Computational Biology and Applied Algorithmics at the Max Planck Institute for Informatics, Saarbrücken, Germany.

Rolf Kaiser (rolf.kaiser@uk-koeln.de) is a staff scientist at the Virological Institute, Köln, Germany.

DOI:10.1145/1666420.1666441

Student participation and resulting expertise is as valuable as having the high-performance resource itself.

BY CAMERON SEAY AND GARY TUCKER

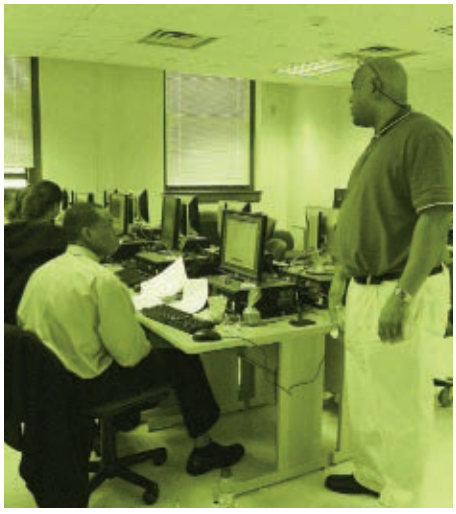
Virtual Computing Initiative at a Small Public University

THE VIRTUAL COMPUTING Lab at North Carolina State University (NC State), established August 2004, proved that the concept of a highly scalable, high-performance computing (HPC) resource providing on-demand applications anywhere/anytime had become a reality in this particular educational setting.

The VCL allows platform-independent access to a variety of computing configurations without having to maintain each one separately. This is done through software images installed (on demand by users) onto blade servers. The result is a highly scalable computing environment that allows users to use what they need when they need it.

The VCL was a groundbreaking project, in that it used entirely open source tools to dramatically increase the accessibly of computing resources for students, but the costs incurred are beyond the means of most smaller universities. However, in light of lessons learned, we now know that a much more affordable implementation is possible. Here, we offer a case study of a follow-on VCL pilot project at North Carolina Central University (NCCU), an historically black college in Durham, NC. But NCCU has less than a third the number of students as NC State while also being a liberal arts college (with a science focus), not an engineering school like NC State. By leveraging NC State expertise, we showed that such technology is within reach of practically any educational institution.

Virtualization is a key component of the way applications are deployed and



used today.[2] Users are no longer tied to a particular locale or limited by a particular workstation environment, and organizations are no longer limited to applications that use platforms commensurate with the expertise of their IT-support staffs. For example, professors who need applications to run on Linux need not be concerned if their universities' IT staffs include all authorized Linux administrators. Virtualization allows operating environments to be simulated in a way that does not require in-house expertise in the environment being used.[5] What is required are users who know the applications they'll be using and their proper configurations. The VCL project at NC State is a large-scale, publicly accessible example of a virtualization application in education,[1] providing transparent access to dozens of applications used by students and their professors in virtually every discipline in the university. It has dramatically altered the way students and faculty access the school's computer resources.

Founded in 1910, NCCU today has an enrollment of approximately 8,500 students. As a liberal arts school with a science focus, it includes the Biomanufacturing Research Institute and Technology Enterprise and the Biomedical and Biotechnology Research Institute and so has an ongoing need to manage diverse computing environments to support their various research projects. The VCL project represented a good model for addressing NCCU computing needs.

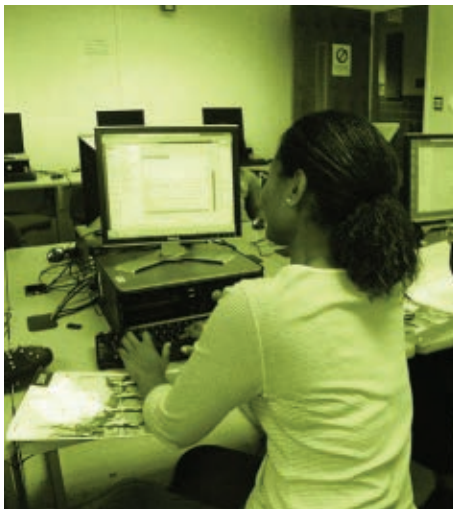
We attended a fall 2004 technology conference in Research Triangle Park, NC, where virtualization was covered. Researchers from NC State and Duke University described a project that allowed an infrastructure built on a Linux, Apache, PHP, MySQL (or LAMP) software base installed on blade servers to host multiple research projects on different platforms. For example, if one faculty member had a project requiring a Windows-based server and another had a project requiring a Solaris-based server, both projects could be hosted on the same infrastructure through virtualization of the respective operating systems. While not newcomers to virtualization, we were nonetheless impressed by how blade servers added a higher level of scalability. We began to seek out relationships with researchers and major technology companies in the Research Triangle Park area to determine how NCCU might get involved in the flow of this innovation.

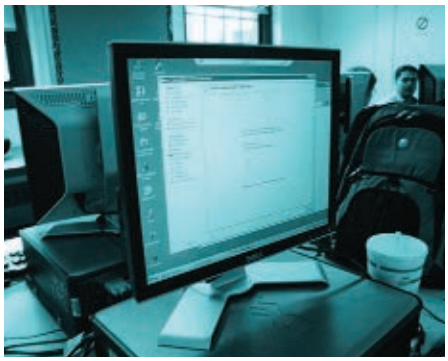
Over the next two years we received a series of hardware grants and cash awards that allowed NCCU to build a simple blade-server infrastructure suitable for a virtualization project. We were introduced to the VCL project in the College of Engineering at NC State where blade servers and virtualization, as well as stored images containing software components, were installed directly on the blades. Strictly speaking, using a full image on a blade is not

» key insights

- **Though virtualization is not new, the VCL provides greater access to computer applications.**
- **The VCL is a sophisticated application that may be prohibitively expensive to install from scratch; NCCU is thus leveraging NC State's expertise to develop a VCL project of its own.**
- **Virtualization has great potential on mainframes, and the NCCU VCL pilot system aims to extend itself to a mainframe platform.**
- **The VCL can be deployed by institutions of any size.**

PHOTOGRAPHS COURTESY OF NORTH CAROLINA STATE UNIVERSITY



"virtualization" per se, but the "virtual" designation fits in that users access the application remotely, not on their local computers. The project's most notable innovation was software-driven management logic that provides resources as needed and allows unused resources to be used for HPC applications, including molecular analysis. Students could use any Web browser with access to adequate bandwidth (at least 125kbs) to connect to dozens of desktop applications anywhere/anytime.

We intended to deploy this innovation at NCCU using blade servers—ultra-thin computers with multiple high-end processors—in a highly flexible "one-stop-shop" infrastructure to provide the same service to our students and faculty. With the two major biotechnology centers at NCCU, along with the university's focus on scientific computing, we felt the NC State approach would also be appropriate for NCCU—use the blades to run virtualized applications when needed but apply all idle processing time to long-running, processor-intensive scientific applications.

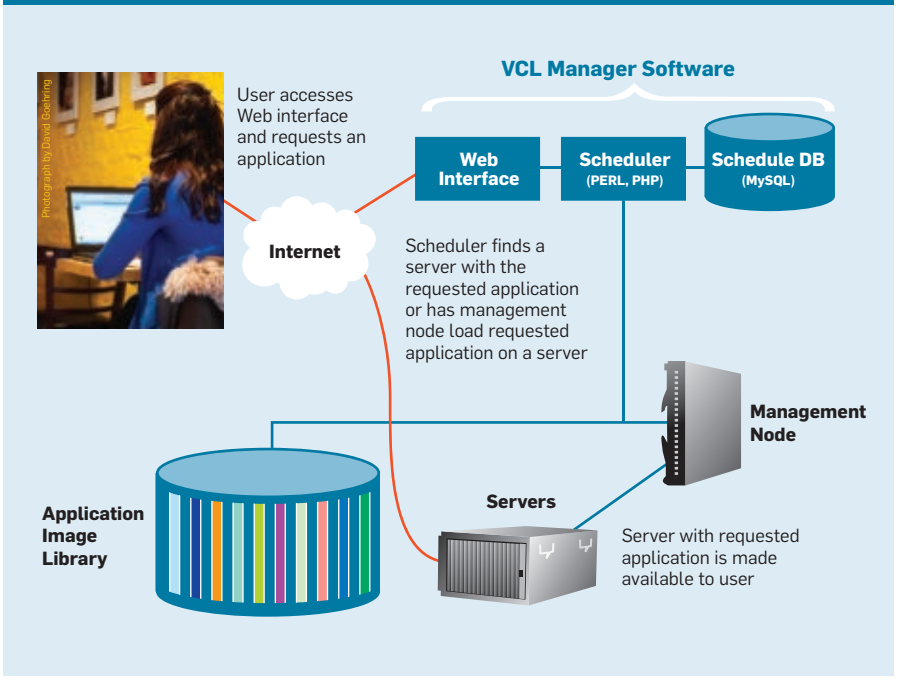
The goal was a scalable, reliable infrastructure for both virtualization and HPC applications. Toward this end, we began by purchasing nine blade servers. In fall 2005, we received a hardware grant from IBM (www.ibm.com/university) providing $84,000 for hardware, though nothing for software or support. Most was apportioned to the infrastructure to support the blades, leaving little for the blades themselves. For example, the rack required to host the chassis for the blades cost $2,649. The network switch for the blades cost $10,000. The monitor, keyboard, and video and monitor connector cost $2,245. After we ordered these foundational pieces, there was funding enough for nine initial blades. We chose IBM HS20 Xeon blade servers with 4GB of RAM and two 3.8GHz processors per server. Each server also had two 36GB mirrored hard drives. The cost per blade, with extra processor, memory, and hard drive, was approximately $6,106.

One technical lesson we learned quick was there really is no need to mirror the hard drives in the blade servers when using them for virtualization and that it is better to spend our grant money on a faster processor and more memory. The failure rate of the IBM blade servers was low, and if students were backing up their work to permanent storage (as we advised them to do), the unlikely event of a drive failure would be only an annoyance, while the students secured another session and resumed their work. No drive failure during any session has been reported by NCCU users.

In 2006, we were awarded a second grant from IBM for the same amount. This time we were able to spend more of the grant money on blade servers. Having learned we did not have to mirror the hard drives, we were able to purchase higher-end blade servers at lower cost. This time, we chose 14 IBM HS21 blade servers (a later model from the previous year) with 72GB hard drives, two 3.8GHz processors, and 4GB of memory. We then had a total of 23 blades in the two chassis.

Staff from NC State and NCCU Information Technology Services (ITS) assisted in the installation of the initial blades. Our intention was to adhere to the NC State LAMP standard as closely as possible, the rationale being that we wanted to leverage the existing knowledge base, deviating from the standards only when absolutely necessary. However, circumstances prevented rigid

Figure 1. NC State VCL infrastructure.



adherence to the NC State installation model. The standard operating system (OS) NC State used was the Red Hat Enterprise License (RHEL) distribution of Linux. While the State of North Carolina has a licensing agreement with Red Hat to use RHEL, we could not get clarity as to how we might properly use the license for our installation. We decided on SuSE 10.1 (available for free at the time) as our OS because one of the consultants working with us was familiar with it, and in our judgment Red Hat Fedora (another free distribution) was not appropriate for our purposes.

As a physical location for the equipment, we chose a data center in Research Triangle Park that already housed many North Carolina University system projects, including much of the VCL infrastructure. Though the decision had not been formalized at the time of our 2006 installation, we knew the data center had sufficient power, cooling, and bandwidth, and its staff was familiar with the equipment the NCCU team would be using. The initial cost for housing the blade center, including power and bandwidth, was approximately $600/month, which we considered reasonable.

The project's driving theme was the use of existing expertise to extend both the VCL footprint and NCCU's computing capability. We knew that virtualization per se is a somewhat mature technology we had not previously tapped due to the limits of NCCU staffing and expertise. Our collaboration with technology companies (one of which, IBM, provided the hardware grant), the data center (which hosted the project as funding was worked out and provided unpaid assistance), and other universities in the area (an endless supply of innovation and expertise) is as much the story here as the technology of virtualization. The model we used thus represents a template for a bare-bones venture into virtualization technology by institutions otherwise lacking the resources to do so. This will prove invaluable for those in remote areas, like rural school districts, and those with limited financial resources, like many in North Carolina today.

The NC State VCL uses blade servers for hardware and a LAMP open source environment for software. Red Hat Enterprise License is the core OS, though we also still use SuSE 10.1. We emulated this environment due to its stability, flexibility, and scalability—characteristics of an infrastructure suited to our purposes. Our resources were limited; we initially (in 2005) lacked funding other than what we received to purchase hardware. But one VCL merit is a system built (basically) with open source software. Meanwhile, the LAMP environment has proved itself robust enough for even the most demanding use in terms of number of users, application size, hours of operation, and performance.

Figure 1 outlines the hardware used to deploy the VCL. Applications are deployed using either full images of the application (including OS and peripheral files) installed directly on an empty blade or virtual machines via VMware. These images are stored in an image library until the scheduler calls them to be loaded. The scheduler, consisting of code written in PERL and PHP, communicates with the MySQL instance containing the data for the system. The entire infrastructure sits on racks of blade servers maintained at a central location in Research Triangle Park; some parts are in NC State's facility in Raleigh, NC, but over the next few years all VCL equipment is expected to be located in the Research Triangle Park facility.

Users access the VCL from their personal locations, whether school, lab, home, or during travel. What they need is some form of broadband connection (as little as 128kbs is sufficient, though at least 256kbs is recommended), a computer with TCP/IP (properly configured), and a Web browser. The VCL uses student and faculty institutional email for authentication, an approach that facilitates the VCL's extensibility. Users external to NC State do not need to maintain separate VCL authentication identities, selecting whatever identity they normally use to get onto their own organizations' networks. Users authenticating to the VCL see a screen like the one in Figure 2.

From that initial screen the users select the application (image) they want to use, the time they want to use it, and the duration of their session. When using applications, they select a remote-access client (varying by OS) to attach to and use the image as if it were their regular desktop. Performance is excellent; only changes to the desktop screen traverse the network, with all processing occurring on the blades or on whatever high-end hardware houses the VCL not otherwise married to a blade configuration.

The NC State implementation is robust and stable, but building a similar application from scratch would be prohibitively expensive for smaller institutions. What emerged is a methodology

that allows any institution, irrespective of size, to use the system. At present, NCCU is able to use it in day-to-day functions, somewhat separate from the NC State VCL team. While support from NC State is still required, it decreases with the passing semesters. As the pilot project expands and is viewed as successful, the NCCU goal is to be completely autonomous in terms of hardware, software, and maintenance, though such autonomy is not strictly necessary. Whether an institution wants to run its own part of the VCL or have it run by NC State varies by institutional mission. For NCCU, having its technology students fully understand the VCL is almost as important as the value it gets from using the tool itself; on the other hand, a middle-school English class might need access only to word processing software. Having the mission of each institution drive the configuration of the tool highlights the VCL's flexibility and fitness as an educational resource.

The evolving VCL model involves a loose confederation of user organizations consisting of several colleges and community colleges in the University of North Carolina system. Some purchase their own blades, deploying them in racks in the data center housing VCL equipment; some use the existing equipment and just add users to infrastructure already deployed. A statewide VCL network will eventually include K–12 school systems in North Carolina, nonprofit corporations, and other organizations in need of the functionality the VCL provides but lacking the means and expertise to deploy themselves.

NCCU has partnered with two high schools that exemplify the VCL's eclectic nature; one is interested in the VCL primarily as a tool for teaching networking concepts, the other more in the easy access to the software it provides. Both serve predominantly African-American student populations that would benefit tremendously from being exposed to VCL innovation.

What about licensing? On the surface it might seem that licensing would add considerable complexity to VCL deployment. However, none of the VCL partners have found this to be the case. All that is required of any institution is a clear understanding with the software vendor that access to its products conforms to the agreed-upon license; access logs provide ready confirmation that the terms of the license have been followed. For certain products (such as widely used statistical software), we meet with the vendor to establish the processes it finds acceptable for accessing its product. For individual licenses, VCL staff maps users to software per the product's license agreement. While some up-front organization is required, licensing is not a major hurdle.

What about VCL bandwidth requirements? Because users access software remotely (the application is not on the user's local system), use of bandwidth does increase somewhat. Programs that are graphically intensive (such as engineering design) send more packets than less graphically intensive programs (such as word processing). However, during NCCU pilot development we found neither performance degradation on NCCU's network nor severe performance issues on graphically intensive programs (such as AutoCAD). The local high schools use Alice, a 3D programming environment that uses graphics extensively; its performance via the VCL is more than acceptable. A more challenging test, perhaps, is a more graphically intensive business simulation, like IBM's business-process simulation Innov8 (http://www-01.ibm.com/software/solutions/soa/innov8/index.html). Because it requires high-end graphics capability on the monitor (a problem not directly related to the VCL), we have been unable to test its effect on the VCL environment because the graphics cards in our lab machines don't support the required graphics. We're eager to see how the VCL handles such applications when the monitors on those machines are upgraded to render graphics for programs like Innov8.

Discussions among faculty and administrators at both NCCU and NC State, along with the local technology companies funding project hardware and the NCCU data center staff were necessary for planning how to use the VCL at NCCU. All the administrators seemed to understand and recognize the value of the project, giving it their full support. Such support is vital to any technology project. An early question involved who would pay to house the equipment. The funding grant NCCU received for the project was limited to hardware. Ultimately, the NCCU School of Business (with permission of its Dean) absorbed the initial housing costs for the equipment.

We installed SuSE 10.1 on the first nine blades in early 2006. That spring we began to poll the NCCU community about what applications it used that might be deployed on our blades. The first was Web MO, a chemistry program for molecular analysis, complementing

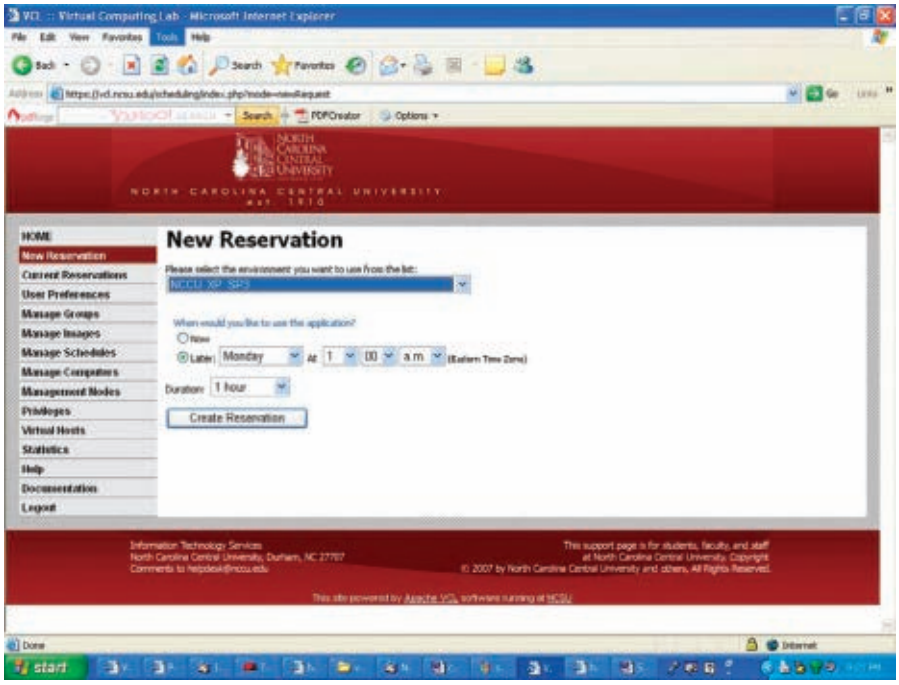


Figure 2. VCL user interface.

our plan to use the processing power of the blades to run scientific programs when not used for virtualization. A blade was dedicated to classes in both the NCCU School of Business and School of Library and Information Science. Classes using these blades were graduate database classes (Library and Information Science), programming and database classes (School of Business), and Web development (School of Business). The entire university could access the resource without interfering with the production infrastructure managed by NCCU ITS. These parts of the project were completed in summer 2006.

Beginning fall 2006, the NC State VCL team began making presentations to faculty and staff beyond the NC State campus, explaining the promise of the VCL. The NCCU Provost attended one and gave the project her full support. NCCU administrators determined that a formal VCL pilot project was warranted for roll-out in fall 2007. This included NCCU's willingness to pay the cost of housing its blade center. We cannot stress enough how important it was for this to be perceived as an NCCU-wide project not restricted to NCCU's School of Business. The greater the VCL footprint, meaning the larger the number of users, the greater would be the value to the entire university.

Once the CIO, Provost, and Dean of the School of Business were in agreement that NCCU would develop the VCL, members of the NCCU VCL team (now NCCU ITS) were assigned to identify the applications the VCL would support. So we asked NCCU faculty what applications they felt would provide the greatest initial benefit. The consensus was the statistical package SAS. One reason for this choice was that the more popular standalone version of the package had to be installed on every workstation that uses it. Since the application is updated regularly, manually updating every workstation is prohibitively labor-intensive, in spite of imaging software. If the application could be accessed centrally, this would produce a considerable cost saving in terms of labor. In addition, students in classes requiring SAS usually come to campus to do their assignments because the software is too expensive for them to purchase on their own, and the SAS academic license requires it be installed only on NCCU computers. The VCL would allow them to access the software remotely.

One technical lesson we learned quick was there really is no need to mirror the hard drives in the blade servers when using them for virtualization and that it is better to spend our grant money on a faster processor and more memory.

The CIOs of NC State and NCCU met with SAS management to determine how SAS could be accessed through the VCL. SAS agreed to allow faculty and students at NCCU, NC State, and other affiliated users who follow all requirements of the SAS licensing agreements to access the product through the VCL.

Now the problem was how to let NCCU users actually aaccess it. The NCCU team lacked the expertise to manage the system's scheduling of images and virtual machines but did not want to add to NC State's administrative burden by having to routinely manage NCCU users. The VCL teams at NC State and NCCU agreed that the NC State infrastructure would serve as a base for other organizations so the scheduling and management logic of the NC State VCL engine would serve additional participants. As resources were acquired and added to the infrastructure (to be centrally located in the Research Triangle Park data center), for the time being, management logic would still be centralized. As the VCL code base was made portable (not the case initially), other entities would be able to use the code to manage their own parts of the infrastructure. The blade configuration was such that all chassis added to it could be managed centrally through a "management module" on each rack, an approach providing maximum extensibility.

Another positive development was IBM awarding an additional hardware grant to NCCU in fall 2006. As we had already purchased much of the infrastructure to house additional blades, we used the new grant to purchase 14 more blades. This time, we chose Centos 5 as the OS. Though the RHEL license issue was still not resolved, the Centos design is very much like RHEL, and we felt this would help move NCCU toward a standard configuration.

One thing apparent early in this process was that a facilitator, or person with direct interest in VCL adoption, is essential. While necessary, academic departments, administrators, and technical managers "getting it" is not sufficient. The project repeatedly ran into obstacles that would have been fa-

tal without the required commitments. Even an executive champion is not sufficient. A facilitator must be able to build communication channels, provide or find expertise, help the organization with funding, and act as handholder/cheerleader as the organization grows into the VCL.

One reason the facilitator is so essential is that almost invariably several entities are involved. The VCL is not an application whose value is best derived through its use in one or two departments. The VCL's greatest value is when an entire institution, across functional units and academic disciplines, uses it to seamlessly access computing resources.

The facilitator (like champions in other technology projects) must present the idea to both administrators and technical people as something desirable, as well as doable. The technology itself is not daunting to most IT shops; virtualization is not new. But to the staff it could be seen as extra work. The facilitator must address this perception, making it clear that the VCL means less work, not more.

It's entirely possible that convincing a university's president, provost, and/or CIO is not sufficient; there may be resistance from department heads and technical leads who see the VCL as problematic. The facilitator can identify and help articulate the most important value-adds; one is that the VCL flattens the hardware landscape so different labs access the same application on the same platform, for the most part irrespective of the configuration of the workstation accessing the VCL.

In larger organizations, the facilitator convincing a dean to try the VCL may indeed be able to marshal the necessary resources but must still ensure all parties follow through; in this context the facilitator is more like a project manager. For smaller organizations, however, the facilitator may do everything, from moving the organization's mail identities into a Lightweight Directory Access Protocol structure to finding funds to purchase software, to creating the images to be used through the VCL.

By fall 2009, the NCCU VCL pilot had been in effect two full academic years, serving several targeted areas of the university. The heaviest users were from the following programs: Computer Information Systems (CIS), Decision Science, Marketing, and Hospitality and Tourism, and the School of Library and Information Science. In addition, special licensing was arranged for a local high school to access an application via NCCU's license with the application's publisher.

While the duration of the pilot was never specified, we (the authors) were comfortable that it is now ready to put into production. However, a delay was encountered (a change of CIO at NCCU), and such a campuswide project cannot be accomplished until a new CIO is in place. We do not know when this will occur.

Meanwhile, from September 1, 2007 to December 31, 2009 a total of 11,529 reservations were submitted to use the VCL; total usage time of all applications was 14,645 hours, with 947 unique users. The most popular images/virtual machines were courseware for a CIS course (2,488 reservations, 328 unique users during the pilot project); SAS (1,657 reservations, 191 unique users); an Alice image at Hillside High School (1,174 reservations, 109 unique users); an Office 2007 image at Hillside High School (1,082 reservations, 86 unique users); an image of the statistical package SPSS (771 reservations, 99 unique users); and tools to access a mainframe for a course in mainframe technology (893 reservations, 34 unique users).

This usage profile shows that if the requested applications are provided through the VCL, users will use them. The NCCU VCL pilot project opened a new realm for virtual access to applications. NCCU is not an engineering school with layers of technology expertise. It has a highly competent but small ITS staff (in many ways the reason for the project's success). Most of our expertise extended from NC State, which has shared lessons learned with us.

Our (the authors) role in the pilot project has been to put the right people in the right places at the right time. Through the business model that emerged from this process, we hope to bring other organizations into the VCL family.

Historically black colleges and universities, both public and private, can benefit from this model. We are in discussion with two larger technical HBCUs: Morgan State University in Baltimore, MD, and NC A&T State University in Greensboro, NC (both engineering schools). Southern University in Baton Rouge, LA, also an engineering school, recently received funding to begin its own VCL project. The NCCU VCL team is in discussion with the Southern VCL team to help facilitate the project.

Comments from faculty and students reflect the ease of VCL access. In each semester of the pilot, NCCU faculty used VCL for undergraduate and graduate classes requiring SAS software, mostly by students with no familiarity with the software; more than 300 participated. The related faculty had all taught SAS-related courses for years, so the content was familiar. The undergraduate students were, for the most part, newcomers to SAS. Most of the graduate students had some experience, but none could be considered an expert.

Each semester of the pilot project we've regularly asked all related professors whether their students reported difficulty accessing SAS software through the VCL. Other than ensuring students use the correct credentials to log into the system (NCCU student identities), we've received no reports of difficulty accessing the software when off campus. Moreover, none of the students with whom we spoke reported any difficulty accessing the software remotely, once they logged into the system once or twice.

Several students functioned as unofficial technical support for the software (more for SAS than for the VCL), greatly enhancing the experience of their less-proficient counterparts, at least according to the students with whom we spoke. These impromptu experts eased the novelty for novices logging in for the first time. One graduate student whose home was adjacent to campus served as a sort of coordinator for graduate novices logging in for the first time.

NCCU has benefited from the fact that the VCL makes complicated installations less burdensome when done concurrently in labs with dissimilar hardware configurations. NCCU ITS uses a set number of master images to deploy OSs and software applications to the labs. While the images are designed for lab heterogeneity, any application outside the existing

configurations could cause problems for ITS staff. An example is an application the CIS Department uses to teach students office-productivity software. Though not complicated to install, it is beyond the scope of the NCCU ITS standard image and involves ITS managers assigning staff to install it. The solution was to have CIS faculty using the application create an image, then have the students access it. The software publisher sees no violation of its license agreement, and NCCU ITS does not assign support staff to install and maintain the package. While this is a rather pedestrian example, it nonetheless caused considerable frustration on the part of both the CIS faculty and the NCCU ITS staff.

Because the NCCU VCL pilot project could count on NC State's extensive expertise and resources, we had little concern that the pilot project's workload would overwhelm our blades. Because we deployed both virtual machines and "bare metal" images (loaded directly onto the blade servers), we addressed any limitation in our resources through NC State's much more robust infrastructure. NC State had been the recipient of both cash grants and hardware donations to support an environment with hundreds of blade servers.

Because we used NC State's node-management software, it was a small matter to direct users to available resources; we have no record of a user lacking available resources to use an image. We calculated that if we could dedicate all 14 blades, we could also support four virtual machines per blade, for a total of 56 concurrent users; most of the applications are not computationally intensive, and we were able to distribute computationally intensive applications, like SAS, across several blades to balance demand on the processor. At no time did demand from the pilot approach this requirement. Our focus was not so much on maximum utilization of resources (though with virtualization this goal is a given) as it was on determining the level, quality, and type of service that would be required for faculty and staff to use the infrastructure.

We planned to address demand in several ways:

- Use the large donation of processors and cash to NC State by IBM and Intel to provide enough resources so the entire 16-campus University of North Carolina system could use the VCL the same way NCCU and NC State use it;
- Provide autonomy to NCCU and not strain the NC State staff for expertise, limiting the pilot until funding was available; and
- Porting the VCL to IBM's system z platform, running it on zLinux. Several companies have offered to support such an effort, but how to run Windows applications smoothly on the z/OS platform is not fully resolved.

It appears that the first approach is the most likely near-term solution because much of the cost of a mega-blade center would be spread across the entire University of North Carolina system, covering power, hardware, and staff.

Of considerable interest to many educators is the possibility of also bringing K–12 organizations into the VCL.[3] The application is ideal for technologically and financially limited schools serving lower-income students.[6] NCCU is involved with two high schools in the Research Triangle Park area, one using the VCL, the other expecting to by fall 2010. These schools will be an excellent test case, further demonstrating the VCL's ability to flatten the technology playing field.

Another compelling VCL development is the possibility of porting some functionality to a mainframe environment. Virtualization technology had an early home there, beginning 40 years ago with the need to replicate a development environment without having to add separate, prohibitively expensive machines. IBM's zVM OS was an early answer to this demand. With the advent of sophisticated partitioning technologies in the 1980s, zVM became much less essential, and the technology was almost retired; x86-based virtualization then gave zVM new life. IBM claims its z10 processor (announced in February 2008) can lower the cost of energy by 85% compared to x86 processors doing the same work.[4] While only the Linux/Unix-based applications are candidates for migration from the VCL to mainframes (due to instruction-set compatibility issues between zOS and Windows). Even if only the Linux/Unix VCL applications are run on the z10, it

will be interesting to see if such impressive cost savings can be realized.

Conclusion

The NCCU VCL pilot is an extension of the VCL project begun at NC State and owes full attribution to the NC State team for its innovative work. But as users of the VCL, we aim to drive it to places where it might not go of its own accord. The VCL began at a public university; implicit in such work is the ethical obligation to allow the public to avail itself of its benefit, with the consent of its creators at NC State. Extensions of the VCL, including the NCCU pilot, are themselves innovations, because what is needed (once the science and engineering issues are addressed) is a replicable business model. The VCL has now been extended to NCCU, a relatively small public university, a significant accomplishment available to other organizations without large technology staffs.

VCL mainframe implications are significant. The VCL runs well on a distributed platform. All reports of performance and reliability in the current blade environment are positive. But virtualization has been part of the mainframe domain (such as IBM's zVM) for decades. That it works is an understatement. To be able to have hundreds, even thousands, of virtual servers running on a mainframe with accompanying dramatic reduction in power demand is a development we are eager to see.

The VCL project is important to virtualization technology in education for four main reasons:

- Though it began in the College of Engineering at NC State, and NC State provides most of the technical direction, the project is developing an increasingly eclectic profile. Participants are able to apply innovation to the hardware and software infrastructure and still enjoy the benefits of being part of the VCL environment. For example, if a high school wants to use a homegrown virtualization solution (perhaps for instructional purposes) and didn't use the VCL per se, accommodations could be made for it to use VCL management logic and networking, as long as its solution does not impede other users;
- For NCCU, development of expertise among internal staff and students is invaluable. We are learning to deploy our production infrastructure more efficiently, and our students are acquiring a valuable and marketable skill set involving virtualization;
- For the predominantly African-American community served by NCCU, the related technology transfer is especially welcome. As we work with high schools in our area, the community at large is involved directly in technological innovation at a much deeper level than it ever was before; and
- With the emergence of cloud computing,[7] the VCL might also serve as a major cloud application, becoming yet another software service for the world at large while delivering services from commercial vendors.

The VCL's greatest value is when an entire institution, across functional units and academic disciplines, uses it to seamlessly access computing resources.

Though commercial solutions may provide the same or similar results for the same or lower cost, they don't (as far as we see) allow our extended community (particularly in North Carolina) to directly participate in the ongoing innovation. However, in many ways this participation is as vital to us as the benefit derived from the technology itself.

Acknowledgments

Our thanks to NC State, Duke University, International Business Machines, and MCNC in Research Triangle Park, NC.

References

1. Averitt, S. et al. Virtual Computing Lab. In *Preliminary Proceedings of the International Conference on the Virtual Computing Initiative* (Research Triangle Park, NC, May 7–8, 2007), 1–5.
2. Barham, P. et al. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review 37*, 5 (Dec. 2003), 164.
3. Gaspar, A. et al. The role of virtualization in computing education. ACM SIGCSE Bulletin 40, 1 (Feb. 2008), 131–132.
4. Kusnetzky, D. IBM announces the z10: Is the mainframe still relevant? ZDNet (Feb. 18, 2008); http://blogs.zdnet.com/virtualization/?p=351&tag=rbxccnbzd1
5. Leitner, L. and Cane, J. A virtual laboratory environment for online IT education. In *Proceedings of the Sixth Conference on Information Technology Education* (Newark, NJ, Oct. 2005), 283–289.
6. Miller, K. and Pegah, M. Virtualization: Virtually at the desktop. In *Proceedings of the 35th Annual ACM SIGUCCS Conference on User Services* (Orlando, FL, Oct. 7–10). ACM Press, New York, 2007, 255–260.
7. Ricadela, A. Computing heads for the clouds. *BusinessWeek* (Nov. 16, 2007); http://www.businessweek.com/technology/content/nov2007/tc20071116_379585.htm

Cameron Seay (cseay@nccu.edu) is an assistant professor of computer information systems in the School of Business at North Carolina Central University, Durham, NC.

Gary Tucker (gtucker@mail.nccu.edu) is a technology analyst at Bank of America in Charlotte, NC.

DOI:10.1145/1666420.1666442

Computer scientists have made great strides in how decision-making mechanisms are used.

BY VINCENT CONITZER

Making Decisions Based on the Preferences of Multiple Agents

PEOPLE OFTEN MUST reach a joint decision even though they have conflicting preferences over the alternatives. Examples range from the mundane (such as allocating chores among the members of a household) to the sublime (such as electing a government and thereby charting the course for a country). The joint decision can be reached by an informal negotiating process or by a carefully specified protocol.

Philosophers, mathematicians, political scientists, economists, and others have studied the merits of various protocols for centuries. More recently, especially over the last decade, computer scientists have also become deeply involved in this study. The perhaps surprising arrival of computer scientists on this scene is due to a variety of reasons, including the following:

1. Computer networks provide a new platform for communicating preferences. Examples include auction Web sites, where preferences are communicated in the form of bids, as well as Web sites that allow one to rate everything from the quality of a product to the attractiveness of a person.

2. Within computer science, there is a growing number of settings where a decision must be made based on the conflicting preferences of multiple parties. Examples include determining whose job gets to run first on a machine, whose network traffic is routed along a particular link, or what advertisement is shown next to a page of search results.

3. Greater computing power and better algorithms, as well as a more computational mind-set in the general public, have made it possible to run computationally demanding protocols that lead to much better outcomes. An example is an auction in which bidders can bid on arbitrary sets of items, rather than just on individual items (I will discuss such auctions in more detail later). Such protocols were once considered theoretical niceties that could never be run in practice (to the extent they were conceived of at all), but now they are actually practical.

4. The paradigms of computer science give a different and useful perspective on some of the classic problems in economics and related disciplines. For

» key insights

- **Computer scientists are contributing to and making use of microeconomic theory.**
- **Better algorithms enable new marketplaces and other mechanisms that lead to increased economic efficiency.**
- **Game theory and mechanism design can be used to analyze and address the problem of strategic users.**

example, various results in economics prove the existence of an equilibrium, but do not provide an efficient method for reaching such an equilibrium.

In this article, I give a (necessarily incomplete) survey of topics that computer scientists are working on in this domain. I discuss voting and rank aggregation, task and resource allocation, kidney exchanges, auctions and exchanges, charitable giving, and prediction markets. I examine the problem of agents acting in their own best interest, which cuts across most of these applications. I also intersperse a few opinions and predictions about where future research should and will go.

Here, parties whose preferences we are interested in are not always human; they can also be, among other things, robots, software agents, or firms.[a] As is done in both computer science and economics, I use the term "agent" to refer to any one of the parties.

Settings Without Payments

I will discuss a variety of settings, so it is helpful to categorize them somewhat. An important aspect is whether the setting allows agents to make payments to each other (in some currency). For example, in a voting setting, we typically do not imagine money changing hands among voters (unethical behavior aside). On the other hand, in an auction, we naturally expect the winning bidder to pay for her winnings. First, I discuss various settings in which no money changes hands.

Voting and rank aggregation. A natural and very general approach for deciding among multiple alternatives is to vote over them. In the general theory of voting, agents can do more than vote for a single alternative: usually, they get to rank all the alternatives. For example, if

a In artificial intelligence, there is the study of multiagent systems, where agents—for example, robots—often need a protocol for coordinating on (say) a joint plan.

ILLUSTRATION BY JOHN HERSEY

a group of people is deciding where to go for dinner together, one of them may prefer American food to Brazilian, and Brazilian to Chinese. This person's vote can then be expressed as A ≻ B ≻ C.

Given everyone's vote, which cuisine should be chosen? The answer is far from obvious. We need a *voting rule* that takes as input a collection of votes, and as output returns the winning alternative. A simple rule known as the plurality rule chooses the alternative that is ranked first the most often. In this case, the agents do not really need to give a full ranking: it suffices to indicate one's most-preferred alternative, so each agent is in fact just voting for a single alternative.

Another rule is the *anti-plurality rule*, which chooses the alternative that is ranked *last* the *least* often. Now, it suffices for agents to report their last-ranked alternative—they are voting against an alternative. Which of these two rules is better? It is difficult to say. The former tries to maximize the number of agents that are happy about the choice; the latter tries to minimize the number that are unhappy. Another rule, known as the *Borda rule*, tries to strike a balance: when there are three alternatives, it will give two points to an alternative whenever it is ranked first, one whenever it is ranked second, and zero whenever it is ranked last. Many other rules, most of them not relying on such a points-based scheme, have been proposed; social choice theorists analyze the desirable and undesirable properties of these rules.

Rather than just choosing a winning alternative, most of these rules can also be used to find an aggregate ranking of all the alternatives. For example, we can sort the alternatives by their Borda score, thereby deciding not only on the "best" alternative but also on the second-best, and so on. There are numerous applications of this that are relevant to computer scientists: as an illustrative example, one can pose the same query to multiple search engines, and combine the resulting rankings of pages into an aggregate ranking.

One particularly nice rule for such settings is the *Kemeny rule*, which finds an aggregate ranking of the alternatives that "minimally disagrees" with the input rankings. More precisely, we say that a disagreement occurs whenever the aggregate ranking ranks one alternative above another, but one of the voters ranks the latter alternative above the former. The Kemeny rule produces a ranking that minimizes the total number of such disagreements (summed over both voters and pairs of alternatives).

While enabling the use of computationally demanding voting rules is valuable, I believe that in the near future, computer scientists will make much larger contributions to the theory and practice of voting.

The Kemeny rule has a number of desirable properties. For one, if we assume that there exists an unobserved "correct" ranking of the alternatives (reflecting their true quality), and each voter produces an estimate of this correct ranking according to a particular noisy process, then the Kemeny rule produces the maximum likelihood estimate of the correct ranking.[40]

Unfortunately, finding the Kemeny rule's output ranking is computationally intractable (formally, NP-hard).[3] Nevertheless, there are algorithms that can usually solve the problem in practice.[8] As an example, in Duke University's computer science department, we have used the Kemeny rule to find an aggregate ranking of our top Ph.D. applicants (based on the rankings of the individual admissions committee members); using the CPLEX solver, we found the Kemeny ranking more than 100 applicants in under a minute.

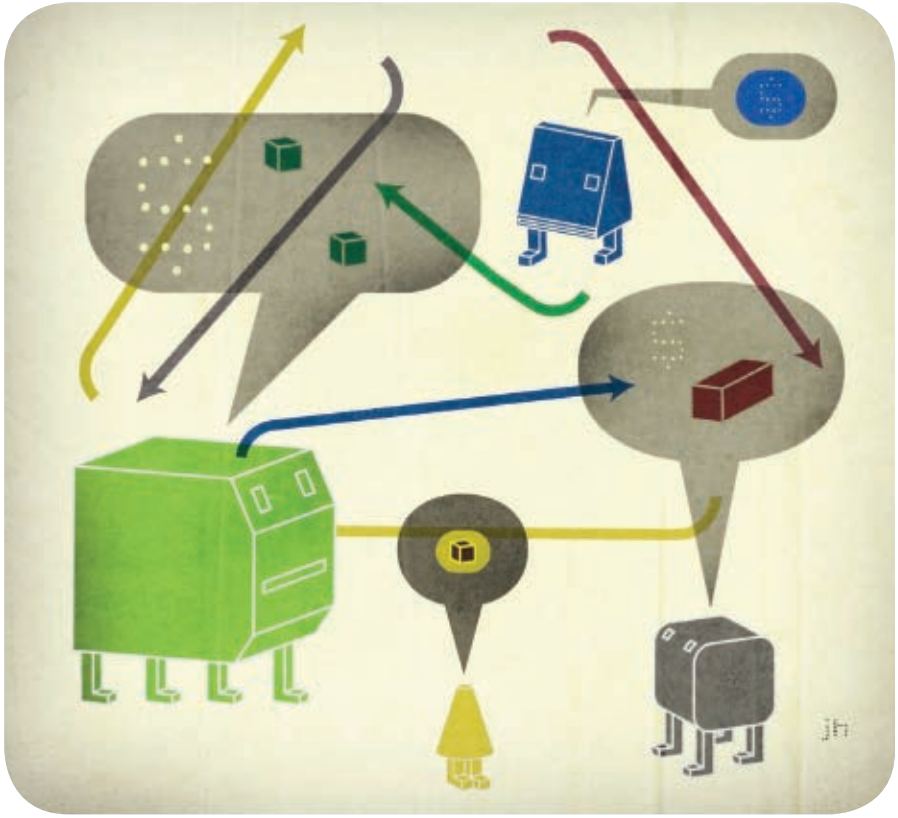
While enabling the use of computationally demanding voting rules such as the Kemeny rule is valuable, I believe that, in the near future, computer scientists (specifically, the computational social choice community) will make much larger contributions to the theory and practice of voting. Real-world organizations often need to make not just a single decision, but rather decisions on a number of interrelated issues. In our dining example, the agents need to decide not only on a restaurant, but also on the time of the dinner; and an agent's preferred restaurant may depend on the time of the dinner. For example, an agent may prefer not to start a heavy Brazilian steakhouse meal shortly before going to bed.

In some sense, the "correct" way of handling this is to make the alternatives combinations of a time and a cuisine, so that an agent can say: "I prefer an early Brazilian meal to a late Chinese meal to..." However, this straightforward approach rapidly becomes impractical as more issues are combined, because the number of al-

ternatives undergoes a combinatorial explosion. Ideally, the agents would have an expressive language in which they can naturally and concisely represent their preferences. One good language for representing such preferences is that of CP-nets[4] (which bear some resemblance to Bayesian networks). A CP-net allows a voter to specify that her preferences for one issue depend on the decisions on some other issues—for example, "If we are eating early, I prefer Brazilian; otherwise, I prefer Chinese." Given a language, we must design new voting rules that can operate on preferences represented in this language, as well as algorithms for running these rules.

While such combinatorial voting[20,38] is in its infancy, it is easy to see its potential value by considering how ad hoc the methods are that we use today for these types of situations. For example, members of Congress must vote on bills that address many different issues, and would often prefer to express preferences on individual issues. Unfortunately, voting on the individual issues separately can easily lead to undesirable results, because there is no guarantee that the issues are resolved in a consistent way. For instance, in the dining example, it may happen that most agents, in general, prefer to eat at a Brazilian steakhouse; and that, in general, most agents prefer to eat late; but most agents do not want to eat at a Brazilian steakhouse late at night. If they vote on the issues separately, the result may well be a late dinner at a Brazilian steakhouse. This is why the language for expressing preferences needs to allow the agents to specify some interactions among the issues.

Allocating tasks and resources. A voting scheme allows an agent to submit arbitrary preferences over the alternatives. While this generality is nice, in many settings, it is not needed, because we can safely make some assumptions about agents' preferences. Let us consider again the example of allocating chores in a household. One alternative might be: "Alice will vacuum and take out the trash, and Bob will do the dishes." It seems safe to assume that Bob will prefer this alternative to the alternative: "Alice will take out the trash, and Bob will vacuum and do the dishes," since the latter alternative gives Bob an additional task. On the other hand, if we are allocating desirable resources instead of cumbersome tasks, then presumably more is preferred to less. For example, if the agents jointly own a car, an alternative might be: "Alice gets to use the car on Friday, and Bob gets to use it on Saturday and Sunday," which Bob presumably prefers to the alternative "Alice gets to use the car on Friday and Saturday, and Bob gets to use it on Sunday." Here, the use of the car on a particular day is a "resource." These assumptions about preferences—receiving more tasks or fewer resources is never preferred—are commonly referred to as *monotonicity* assumptions.



Another reasonable assumption about preferences is that an agent only cares about which tasks or resources are allocated to her. For example, Alice is likely to be indifferent between "Alice gets the car on Friday, Bob on Saturday, and Carol on Sunday" and "Alice gets the car on Friday, Bob on Saturday and Sunday, and Carol never." In economics, the assumption that an agent, given her own resources and tasks, does not care about how the remaining resources and tasks are allocated to the other agents is known as the *no-externalities* assumption. It is not always completely accurate—Alice may dislike the alternative where Carol never gets the car slightly more, for example because Carol will ask Alice to run errands for her in that case—but it is usually assumed.

Reasonable assumptions such as these allow us to get away from the full generality of the voting model, and make decisions in a way that is more specific to task and resource allocation. Incidentally, there are many applications of task and resource allocation within computer science. For example, we may allocate time on a supercomputer (or other computing resources) instead of time with a car. Also, instead of allocating the chores of a household to its inhabitants, we may allocate jobs to machines.

So, how should we allocate tasks and resources? By far the most common approach to this is to assume that the agents can make or receive payments in some currency, which leads us to auction and exchange mechanisms. I will discuss such mechanisms in more detail later on, but for now, I first consider methods that do not require payments. These methods will generally try to find an allocation that is "fair" in some sense.

ILLUSTRATION BY JOHN HERSEY

One fairness criterion is *envy-freeness*: We should find an allocation such that every agent prefers her bundle (that is, the tasks or resources allocated to her) to each other agent's bundle. When resources are not divisible, an envy-free allocation is not always possible, and deciding whether one exists is NP-hard.[22] Moreover, one can argue that envy-freeness alone is not sufficient: even if an allocation is envy-free, it is possible that reallocating the tasks or resources can make everyone better off, in which case we say that the original allocation is not Pareto efficient. For example, consider a situation where one agent owns two left shoes, and another agent owns two right shoes. Neither agent envies the other's situation, but both agents can be made better off by trading a left shoe for a right shoe. Pareto efficiency is generally considered to be of paramount importance. There has been work characterizing the computational complexity of finding an allocation that is both envy-free and Pareto efficient.[5]

In a context where every resource is initially owned by one of the agents, it makes sense to use an exchange—even if, for some reason, payments are not possible. The following is one such example of an exchange without payments.

Kidney exchanges. In most exchanges, the participants can make payments to each other, which facilitates trade. However, there are some exchanges in which no payments can be made, so that only items change hands. These are known as barter exchanges. An example is a kidney exchange.[27]

Buying and selling kidneys is illegal in most countries; however, this is not the case for swapping kidneys. As an example, suppose a patient is in need of a kidney transplant, and there is a donor who is willing to give up her kidney for this particular patient, but unfortunately they are not compatible. There may be a second patient-donor pair in the same situation; moreover, it may be the case that the second patient is compatible with the first donor, and the first patient is compatible with the second donor. In this case, it is beneficial for the two patient-donor pairs to swap their donors' kidneys.

It is helpful to think of each patient-donor pair as a single agent, so that each agent has a kidney and needs a(nother) kidney. This makes it easier to see that more complex trades can be beneficial: agent 1's kidney can go to agent 2, agent 2's kidney to agent 3, and agent 3's kidney to agent 1—this is known as a cycle of length 3. Of course, we can also have cycles of length 4, and so on—but it is preferable to not have very long cycles (all the operations in a cycle have to be performed simultaneously so that nobody will back out, which poses a logistical problem for long cycles; also, if last-minute testing discovers an incompatibility in the cycle, the entire cycle collapses).

The key benefit of using an auction is that the resource ends up with the agent who values it the most (or the task ends up with the agent who minds doing it the least). In this case, we say that the auction results in an efficient allocation.

Kidney exchanges are a reality, and computer scientists are involved in them.[1] Indeed, they have started working on the computational problem of clearing the exchange: the input describes which patients are compatible with which of the donors' kidneys, and the output specifies which cycles will be used. Using matching algorithms, the problem can be solved in polynomial time if there are no restrictions on how long cycles can be, or if only cycles of length two are allowed. However, if the maximum cycle length is three or more, then the problem is NP-hard. Nevertheless, in practice, large exchanges can be solved to optimality, using optimization techniques including column generation and branch-and-price search.[1]

Setting with Payments

We now move on to settings where agents can make or receive payments. Payments are useful because they allow us to quantify agents' preferences. Informally, agents now need to put their money where their mouths are. Payments also allow us to transfer happiness (utility) from one agent to another.

Auctions and exchanges. In many problems that require us to decide on an allocation of tasks or resources, it makes sense to also determine payments that some agents should make to other agents. Returning to our example of allocating chores, imagine that the inhabitants are roommates who each pay a share of the rent, and we end up assigning a disproportionate number of chores to one of the roommates. It seems fair that this roommate should pay a smaller share of the rent, which effectively represents a mon-

etary transfer to this roommate from the others. This arrangement may well be to everyone's benefit, for example, if this roommate is unemployed and has plenty of time for completing chores but little money to spend on rent.

Once we start to consider payments in the allocation of tasks and resources, we are quickly drawn into auction theory. (An article on auctions and computer science appeared in the August 2008 issue of *Communications*.[35]) Most people are familiar with the English auction format, where a single item (or a single lot of items) is for sale, and bidders call out increasing bids until nobody is willing to place a higher bid. There are many other auction formats, such as the Dutch auction, where the price is high initially and bidders stay silent as the price gradually decreases, until a bidder announces that she wants to purchase the item at that price, at which point the auction ends immediately.

Yet another format is the sealed-bid format, where bidders write down a bid on a piece of paper, place it in an envelope, and give it to the auctioneer; the auctioneer opens the envelopes and declares the highest bid the winner. Because at this point, we are mostly concerned with how to make a decision based on the agents' preferences, rather than with how these preferences are communicated, it will be easiest for us to think about the sealed-bid format for now.

If we are assigning a task rather than allocating a resource, we can use a reverse auction. Here, a bid of $10 on a task indicates that the bidder wants to be paid $10 for completing the task; in this context, the lowest bid wins. Generally, in an auction, there is a seller who receives the payment from the winning bidder (or, in a reverse auction, a buyer who makes the payment to the winning bidder). A seller is not always present, however: for example, if the agents are trying to decide who gets the right to drive the car on a particular day, they can hold an auction for this right, but in this case it would be natural for the winning agent's payment to go to the losing agents. Some recent work has been devoted to designing mechanisms for redistributing the auction's revenue to the agents.

The key benefit of using an auction (or reverse auction) is that generally, the resource ends up with the agent who values it the most (or the task ends up with the agent who minds doing it the least); in this case, we say that the auction results in an efficient allocation. If an allocation is inefficient, then it is possible to make everyone better off by reallocating some of the tasks/resources, as well as some money. By this argument, efficiency and Pareto efficiency are the same concept in this context.

When there are multiple resources (or tasks) that need to be allocated, one straightforward way of doing this is to hold a separate auction for each resource. However, this approach has a significant downside, which is related to the following observation: how much one of the resources is worth to an agent generally depends on which other resources that agent receives.

For example, if Alice already has the right to drive the car on Friday, then probably having it on Thursday as well is not worth much to her because she can already run her errands on Friday. In contrast, if she does not have the car on any other day, then having it on Thursday is probably very valuable to her. When having one resource makes having another worth less, then we say that the resources are substitutes. On the other hand, Alice may want to go on a two-day trip, in which case having the car on Thursday is worth nothing unless she also has it on Friday. When having one resource makes having another worth more, then we say that the resources are complements.

Substitutability and complementarity make it suboptimal to sell the resources in separate auctions, for the following reason. If the auction for the right to use the car on Thursday is run first, in some sense Alice does not know how much she values it, because she does not yet know whether she will win the auctions for the other days. This uncertainty can result in inefficient allocations.

Combinatorial auctions[12] provide a solution. In a (sealed bid) combinatorial auction, a bidder's bid does not just indicate how much the bidder values each individual item; rather, the bidder expresses a value for every nonempty subset (bundle) of the items. For example, Alice's bid could say: "Having the car on Thursday is worth $5 to me, having it on Friday is worth $6, and having it on both Thursday and Friday is worth $8." Given all this information (for all bidders), an algorithm can search through all possibilities for allocating the items to the bidders, and find the most efficient one—that is, the allocation that maximizes the sum of the agents' valuations.

Similarly, in a combinatorial reverse auction, each bidder expresses how much she wants to be compensated for every bundle of tasks that might be assigned to her. Yet another variant is a combinatorial exchange, in which agents can take the role of a seller as well as the role of a buyer, and they express combinatorial valuations for these more complex trades. These variants face many of the same issues as combinatorial auctions.[12]

Once there are more than a few items in a combinatorial auction, the straightforward approach in which each bidder explicitly states how much every bundle of items is worth to her becomes completely impractical, since there are exponentially many bundles. Instead, we can let bidders use an expressive bidding language that allows them to express natural valuation functions concisely (similarly to the CP-nets that I mentioned in the context of combinatorial voting).

A simple example is the XOR language, in which a bidder explicitly expresses valuations for some (but generally not all) bundles. For example, if the items for sale are $\{a, b, c\}$, a bidder could bid $(\{a\}, 5)$ XOR $(\{b, c\}, 10)$. This indicates that she values the bundle $\{a\}$ at 5; the bundle $\{b, c\}$ at 10; the bundle $\{a, b\}$ at 5, since it is not explicitly listed, but it contains the bundle $\{a\}$; and the bundle $\{a, b, c\}$ at 10, since the highest-value listed bundle that it contains is $\{b, c\}$ (the use of XOR, rather than OR, indicates that we cannot simply add up the values of the two listed bundles to get 15).

The choice of bidding language affects issues such as the computational complexity of the winner determination problem—that is, the problem of finding the efficient allocation of the items, given the bids. Even if each bidder only bids on a single bundle, the combinatorial auction winner determination problem is NP-hard[28] and inapproximable.[29] On the other hand, it is known that under certain conditions on the bids, the winner determination

problem can be solved in polynomial time.[23] For example, if bidders bid only on bundles of at most two items, then the winner determination problem can be solved in polynomial time, via matching algorithms. In general, the runtime heavily depends on how the bids are generated: in some cases, it is possible to scale to hundreds of thousands of items and tens of thousands of bids, whereas in other cases, current techniques have trouble scaling beyond tens of items and hundreds of bids.[30]

Instead of letting bidders bid only once—that is, requiring them to give all their valuation information at once—it is possible to use an iterative (or preference elicitation) format, in which bidders repeatedly respond to queries about their valuations.[25,32] In a single-item setting, this corresponds to the distinction between a sealed-bid auction, in which each bidder bids only once, and an English auction, in which the auctioneer repeatedly queries the bidders for higher valuations. Using preference elicitation in a combinatorial auction has the potential to greatly decrease the total amount of valuation information that the bidders need to communicate, while still finding the efficient allocation. This leads to the following inherently computational question: how should the procedure for querying the bidders be designed to minimize the required amount of communication?

Combinatorial auctions are more than a theoretical curiosity: they are used in practice in settings where the items display significant complementarities. Prominent examples include auctions for radio spectrum, as well as reverse auctions for strategic sourcing (in which large companies set up contracts with suppliers).[12,31,35]

In a context that is perhaps closer to home for most computer scientists, auctions are now also used by the leading search engines to allocate the advertising space on their search results pages. This is another example of an auction with multiple resources for sale: any search performed by a user results in multiple advertisement slots becoming available. These auctions are called *sponsored search auctions*, and they introduce a variety of new issues. For example, in a typical sponsored search auction, an advertiser pays only if the user clicks on her ad, rather than every time her ad is displayed. The prominent place sponsored search auctions occupy in the business models of the companies that use them has helped to bring about an explosion of research on them in recent years;[19] a thorough discussion would easily merit its own article.

While auctions and exchanges are the settings with payments that have attracted the most attention from computer scientists, there are numerous other, more specialized applications. Some of these are discussed here.

Charitable giving. Let us consider a person who is contemplating donating some money (say, $100) to a charitable cause. It may seem that the potential donor should just evaluate what else she would do with the money, and whether that is worth more to her than to see the charity receive $100. While this is a reasonable way to proceed, there are other options if there are multiple potential donors.

Suppose there is a second donor that is making the same decision. Also, let us suppose each donor concludes that she would slightly prefer spending $100 on other things over seeing the charity receive $100. Hence, using the straightforward decision procedure described earlier, neither donor will give any money. However, it may well be that even with these preferences, each donor would prefer the outcome where *both* donors give. That is, each donor may prefer the outcome where the charity receives $200, and she contributes only $100 of this. This is because, other things being equal, they would like the other donor to give as much money to the charity as possible. (Unlike settings discussed earlier in this article, this is inherently a setting with a type of externality: a donor has preferences over what another donor does with her money.) However, with the straightforward decision procedure, neither donor has the ability to influence what the other gives. This is the reason that neither donor gives to the charity.

In fact, there is a way in which a donor can affect another donor's decision. Suppose that one of the two donors can make a binding matching offer, committing to donating the same amount as the other donor. In this case, the other donor has a choice between giving $100, resulting in a $200 total contribution to the charity, and giving nothing, resulting in a $0 total contribution to the charity. Given the preferences that we assumed, the donor will in fact give $100, thereby forcing the other donor

ILLUSTRATION BY JOHN HERSEY

to give $100 as well. It should be noted that both donors (as well as the charity) prefer this outcome to the outcome that results when they make their decisions separately (which is for both of them to give $0).

In practice, a matching offer is generally made by a single large donor, offering to match donations by multiple smaller donors. As we just saw, simple matching offers can lead to improved results, but they are still restrictive. What can be done if multiple donors want to make their donations conditional on the others' donations? This type of expressiveness can lead to even better outcomes, but one has to be careful to avoid circularities. For example, consider the case where *A* will match *B*'s contribution, and *B* will match *A*'s contribution.

We proposed a system in which each donor can make her donation conditional on the total donated to the charity by all the donors combined.[10] In fact, the framework allows for donations to be conditional on the total amounts donated to multiple charities. We also designed algorithms for determining the final outcome based on everyone's offers, which is NP-hard in general but tractable in special cases. We used this system to collect donations for the victims of the Indian Ocean tsunami, and later for the victims of Hurricane Katrina. While the total amount collected from these events was small (about $1,000), the events gave some insight into how donors use the system. About 75% of the donors made their donations conditional on the total amount collected, suggesting that donors appreciated being able to do so. One interesting observation is that the effectiveness of the system (in terms of how much participants were willing to donate) apparently depended on whose donations the donors were matching. The tsunami event was conducted among the participants of a workshop, so that to some extent everyone knew everyone else; in contrast, the hurricane event was open to anyone. The tsunami event was more successful, perhaps because the participants knew whose donations they were matching. More recent systems also allow donors to make their donations conditional only on the donations from selected parties, taking social network structure into account.[14] I believe this innovation has the potential to make such systems much more successful.

Are the agents incentivized to communicate their preferences and beliefs truthfully, or can they benefit from misreporting them?

Prediction markets. The markets I have considered so far generally produce a tangible outcome, such as an allocation of resources. The participating agents have different preferences over the possible outcomes, and the market is a mechanism for finding a good outcome for these preferences. The type of market that I discuss next is a little different.

A prediction market[37] concerns a particular future event whose outcome is currently uncertain. For example, the event could be an upcoming sports game, or an election. The agents trading in the prediction market generally cannot (significantly) influence the outcome of the event; the goal of the market is merely to predict the outcome of the event, based on the collective information and reasoning of the participating agents. Typically, the market prediction is in the form of a probability: for example, the market's assessment may be that the probability that team *A* will beat team *B* is 43%. Prediction markets are quite popular on the Web: examples include the Iowa Electronic Markets as well as Intrade. Each of these runs prediction markets on a variety of events; it appears that the political events (for example, predicting the winner of an election) are the most popular.

A common way to run a prediction market is as follows. We create a security that pays out (say) $1 if team *A* wins, and $0 if team *A* does not win. We then let agents trade these securities. Eventually, this should result in a relatively stable market price: for example, the security may trade at about $0.43. This can be interpreted to mean that the market (that is, the collection of agents) currently believes the probability that team *A* will win is about 43%.

If an agent disagrees with this assessment, then she should buy or sell some of the securities. For example, if an agent believes that the probability is 46% (even after observing the current market price of $0.43), then she can buy one of the securities at price $0.43, and her expected payout for this security will be 46% · $1 = $0.46. As she buys more securities, the market price will eventually go up to $0.46.

If the agent believes the probability is 40%, then she should sell some

of the securities. If she currently does not own any of the securities, she can either short-sell, so that she effectively owns a negative number of the securities; or, she can buy securities for the complementary outcome(s): for example, if the match between *A* and *B* is guaranteed to have a winner, she can buy a security that pays out if *B* wins. The prices of these securities are related: if the match is guaranteed to have a winner, then the sum of the current prices of the security that pays out $1 if *A* wins, and the security that pays out $1 if *B* wins, must always be equal to $1. If it were not, then there would be an opportunity for arbitrage: a combination of deals that leads to a risk-free profit. Specifically, if the sum of the current prices is (say) $0.9, then one can buy both of the securities, and have a guaranteed profit of $0.1, because one of them must pay out. If the sum is (say) $1.1, then one can sell both securities, which again will result in a guaranteed profit of $0.1.

One complication for standard prediction markets is that many real-world events have exponentially many possible outcomes. For example, consider a U.S. presidential election. In a sense, every state (and the District of Columbia) has a separate outcome, so that even with two presidential candidates there are 2^{51} possible outcomes of the election.[b] Of course, we can have a separate market for each of the states, but this will still result in some missed opportunities.

For example, I may believe that with probability 80%, the Democratic candidate will win at least one of Florida, Ohio, and North Carolina. It is not immediately clear how this belief should translate into trading strategies for securities for the individual states. I would much rather simply buy a security that pays out exactly if the Democratic candidate wins at least one of Florida, Ohio, and North Carolina. Now, suppose there is another trader who believes that with probability 30%, the Republican candidate will win all of Florida, Ohio, North Carolina, and Missouri, and would like to buy a security that pays out precisely under these conditions. Ideally, the prediction market could automatically create both of these securities, charge me (say) $0.79 for mine, and charge the other trader (say) $0.29 for hers. Both of us will accept these deals; moreover, since at most one of our two securities will pay out, the prediction market is guaranteed a risk-free profit of at least $0.08. Such combinatorial prediction markets have recently started to receive attention.[6] Running such markets requires solving computationally hard problems: for example, determining whether there is a risk-free combination of securities that can be created is generally NP-hard.

I (speculatively) imagine that in the future, more Web-based mechanisms will be oriented around social networking sites such as Facebook and MySpace. ...The paradigms of computer science give a different and useful perspective on some classic problems in economics.

Strategic Behavior: Game Theory and Mechanism Design

So far, I have focused on allowing agents to communicate their preferences (or, in the case of prediction markets, their beliefs), ideally in an expressive and natural way, as well as on making good decisions based on what was communicated. I have ignored one key aspect, though: Are the agents incentivized to communicate their preferences and beliefs truthfully, or can they benefit from misreporting them?

For example, in an election, an agent's true preferences may be $a \succ b \succ c$. However, if the agent realizes that a has no chance of winning, she may instead choose to vote $b \succ a \succ c$, so as to at least maximize the chances of b winning. Similarly, in an auction, an agent who values the item for sale at $10 may instead bid only $5, in the hope of paying less. While such strategic behavior may be beneficial for the agent who engages in it, it generally makes the quality of the overall outcome worse, because now it is chosen based on input that does not reflect the true preferences.

These considerations lead us into mechanism design. Informally stated, the goal of mechanism design is to design rules for choosing the outcome that lead to good results even in settings where agents are strategic—that is, an agent will lie about her preferences if this is in her best interest. Mechanism design has been studied primarily (until recently, almost exclusively) in

b Actually, this is slightly inaccurate: the states of Maine and Nebraska do not use a winner-takes-all system, further increasing the number of possible outcomes.

economics.[c] Evaluating the quality of a mechanism is nontrivial: it requires being able to predict how multiple strategic agents will act in each other's presence. Game theory provides tools for making such predictions.[d] (An article about computer science and game theory appeared in the August 2008 issue of *Communications*.[34])

The standard approach to mechanism design is simply to ensure it is never beneficial for an agent to lie about her preferences. A result known as the revelation principle suggests that this approach is, from the point of view of strategic behavior, without loss of optimality. A mechanism under which it is never beneficial to lie is called truthful. Unfortunately, it turns out that in general voting settings, no good truthful mechanisms exist, by a result known as the Gibbard-Satterthwaite impossibility theorem.[15,33]

For settings such as auctions and exchanges, where payments can be made, there are much more positive results. For one, if our goal is to allocate the resources efficiently, there are rules for specifying how much agents should pay that make the mechanism as a whole truthful.

A simple example of such a payment rule is the second-price sealed-bid auction for a single item. In this auction, the bidder with the highest bid wins, but only pays the second-highest bid. As a result, the winning bidder's bid does not affect the price she pays; so the only effect that misreporting her valuation for the item can possibly have is that she does not win, which would make her worse off. Similarly, the only effect that misreporting can possibly have for a losing bidder is that she ends up winning at a price that is too high for her, which would make her worse off. So, a bidder is always best off reporting her true valuation for the item—that is, the second-price sealed-bid auction is truthful. This scheme can be generalized to combinatorial auctions and exchanges (and other settings), resulting in the class of Vickrey-Clarke-Groves (VCG) mechanisms.[7,16,36]

c In 2007, Hurwicz, Maskin, and Myerson received the Nobel Prize in Economics for their fundamental work on mechanism design.

d Game theory has led to two other Nobel Prizes in Economics: Nash, Selten, and Harsanyi received one in 1994, and Aumann and Schelling in 2005.

The issues studied in mechanism design interact with the computational issues I discussed before in subtle ways. For example, suppose we want to run a combinatorial auction using a VCG mechanism. Technically, this means we should always solve the winner determination problem to optimality, that is, find the most efficient allocation—which we know is NP-hard. If we do not always succeed at finding the most efficient allocation, then the resulting mechanism will, in general, not be truthful. A significant amount of research has addressed the problem of designing polynomial-time approximation algorithms that, in combination with the right payment rule, are truthful.[21] More generally, the problem of designing efficient algorithms that can be made truthful is the main topic of algorithmic mechanism design.[24] This line of research has also been extended to distributed settings without a trusted center.[13]

We can use computers not only to run existing mechanisms, but also to design new mechanisms from scratch. That is, for a given setting, we let an algorithm search through the space of all possible truthful mechanisms for an optimal one.[9] This approach is called automated mechanism design. Finding an optimal mechanism is computationally much harder than running an existing mechanism, and as a result automated mechanism design has so far been successful only on small instances. Nevertheless, some real instances are in fact small, and even for larger instances, solving a simplified version can give some helpful intuition. Automated mechanism design can also be used to solve some small instances of a general mechanism design problem; then, a human mechanism designer can try to identify a pattern in these small solutions, conjecture the general solution, and prove it analytically. In this way, automated mechanism design can contribute to microeconomic theory. This methodology has recently been used to design mechanisms for redistributing an auction's revenue to the bidders in a truthful way (for example, Guo and Conitzer[17]), and the methodology is starting to be adopted more widely.

It is not always the mechanism designer or the party running the mechanism that faces hard computational problems. Under some mechanisms, it is computationally hard for the agents to find the strategically optimal action to take. This is not the case for truthful mechanisms, where strategically optimal behavior simply means telling the truth. However, no reasonable voting rule is truthful in sufficiently general settings (by the Gibbard-Satterthwaite theorem mentioned above). It has been shown that in a variety of voting settings, it is NP-hard to find the strategically optimal vote(s) to cast, even if the other agents' votes are already known (for example, Bartholdi,[2] Conitzer,[11] and Hemaspaandra[18]). This is a case where computational hardness can be desirable: it can be argued that if a voter cannot find a way of misreporting her preferences that benefits her, then she will presumably tell the truth. For now, the impact of this type of result is limited by the fact that NP-hardness is a worst-case measure, and it may well be the case that it is easy to find an effective way of misreporting one's preferences *most of the time*.

Another important issue is that the mechanisms from traditional mechanism design mainly guard against a single type of manipulation: misreporting one's preferences. However, mechanisms run in highly anonymous environments such as the Internet are vulnerable to other types of manipulation. Specifically, it is often possible for a single agent to pretend to be multiple agents (known as false-name manipulation or a Sybil attack). The standard mechanisms for guarding against misreporting, such as the VCG mechanisms, are generally not robust to false-name manipulation. A mechanism that is robust to it—that is, under which no agent ever benefits from using multiple identifiers—is said to be false-name-proof,[39] and a growing body of research attempts to design such mechanisms.

A final direction in mechanism design concerns extending its techniques to dynamic environments, where decisions must be made over time as additional information enters the system. Recent years have seen rapid progress in generalizing mechanism design techniques from static to dynamic settings.[26] For example, sponsored search auctions are, in principle, a good application domain for such techniques: the demand for, as well as the supply of, advertise-

ment slots next to the results for specific searches changes over time, but allocation decisions must be made now.

Conclusion

In this article, I have considered a number of settings in which a decision needs to be made based on the preferences of multiple agents, as well as mechanisms for reaching the decision. People have been using such mechanisms for millennia, and have studied them formally for centuries (although their game-theoretic analysis has taken place mostly in the last 50 years). Still, computer scientists are fundamentally changing these mechanisms and how they are being used.

Increased computing power and better algorithms enable the use of mechanisms, such as the Kemeny voting rule and combinatorial auctions, that were once considered impractical. Also, the Internet provides a great platform for these mechanisms: it makes it easy for spatially distributed users to communicate their preferences to the mechanism, and they will generally be forced to communicate them in a precise way (for example, a bidder will have to enter a number on a Web site rather than vaguely communicating her preferences over the phone), which makes it possible to run the mechanism automatically. I (speculatively) imagine that in the future, more Web-based mechanisms will be oriented around social networking sites such as Facebook and MySpace; the charitable donations work[14] is a good example of how such social network structure can be used. Computer scientists are also encountering mechanism design problems in their own work, for example, when shared computing resources need to be allocated to users. Finally, the paradigms of computer science give a different and useful perspective on some classic problems in economics.

This article has summarized a number of applications where computer scientists have already become involved in the design of markets and other protocols for making decisions based on the preferences of multiple agents. I anticipate that the number and importance of such applications will grow steeply in the years to come. One major reason for this is that computer scientists and economists interested in market design have grown closer together in recent years, and are now seen working together more often (this is necessitated by high-value applications such as sponsored search auctions). Computer scientists have caught up on many of the key techniques developed in the microeconomics theory literature. On the other side, economists are becoming increasingly familiar with techniques from modern computer science. This is a very nice example where "computational thinking" is being exported to another discipline (which is certainly not to say that there were no prior instances of economists thinking computationally).

Acknowledgments

This work is supported by NSF award number IIS-0812113, a Research Fellowship from the Alfred P. Sloan Foundation, and a Yahoo! Faculty Research Grant. I thank the reviewers for very detailed and helpful feedback. I also thank Tuomas Sandholm for feedback on the kidney exchange section, and David Pennock for feedback on the prediction markets section. All errors and omissions are my own (though of course I faced constraints on length and number of citations).

References

1. Abraham, D., Blum, A., and Sandholm, T. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the ACM Conference on Electronic Commerce* (San Diego, CA, 2007), 295–304.
2. Bartholdi III, J., Tovey, C., and Trick, M. The computational difficulty of manipulating an election. *Social Choice and Welfare 6*, 3 (1989), 227–241.
3. Bartholdi III, J., Tovey, C., and Trick, M. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare 6* (1989), 157–1659.
4. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., and Poole, D. CP-nets: A tool for representing and reasoning with conditional ceteris paribus statements. *J. Artificial Intelligence Research 21* (2004), 135–191.
5. Bouveret, S. and Lang, J. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (Edinburgh, Scotland, 2005), 935–940.
6. Chen, Y., Fortnow, L., Nikolova, E., and Pennock, D.M. Combinatorial betting. *ACM SIGecom Exchanges 7*, 1 (2007), 61–64.
7. Clarke, E.H. Multipart pricing of public goods. *Public Choice 11* (1971), 17–33.
8. Conitzer, V., Davenport, A., and Kalagnanam, J. Improved bounds for computing Kemeny rankings. In *Proceedings of the National Conference on Artificial Intelligence* (Boston, MA, 2006), 620–626).
9. Conitzer, V. and Sandholm, T. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence* (Edmonton, Canada, 2002), 103–110.
10. Conitzer, V. and Sandholm, T. Expressive negotiation over donations to charities. In *Proceedings of the ACM Conference on Electronic Commerce. ACM*, NY (2004), 51–60.
11. Conitzer, V., Sandholm, T., and Lang, J. When are elections with few candidates hard to manipulate? *J. ACM 54*, 3 (2007), 1–33.
12. Cramton, P., Shoham, Y., and Steinberg, R. *Combinatorial Auctions.* MIT Press, Cambridge, MA, 2006.
13. Feigenbaum, J., Schapira, M., and Shenker, S. Distributed algorithmic mechanism design. *Algorithmic Game Theory.* N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, Eds. Cambridge University Press, 2007.
14. Ghosh, A. and Mahdian, M. Charity auctions on social networks. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* (2008), 1019–1028.
15. A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica 41* (1973), 587–602.
16. Groves, T. Incentives in teams. *Econometrica 41* (1973), 617–631.
17. Guo, M. and Conitzer, V. Worst-case optimal redistribution of VCG payments in multi-unit auctions. *Games and Economic Behavior*, 2009. To appear.
18. Hemaspaandra, E. and Hemaspaandra, L.A. Dichotomy for voting systems. *J. Comput. Syst. Sci. 73*, 1 (2007), 73–83.
19. Lahaie, S., Pennock, D.M., Saberi, A., and Vohra, R.V. Sponsored search auctions. *Algorithmic Game Theory.* N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, Eds. Cambridge University Press, 2007.
20. Lang, J. Vote and aggregation in combinatorial domains with structured preferences. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (Hyderabad, India, 2007), 1366–1371.
21. Lehmann, D., O'Callaghan, L.I., and Shoham, Y. Truth revelation in rapid, approximately efficient combinatorial auctions. *J. ACM 49*, 5 (2002), 577–602.
22. Lipton, R., Markakis, E., Mossel, E., and Saberi, A. On approximately fair allocations of indivisible goods. In *Proceedings of the ACM Conference on Electronic Commerce.* ACM, NY, (2004), 125–131.
23. Müller, R. Tractable cases of the winner determination problem. *Combinatorial Auctions.* P. Cramton, Y. Shoham, and R. Steinberg, Eds. MIT Press, Cambridge, MA, 2006, 319–336.
24. Nisan, N and Ronen, A. Algorithmic mechanism design. *Games and Economic Behavior 35* (2001), 166–196.
25. Parkes, D. Iterative combinatorial auctions. *Combinatorial Auctions.* P. Cramton, Y. Shoham, and R. Steinberg, Eds. MIT Press, 2006, 41–77.
26. Parkes, D. Online mechanisms. *Algorithmic Game Theory.* N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, Eds. Cambridge University Press, 2007.
27. Roth, A.E., Sonmez, T., and Unver, M.U. Kidney exchange. *Quarterly J. Economics, 119*, (2004), 457–488.
28. Rothkopf, M., Pekec, A., and Harstad, R. Computationally manageable combinatorial auctions. *Management Science 44*, 8 (1998), 1131–1147.
29. Sandholm, T. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence 135* (Jan. 2002), 1–54.
30. Sandholm, T. Optimal winner determination algorithms. *Combinatorial Auctions.* P. Cramton, Y. Shoham, and R. Steinberg, Eds. MIT Press, 2006, 337–368.
31. Sandholm, T. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine 28*, 3 (2007), 45–58.
32. Sandholm, T. and Boutilier, C. Preference elicitation in combinatorial auctions. *Combinatorial Auctions.* P. Cramton, Y. Shoham, and R. Steinberg, Eds. MIT Press, 2006, 233–263.
33. Satterthwaite, M. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *J. Economic Theory 10* (1975), 187–217.
34. Shoham, Y. Computer science and game theory. *Commun. ACM 51*, 8 (Aug. 2008), 74–79.
35. Varian, H.R. Designing the perfect auction. *Commun. ACM 51*, 8 (Aug. 2008) 9–11.
36. Vickrey, W. Counterspeculation, auctions, and competitive sealed tenders. *J. Finance 16* (1961), 8–37.
37. Wolfers, J. and Zitzewitz, E. Prediction markets. *J. Economic Perspectives 18*, 2 (2004), 107–126.
38. Xia, L., Conitzer, V., and Lang, J. Voting on multiattribute domains with cyclic preferential dependencies. In *Proceedings of the National Conference on Artificial Intelligence* (Chicago, IL, 2008), 202–207.
39. Yokoo, M., Sakurai, Y., and Matsubara, S. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior 46*, 1 (2004), 174–188.
40. Young, H.P. Optimal voting rules. *J. Economic Perspectives 9*, 1 (1995), 51–64.

Vincent Conitzer (conitzer@cs.duke.edu) is an assistant professor of computer science and economics at Duke University, Durham, NC.

research highlights

DOI:10.1145/1666420.1666445

Technical Perspective
A First Glimpse of Cryptography's Holy Grail

By Daniele Micciancio

WE ALL KNOW how to protect our private or most valuable data from unauthorized access: encrypt it. When a piece of data M is encrypted under a key K to yield a ciphertext $C=Enc_K(M)$, only the intended recipient (who knows the corresponding secret decryption key S) will be able to invert the encryption function and recover the original plaintext using the decryption algorithm $Dec_S(C)=Dec_S(Enc_K(M))=M$.

Encryption today—in both symmetric (where S=K) and public key versions (where S remains secret even when K is made publicly available)—is widely used to achieve confidentiality in many important and well-known applications: online banking, electronic shopping, and virtual private networks are just a few of the most common applications using encryption, typically as part of a larger protocol, like the TLS protocol used to secure communication over the Internet.

Still, the use of encryption to protect valuable or sensitive data can be very limiting and inflexible. Once the data M is encrypted, the corresponding ciphertext *C* behaves to a large extent as a black box: all we can do with the box is keep it closed or opened in order to access and operate on the data.

In many situations this may be exactly what we want. For example, take a remote storage system, where we want to store a large collection of documents or data files. We store the data in encrypted form, and when we want to access a specific piece of data, we retrieve the corresponding ciphertext, decrypting it locally on our own trusted computer. But as soon as we go beyond the simple data storage/retrieval model, we are in trouble. Say we want the remote system to provide a more complex functionality, like a database system capable of indexing and searching our data, or answering complex relational or semistructured queries. Using standard encryption technology we are immediately faced with a dilemma: either we store our data unencrypted and reveal our precious or sensitive data to the storage/database service provider, or we encrypt it and make it impossible for the provider to operate on it.

If data is encrypted, then answering even a simple counting query (for example, the number of records or files that contain a certain keyword) would typically require downloading and decrypting the entire database content.

Homomorphic encryption is a special kind of encryption that allows operating on ciphertexts without decrypting them; in fact, without even knowing the decryption key. For example, given ciphertexts $C=Enc_K(M)$ and $C'=Enc_K(M')$, an additively homomorphic encryption scheme would allow to combine C and C' to obtain $Enc_K(M+M')$. Such encryption schemes are immensely useful in the design of complex cryptographic protocols. For example, an electronic voting scheme may collect encrypted votes $C_i=Enc_K(M_i)$ where each vote M_i is either 0 or 1, and then tally them to obtain the encryption of the outcome $C=Enc_K(M_1+..+M_n)$. This would be decrypted by an appropriate authority that has the decryption key and ability to announce the result, but the entire collection and tallying process would operate on encrypted data without the use of the secret key. (Of course, this is an oversimplified protocol, as many other issues must be addressed in a real election scheme, but it well illustrates the potential usefulness of homomorphic encryption.)

To date, all known homomorphic encryption schemes supported essentially only one basic operation, for example, addition. But the potential of fully homomorphic encryption (that is, homomorphic encryption supporting arbitrarily complex computations on ciphertexts) is clear. Think of encrypting your queries before you send them to your favorite search engine, and receive the encryption of the result without the search engine even knowing what the query was. Imagine running your most computationally intensive programs on your large datasets on a cluster of remote computers, as in a cloud computing environment, while keeping both your programs, data, and results encrypted and confidential. The idea of fully homomorphic encryption schemes was first proposed by Rivest, Adleman, and Dertouzos the late 1970s, but remained a mirage for three decades, the never-to-be-found Holy Grail of cryptography. At least until 2008, when Craig Gentry announced a new approach to the construction of fully homomorphic cryptosystems.

In the following paper, Gentry describes his innovative method for constructing fully homomorphic encryption schemes, the first credible solution to this long-standing major problem in cryptography and theoretical computer science at large. While much work is still to be done before fully homomorphic encryption can be used in practice, Gentry's work is clearly a landmark achievement. Before Gentry's discovery many members of the cryptography research community thought fully homomorphic encryption was impossible to achieve. Now, most cryptographers (me among them) are convinced the Holy Grail exists. In fact, there must be several of them, more or less efficient ones, all out there waiting to be discovered.

Gentry gives a very accessible and enjoyable description of his general method to achieve fully homomorphic encryption as well as a possible instantiation of his framework recently proposed by van Dijik, Gentry, Halevi, and Vaikuntanathan. He has taken great care to explain his technically complex results, some of which have their roots in lattice-based cryptography, using a metaphorical tale of a jeweler and her quest to keep her precious materials safe, while at the same time allowing her employees to work on them.

Gentry's homomorphic encryption work is truly worth a read. C

Daniele Micciancio is a professor in the computer science and engineering department at the University of California, San Diego.

DOI:10.1145/1666420.1666444

Computing Arbitrary Functions of Encrypted Data

By Craig Gentry

Abstract

Suppose that you want to delegate the ability to *process* your data, without giving away *access* to it. We show that this separation is possible: we describe a "fully homomorphic" encryption scheme that keeps data private, but that allows a worker that *does not have the secret decryption key* to compute any (still encrypted) result of the data, even when the function of the data is very complex. In short, a third party can perform complicated processing of data without being able to see it. Among other things, this helps make cloud computing compatible with privacy.

1. INTRODUCTION

Is it possible to delegate *processing* of your data without giving away *access* to it?

This question, which tests the tension between convenience and privacy, has always been important, but seems especially so now that we are headed toward widespread use of cloud computing. To put everything online "in the cloud," unencrypted, is to risk an Orwellian future. For certain types of data, such as medical records, storing them off-site unencrypted may be illegal. On the other hand, encrypting one's data seems to nullify the benefits of cloud computing. Unless I give the cloud my secret decryption key (sacrificing my privacy), what can I expect the cloud to do with my encrypted data except send it back to me, so that I can decrypt it and process it myself?

Fortunately, this is a false dilemma, or at least convenience and privacy can be reconciled to a large extent. For data that is encrypted with an "ordinary" encryption scheme, it is virtually impossible for someone without the secret decryption key (such as the cloud) to manipulate the underlying data in any useful way. However, some encryption schemes are *homomorphic* or *malleable*. They let anyone manipulate (in a meaningful way) what is encrypted, even without knowing the secret key!

In this paper, we describe the first *fully homomorphic* encryption (FHE) scheme, where "fully" means that there are no limitations on what manipulations can be performed. Given ciphertexts $c_1, ..., c_t$ that encrypt $m_1, ..., m_t$ with our scheme under some key, and given any efficiently computable function f, anyone can efficiently compute a ciphertext (or set of ciphertexts) that encrypts $f(m_1, ..., m_t)$ under that key. In short, this permits general computations on encrypted data. No information about $m_1, ..., m_t$ or the value of $f(m_1, ..., m_t)$ is leaked.

This means that cloud computing is consistent with privacy. If I want the cloud to compute for me some function f of my (encrypted) data $m_1, ..., m_t$—for example, this function could be "all files containing 'CACM' or 'Communications' within three words of 'ACM' "—I send a description of f to the cloud, which uses the scheme's malleability to compute an encryption of $f(m_1, ..., m_t)$, which I decrypt. The cloud never sees any unencrypted data. If I want, I can even use the scheme to encrypt a description of f, so that the cloud does not even see what I am searching for.

Rivest, Adleman, and Dertouzos[5] suggested that fully homomorphic encryption may be possible in 1978, shortly after the invention of the RSA cryptosystem,[6] but were unable to find a secure scheme. As an application, they described our private cloud computing scenario above, though of course they used different terminology. There are many other applications. Homomorphic encryption is useful whenever it is acceptable if a response (e.g., to a search engine query) is encrypted.

Below, we begin by describing homomorphic encryption in more detail. Then, we describe a concrete scheme due to van Dijk, Gentry, Halevi, and Vaikuntanathan,[9] which uses only simple integer operations, and is a conceptually simpler version of the first scheme by Gentry,[2, 3] which uses lattices. Toward the end, we discuss the scheme's (rather slow) performance. Throughout, we try to make the ideas more tangible by constantly returning to a physical analogy: a jewelry store owner, Alice, who wants her workers to *process* raw precious materials into intricately designed rings and necklaces, but who is afraid to give her workers complete *access* to the materials for fear of theft.

2. HOMOMORPHIC ENCRYPTION

2.1. Alice's jewelry store

At first, the notion of processing data without having access to it may seem paradoxical, even logically impossible. To convince you that there is no fallacy, and to give you some intuition about the solution, let us consider an analogous problem in (a fictional version of) the "physical world."

Alice owns a jewelry store. She has raw precious materials—gold, diamonds, silver, etc.—that she wants her workers to assemble into intricately designed rings and

This paper draws from the STOC 2009 paper "Fully Homomorphic Encryption Using Ideal Lattices," my thesis, and a recent manuscript co-authored with van Dijk, Halevi, and Vaikuntanathan.

necklaces. But she distrusts her workers and assumes that they will steal her jewels if given the opportunity. In other words, she wants her workers to *process* the materials into finished pieces, without giving them *access* to the materials. What does she do?

Here is her plan. She uses a transparent impenetrable glovebox, secured by a lock for which only she has the key. She puts the raw precious materials inside the box, locks it, and gives it to a worker. Using the gloves, the worker assembles the ring or necklace inside the box. Since the box is impenetrable, the worker cannot get to the precious materials, and figures he might as well return the box to Alice, with the finished piece inside. Alice unlocks the box with her key and extracts the ring or necklace. In short, the worker processes the raw materials into a finished piece, without having true access to the materials.

The locked impenetrable box, with raw precious materials inside, represents an encryption of the initial data $m_1, ..., m_t$, which can be accessed only with the secret decryption key. The gloves represent the homomorphism or malleability of the encryption scheme, which allows the raw data to be manipulated while it is inside the "encryption box." The completed ring or necklace inside the box represents the encryption of $f(m_1, ..., m_t)$, the desired function of the initial data. Note that "lack of access" is represented by lack of physical access, as opposed to lack of visual access, to the jewels. (For an analogy that uses lack of visual access, consider a photograph developer's darkroom.)

Of course, Alice's jewelry store is only an analogy. It does not represent some aspects of homomorphic encryption well, and taking it too literally may be more confusing than helpful. We discuss some flaws in the analogy at the end of this section, after we describe homomorphic encryption more formally. Despite its flaws, we return to the analogy throughout, since it motivates good questions, and represents some aspects of our solution quite well—most notably, "bootstrapping," which we discuss in Section 4.

2.2. Homomorphic encryption: functionality

An encryption scheme ε has three algorithms: $\mathsf{KeyGen}_\varepsilon$, $\mathsf{Encrypt}_\varepsilon$, and $\mathsf{Decrypt}_\varepsilon$, all of which must be *efficient*—that is, run in time poly(λ), polynomial in a security parameter λ that specifies the bit-length of the keys. In a *symmetric*, or *secret key*, encryption scheme, $\mathsf{KeyGen}_\varepsilon$ uses λ to generate a single key that is used in both $\mathsf{Encrypt}_\varepsilon$ and $\mathsf{Decrypt}_\varepsilon$, first to map a message to a ciphertext, and then to map the ciphertext back to the message. In an *asymmetric*, or *public key*, encryption scheme, $\mathsf{KeyGen}_\varepsilon$ uses λ to generate two keys—a public encryption key pk, which may be made available to everyone, and a secret decryption key sk. As a physical analogy for an asymmetric encryption scheme, one can think of Alice's public key as a padlock, which she constructs and distributes, that can be locked without a key. Anyone can put a message inside a box secured by Alice's padlock (encrypt), and mail it via a public channel to Alice, but only Alice has the key needed to unlock it (decrypt).

A homomorphic encryption scheme can be either symmetric or asymmetric, but we will focus on the asymmetric case. It has a fourth algorithm $\mathsf{Evaluate}_\varepsilon$, which is associated to a set $\mathcal{F}_\varepsilon$ of *permitted functions*. For any function f in $\mathcal{F}_\varepsilon$ and any ciphertexts $c_1, ..., c_t$ with $c_i \leftarrow \mathsf{Encrypt}_\varepsilon(\mathrm{pk}, m_i)$, the algorithm $\mathsf{Evaluate}_\varepsilon(\mathrm{pk}, f, c_1, ..., c_t)$ outputs a ciphertext c that encrypts $f(m_1, ..., m_t)$—i.e., such that $\mathsf{Decrypt}_\varepsilon(\mathrm{sk}, c) = f(m_1, ..., m_t)$. (For convenience, we will assume that f has one output. If f has k outputs, then $\mathsf{Evaluate}_\varepsilon$ outputs k ciphertexts that encrypt $f(m_1, ..., m_t)$ collectively.) As shorthand, we say that ε can *handle* functions in $\mathcal{F}_\varepsilon$. For a function f not in $\mathcal{F}_\varepsilon$, there is no guarantee that $\mathsf{Evaluate}_\varepsilon$ will output anything meaningful. Typically $\mathsf{Evaluate}_\varepsilon$ is undefined for such a function.

As described thus far, it is trivial to construct an encryption scheme that can handle all functions. Just define $\mathsf{Evaluate}_\varepsilon$ as follows: simply output $c \leftarrow (f, c_1, ..., c_t)$, without "processing" the ciphertexts at all. Modify $\mathsf{Decrypt}_\varepsilon$ slightly: to decrypt c, decrypt $c_1, ..., c_t$ to obtain $m_1, ..., m_t$, and then apply f to these messages.

But this trivial solution obviously does not conform to the spirit of what we are trying to achieve—to *delegate* the data processing (while maintaining privacy). The trivial solution is as if, in Alice's jewelry store, the worker simply sends the box (which need not have gloves) back to Alice without doing any work on the raw precious materials, and Alice unlocks the box, extracts the materials, and assembles the ring or necklace herself.

So, how do we formalize what it means to *delegate*? Intuitively, the purpose of delegation is to reduce one's workload. We can formalize this in terms of the running times (i.e., complexity) of the algorithms. Specifically, we require that decrypting c (the ciphertext output by $\mathsf{Evaluate}_\varepsilon$) takes the *same amount of computation* as decrypting c_1 (a ciphertext output by $\mathsf{Encrypt}_\varepsilon$). Moreover, we require that c is the same size as c_1. We refer to these as the *compact ciphertexts* requirement. Again, the size of c and the time needed to decrypt it do not grow with the complexity of f; rather, they are *completely independent* of f (unless f has multiple outputs). Also, of course, the complexity of $\mathsf{Decrypt}_\varepsilon$, as well as the complexity of $\mathsf{KeyGen}_\varepsilon$ and $\mathsf{Encrypt}_\varepsilon$, must remain polynomial in λ.

ε is *fully homomorphic* if it can handle all functions, has compact ciphertexts, and $\mathsf{Evaluate}_\varepsilon$ is efficient in a way that we specify below. The trivial solution above certainly is not fully homomorphic, since the size of the ciphertext output by $\mathsf{Evaluate}_\varepsilon$, as well as the time needed to decrypt it, depend on the function being evaluated. In terms of Alice's jewelry store, our definition of fully homomorphic captures the best-case scenario for Alice: her workers can assemble arbitrarily complicated pieces inside the box, but the work needed to assemble has no bearing on the work Alice needs to do to unlock the box and extract the piece.

We want our fully homomorphic scheme to be efficient for the worker, as well. In particular, we want the complexity of $\mathsf{Evaluate}_\varepsilon$—like the other algorithms of ε—to depend only polynomially on the security parameter. But clearly its complexity must also depend on the function being evaluated. How do we measure the complexity of f? Perhaps the most obvious measure is the running time T_f of a Turing machine that computes f. We use a related measure, the size

S_f of a *boolean circuit* (i.e., the number of AND, OR, and NOT gates) that computes f. Any function that can be computed in T_f steps on a Turing machine can be expressed as a circuit with about T_f gates. More precisely, $S_f < k \cdot T_f \cdot \log T_f$ for some small constant k. Overall, we say that $\mathsf{Evaluate}_\varepsilon$ is efficient if there is a polynomial g such that, for any function f that is represented by a circuit of size S_f, $\mathsf{Evaluate}_\varepsilon(\mathrm{pk}, f, c_1, \ldots, c_t)$ has complexity at most $S_f \cdot g(\lambda)$.

The circuit representation of f is also useful because it breaks the computation of f down into simple steps—e.g., AND, OR, and NOT gates. Moreover, to evaluate these gates, it is enough to be able to add, subtract, and multiply. (In fact, it is enough if we can add, subtract and multiply *modulo 2*.) In particular, for $x, y \in \{0, 1\}$, we have $\mathsf{AND}(x, y) = xy$, $\mathsf{OR}(x, y) = 1 - (1 - x)(1 - y)$ and $\mathsf{NOT}(x) = 1 - x$. So, to obtain a fully homomorphic encryption scheme, all we need is a scheme that operates on ciphertexts so as to add, subtract, and multiply the underlying messages, indefinitely.

But is the circuit representation of f—or some arithmetized version of it in terms of addition, subtraction, and multiplication—necessarily the *most efficient* way to evaluate f? In fact, some functions, like binary search, take much longer on a Turing machine or circuit than on a random access machine. On a random access machine, a binary search algorithm on t ordered items only needs to "touch" $O(\log t)$ of its inputs.

A moment's thought shows that random-access speedups cannot work if the data is encrypted. Unless we know something a priori about the relationship between f and $m_1, \ldots, m_t$, the algorithm $\mathsf{Evaluate}_\varepsilon(\mathrm{pk}, f, c_1, \ldots, c_t)$ must touch all of the input ciphertexts, and therefore have complexity at least linear in the number of inputs. To put it another way, if $\mathsf{Evaluate}_\varepsilon$ (for some reason) did not touch the second half of the ciphertexts, this would leak information about the second half of the underlying messages—namely, their irrelevance in the computation of f—and this leakage would contradict the security of the encryption scheme. While $\mathsf{Evaluate}_\varepsilon$ must have running time at least linear in t as an unavoidable cost of the complete privacy that homomorphic encryption provides, a trade-off is possible. If I am willing to reveal—e.g., in the cloud computing context—that the files that I want are contained in a certain 1% of my data, then I may help the cloud reduce its work by a factor of 100.

Another artifact of using a fixed circuit representation of f is that the size of the output—i.e., the number of output wires in the circuit—must be fixed in advance. For example, when I request all of my files that contain a combination of keywords, I should also specify how much data I want retrieved—e.g., 1MB. From my request, the cloud will generate a circuit for a function that outputs the first megabyte of the correct files, where that output is truncated (if too much of my data satisfies my request), or padded with zeros (if too little). A moment's thought shows that this is also unavoidable. There is no way the cloud can avoid truncating or padding unless it knows something a priori about the relationship between the function and my data.

2.3. Homomorphic encryption: security

In terms of security, the weakest requirement for an encryption scheme is *one-wayness*: given the public key pk and a ciphertext c that encrypts unknown message m under pk, it should be "hard" to output m. "Hard" means that any algorithm or "adversary" $\mathcal{A}$ that runs in poly(λ) time has a negligible probability of success over the choices of pk and m (i.e., the probability it outputs m is less than $1/\lambda^k$ for any constant k).

Nowadays, we typically require an encryption scheme to have a stronger security property, called *semantic security* against chosen-plaintext attacks (CPA)[4]: given a ciphertext c that encrypts either m_0 or m_1, it is hard for an adversary to decide which, even if it is allowed to choose m_0 and m_1. Here, "hard" means that if the adversary $\mathcal{A}$ runs in polynomial time and guesses correctly with probability $1/2 + \epsilon$, then ϵ, called $\mathcal{A}$'s *advantage*, must be negligible. If this advantage is nonnegligible, then we say (informally) that the adversary *breaks* the semantic security of the encryption scheme.

If an encryption scheme is *deterministic*—i.e., if there is only one ciphertext that encrypts a given message—then it cannot be semantically secure. An attacker can easily tell whether c encrypts m_0, by running $c_0 \leftarrow \mathsf{Encrypt}(\mathrm{pk}, m_0)$ and seeing if c and c_0 are the same. A semantically secure encryption scheme must be *probabilistic*—i.e., there must be many ciphertexts that encrypt a given message, and $\mathsf{Encrypt}_\varepsilon$ must choose one randomly according to some distribution.

One can *prove* the (conditional) one-wayness or semantic security of an encryption scheme by *reducing* a hard problem to breaking the encryption scheme. For example, suppose one shows that if there is an efficient algorithm that breaks the encryption scheme, then this algorithm can be used as a subroutine in an efficient algorithm that factors large numbers. Then, under the assumption that factoring is hard—i.e., that no poly(λ)-time algorithm can factor λ-bit numbers—the reduction implies that the encryption scheme must be hard to break.

Semantic security of a homomorphic encryption scheme is defined in the same way as for an ordinary encryption scheme, without reference to the $\mathsf{Evaluate}_\varepsilon$ algorithm. If we manage to prove a reduction—i.e., that an attacker that breaks ε can be used to solve a hard problem like factoring—then this reduction holds whether or not ε has an $\mathsf{Evaluate}_\varepsilon$ algorithm that works for a large set of functions.

To understand the power of semantic security, let us reconsider our cloud computing application. Sometime after storing her encrypted files in the cloud, Alice wants the cloud to retrieve the files that have a certain combination of keywords. Suppose that in its response, the cloud sends ciphertexts that encrypt the first three files. Can't the cloud just *see* that the first three encrypted files that it is storing for Alice happen to encrypt the same content as the three files that it sends to Alice? Not if the scheme is semantically secure. Even though some of the stored ciphertexts encrypt the same content as the sent ciphertexts, the cloud cannot *see* this, because semantic security guarantees that it is hard to tell whether two ciphertexts encrypt the same content.

Intuitively, it seems like the $\mathsf{Evaluate}_\varepsilon$ algorithm should make ε easier to break, simply because this additional algorithm gives the attacker more power. Or, to put it in terms of the physical analogy, one would think that the easiest way to get inside the glovebox is to cut through the gloves, and that, the more flexible the gloves are, the easier the glovebox

is to compromise; this suggests that, the more malleable the encryption scheme is, the easier it is to break. There is some truth to this intuition. Researchers[1, 8] showed that if ε is a *deterministic* fully homomorphic encryption scheme (or, more broadly, one for which it is easy to tell whether two ciphertexts encrypt the same thing), then ε can be broken in subexponential time, and in only polynomial time (i.e., efficiently) on a quantum computer. So, malleability seems to weaken the security of deterministic schemes. But these results do not apply to semantically secure schemes, such as ours.

2.4. Some flaws in the physical analogy

The physical analogy represents some aspects of homomorphic encryption poorly. For example, the physical analogy suggests that messages that are encrypted separately are in different "encryption boxes" and cannot interact. Of course, this interaction is precisely the purpose of homomorphic encryption. To fix the analogy, one may imagine that the gloveboxes have a one-way insertion slot like the mail bins used by the post office. Then, messages can be added to the same encryption box as they arrive. (Even this fix is not entirely satisfactory.)

Another flaw is that the output $f(m_1, ..., m_t)$ may have significantly fewer bits than $m_1, ..., m_t$, whereas in the analogy (absent significant nuclear activity inside the glovebox) the conservation of mass dictates that the box will have at least as much material inside when the worker is done as when he started. Finally, in Alice's jewelry store, even though a worker cannot extract the materials from a locked glovebox, he can easily tell whether or not a box contains a certain set of materials—i.e., the gloveboxes do not provide "semantic security."

3. A SOMEWHAT HOMOMORPHIC ENCRYPTION SCHEME

On our way to fully homomorphic encryption, we begin by constructing a *somewhat homomorphic* encryption scheme ε that can handle a limited class $\mathcal{F}_\varepsilon$ of permitted functions. $\mathsf{Evaluate}_\varepsilon(\mathrm{pk}, f, c_1, ..., c_t)$ does not work for functions f that are too complicated. Later, we will show to use ε to obtain fully homomorphic encryption.

3.1. Meanwhile in Alice's jewelry store

After figuring out how to use locked gloveboxes to get her workers to process her precious materials into fancy rings and necklaces, Alice puts in an order with Acme Glovebox Company. Unfortunately, the gloveboxes she receives are defective. After a worker uses the gloves for 1 min, the gloves stiffen and become unusable. But some of the fanciest pieces take up to an hour to assemble. Alice sues Acme, but meanwhile she wonders: Is there some way I can use these defective boxes to get the workers to securely assemble even the most complicated pieces?

She notices that the boxes, while defective, do have a property that might be useful. As expected, they have a one-way insertion slot, like post office mail bins. But they are also flexible enough so that it is possible to put one box inside another through the slot. She wonders whether this property might play a role in the solution to her problem, etc.

3.2. Our somewhat homomorphic scheme

Our somewhat homomorphic encryption scheme ε, described below, is remarkably simple.[9] We describe it first as a symmetric encryption scheme. As an example parameter setting, for security parameter λ, set $N = \lambda$, $P = \lambda^2$ and $Q = \lambda^5$.

An Encryption Scheme:

$\mathsf{KeyGen}_\varepsilon(\lambda)$: The key is a random P-bit odd integer p.

$\mathsf{Encrypt}_\varepsilon(p, m)$: To encrypt a bit $m \in \{0, 1\}$, set m' to be a random N-bit number such that $m' = m \bmod 2$. Output the ciphertext $c \leftarrow m' + pq$, where q is a random Q-bit number.

$\mathsf{Decrypt}_\varepsilon(p, c)$: Output $(c \bmod p) \bmod 2$, where $(c \bmod p)$ is the integer c' in $(-p/2, p/2)$ such that p divides $c - c'$.

Ciphertexts from ε are *near-multiples* of p. We call $(c \bmod p)$ the *noise* associated to the ciphertext c. It is the distance to the nearest multiple of p. Decryption works because the noise is m', which has the same parity as the message. We call a ciphertext output by Encrypt a *fresh* ciphertext, since it has small (N-bit) noise.

How is the scheme homomorphic? By simply adding, subtracting, or multiplying the ciphertexts as integers, we can add, subtract, or multiply (modulo 2) the underlying messages. However, complications arise, because these operations increase the noise associated to resulting ciphertexts. Eventually, the noise become so large that decryption no longer reliably returns the correct result.

Homomorphic Operations:

$\mathsf{Add}_\varepsilon(c_1, c_2)$, $\mathsf{Sub}_\varepsilon(c_1, c_2)$, $\mathsf{Mult}_\varepsilon(c_1, c_2)$: the output ciphertext c is $c_1 + c_2$, $c_1 - c_2$, or $c_1 \cdot c_2$.

$\mathsf{Evaluate}_\varepsilon(f, c_1, ..., c_t)$: Express the boolean function f as a circuit C with XOR and AND gates. Let $C^\dagger$ be the same circuit as C, but with XOR and AND gates replaced by addition and multiplication gates over the integers. Let $f^\dagger$ be the multivariate polynomial that corresponds to $C^\dagger$. Output $c \leftarrow f^\dagger(c_1, ..., c_t)$.

Let us check that ciphertexts output by $\mathsf{Evaluate}_\varepsilon$ decrypt correctly. As a warm-up, let us consider $\mathsf{Mult}_\varepsilon$. Let $c = c_1 \cdot c_2$, where c_i's noise is m'_i, which has the same parity as the message m_i. We have that

$$c = m'_1 \cdot m'_2 + pq'$$

for some integer q'. As long as the noises are small enough so that $|m'_1 \cdot m'_2| < p/2$, we have that

$$(c \bmod p) = m'_1 \cdot m'_2$$

and therefore $(c \bmod p) \bmod 2 = m_1 \cdot m_2$, as it should be. We will consider the evaluation of more complicated functions momentarily, in Section 3.3.

So far we only described a symmetric homomorphic encryption scheme. Turning it into a public-key scheme is easy, but adds some complexity. As before, the secret key is p. The public key consists of a list of integers that are

essentially "encryptions of zero." The list has length polynomial in λ. To encrypt a bit m, the ciphertext c is (essentially) m plus a random subset sum of the ciphertexts in the public key. If these ciphertexts have very small noise, the resulting ciphertext will also have small noise, and decryption will work properly: (c mod p) mod 2 will equal m, as before.

3.3. How homomorphic is it?

What is the set of permitted functions that our homomorphic encryption scheme $\mathcal{E}$ can handle?

To answer this question, we need to analyze how the noise grows as we add and multiply ciphertexts. $\mathsf{Encrypt}_{\mathcal{E}}$ outputs a *fresh* ciphertext with a small noise, at most N bits. As we $\mathsf{Add}_{\mathcal{E}}$, $\mathsf{Sub}_{\mathcal{E}}$, or $\mathsf{Mult}_{\mathcal{E}}$ ciphertexts, the output ciphertext becomes more noisy. Multiplication tends to increase the noise faster than addition or subtraction. In particular, for ciphertexts c_1 and c_2 with k_1- and k_2-bit noises, the ciphertext $c \leftarrow c_1 \cdot c_2$ has (roughly) $(k_1 + k_2)$-bit noise.

What happens when we perform *many* $\mathsf{Add}_{\mathcal{E}}$, $\mathsf{Sub}_{\mathcal{E}}$, and $\mathsf{Mult}_{\mathcal{E}}$ operations, as prescribed by the circuit representing a function f? Similar to what we saw above with multiplication, we have

$$f^{\dagger}(c_1, \ldots, c_t) = f^{\dagger}(m'_1, \ldots, m'_t) + pq'$$

for some integer q', where m'_t is the noise associated to c_i. If $|f^{\dagger}(m'_1, \ldots, m'_t)| < p/2$, then ($f^{\dagger}(c_1, \ldots, c_t)$ mod p) equals $f^{\dagger}(m'_1, \ldots, m'_t)$. And if we reduce this result modulo 2, we obtain the correct result: $f(m_1, \ldots, m_t)$.

In short, the functions that $\mathcal{E}$ can handle are those for which $|f^{\dagger}(a_1, \ldots, a_t)|$ is *always* less than $p/2$ if all of the a_i are at most N bits.

$\mathcal{E}$ is already quite powerful. As an example, it can handle an elementary symmetric polynomial of degree d in t variables, as long as $2^{Nd} \cdot \binom{t}{d} < p/2$, which is true (roughly) when $d < P/(N \cdot \log t)$. For our suggested parameters, this degree can be quite large: $\lambda/(\log t) = \Omega(\lambda/\log \lambda)$. That $\mathcal{E}$ can evaluate polynomials of such high degree makes it "homomorphic enough" for many applications. For example, it works well when f is a highly parallelizable function—e.g., a basic keyword search—in which case f has fairly low degree.

3.4. Semantic security and approximate GCDs

Euclid showed that, given two integers x_1 and x_2, it is easy to compute their greatest common divisor (gcd). But suppose that $x_1 = s_1 + p \cdot q_1$ and $x_2 = s_2 + p \cdot q_2$ are *near*-multiples of p, with s_1 and s_2 much smaller than p. When p is only an approximate gcd, is it still possible to compute p efficiently—i.e., in time polynomial in the bit-lengths of x_1 and x_2? Not in general, as far as we know.

In fact, if we sample s_i, p and q_i with λ, λ^2, and λ^5 bits (similar to our scheme $\mathcal{E}$), then the *approximate gcd problem* seems to remain hard even if we are given arbitrarily many samples $x_i = s_i + p \cdot q_i$, rather than just two. For these parameters, known attacks—including those using continued fractions and simultaneous diophantine approximation—take time essentially exponential in λ.

Moreover, we can reduce the approximate gcd problem to the security of our somewhat homomorphic encryption scheme. That is, we can prove that an attacker cannot efficiently break the semantic security of our encryption scheme unless the approximate gcd problem is easy.

4. BOOTSTRAPPABLE ENCRYPTION

4.1. Alice's eureka moment

One night, Alice dreams of immense riches, caverns piled high with silver, gold, and diamonds. Then, a giant dragon devours the riches and begins to eat its own tail! She awakes with a feeling of peace. As she tries to make sense of her dream, she realizes that she has the solution to her problem. She knows how to use her defective boxes to securely delegate the assembly of even the most intricate pieces!

Like before, she gives a worker a glovebox, box #1, containing the raw materials. But she also gives him several additional gloveboxes, where box #2 contains (locked inside) the key to box #1, box #3 contains the key to box #2, and so on. To assemble an intricate design, the worker manipulates the materials in box #1 until the gloves stiffen. Then, he places box #1 inside box #2, where the latter box already contains a the key to box #1. Using the gloves for box #2, he opens box #1 with the key, extracts the partially assembled trinket, and continues the assembly within box #2 until its gloves stiffen. He then places box #2 inside box #3, and so on. When the worker finally finishes his assembly inside box #n, he hands the box to Alice.

Of course, Alice observes, this trick does not work unless the worker can open box #i within box #$(i + 1)$, and still have time to make a little bit of progress on the assembly, all before the gloves of box #$(i + 1)$ stiffen. But as long as the unlocking operation (plus a little bit of assembly work) takes less than a minute, and as long as she has enough defective gloveboxes, then it is possible to assemble any piece, no matter how complicated!

4.2. A dream deciphered

In the analogy, the defective gloveboxes represent our somewhat homomorphic encryption scheme, which can perform Add, Sub, and Mult operations on ciphertexts for a little while—it can handle functions in a limited set $\mathcal{F}_{\mathcal{E}}$—until the noise becomes too large. What we would like to do is use this somewhat homomorphic scheme to construct a fully homomorphic one.

As before, box #1 with the precious materials inside represents the ciphertexts that encrypt the initial data. Box #$(i + 1)$ with the key for box i inside represents an *encrypted secret decryption key*—i.e., sk_i encrypted under pk_{i+1}.

In the analogy, Alice discovers that there is only one thing that her workers really need to be able to do in less than 1 min with the gloves, aside from performing a very small operation on the piece: unlock box #i within box #$(i + 1)$ and extract the piece. It will turn out that there is only one function that our scheme $\mathcal{E}$ really needs to be able to handle, with a tiny bit of room left over to perform one more Add, Sub, or Mult: the decryption function (which is like unlocking the "encryption box").

If ε has this self-referential property of being able to handle its own decryption function (augmented by a single gate), we say that it is *bootstrappable*. As we will show, if ε is bootstrappable, then one can use ε to construct a fully homomorphic encryption scheme $\varepsilon^{\dagger}$.

4.3. Bootstrappable to fully homomorphic

Suppose that ε is bootstrappable. In particular, suppose that ε can handle the following four functions: the decryption function, expressed as a circuit D_ε of size polynomial in λ, as well as D_ε augmented by an Add, Sub, or Mult gate modulo 2. (D_ε augmented by Add consists of two copies of D_ε connected by an Add gate.) We will show that this is a *complete* set of circuits, in the sense that if these four circuits are in $\mathcal{F}_\varepsilon$, then one can construct from ε a scheme $\varepsilon^{\dagger}$ that is fully homomorphic.

As a warm-up, suppose that ciphertext c_1 encrypts the bit m under key pk_1. Suppose also that we have an encrypted secret key: let $\overline{\mathrm{sk}_1}$ be a vector of ciphertexts that encrypt the bits of sk_1 under pk_2 via $\mathsf{Encrypt}_\varepsilon(\mathrm{pk}_2, \mathrm{sk}_{1j})$. Consider the following algorithm.

$\mathsf{Recrypt}_\varepsilon(\mathrm{pk}_2, D_\varepsilon, \overline{\mathrm{sk}_1}, c_1)$.

Generate $\overline{c_1}$ via $\mathsf{Encrypt}_\varepsilon(\mathrm{pk}_2, c_{1j})$ over the bits of c_1
Output $c \leftarrow \mathsf{Evaluate}_\varepsilon(\mathrm{pk}_2, D_\varepsilon, \overline{\mathrm{sk}_1}, \overline{c_1})$

The decryption circuit D_ε has input wires for the bits of a secret key and the bits of a ciphertext. Above, $\mathsf{Evaluate}_\varepsilon$ takes in the bits of sk_1 and c_1, each encrypted under pk_2. Then, ε is used to evaluate the decryption circuit homomorphically. As long as ε can handle D_ε, the output c is an encryption under pk_2 of $\mathsf{Decrypt}_\varepsilon(\mathrm{sk}_1, c_1) = m$. $\mathsf{Recrypt}_\varepsilon$ therefore outputs a new encryption of m, but under pk_2.

One fascinating thing about $\mathsf{Recrypt}_\varepsilon$ is that the message m is doubly encrypted at one point, first under pk_1 and next under pk_2. Ordinarily, the only thing one can do with a doubly encrypted message is to peel off the outer encryption first, and then decrypt the inner layer. However, in $\mathsf{Recrypt}_\varepsilon$, the $\mathsf{Evaluate}_\varepsilon$ algorithm is used to remove the *inner* encryption, just like Alice unlocks box $\#i$ while it is inside box $\#(i+1)$.

It is also useful to imagine that ε is our somewhat homomorphic encryption scheme from Section 3, and consider what $\mathsf{Recrypt}_\varepsilon$ does to the noise of the ciphertexts. Evaluating D_ε removes the noise associated to the first ciphertext under pk_1 (because, of course, decryption removes noise), but $\mathsf{Evaluate}_\varepsilon$ simultaneously introduces new noise while evaluating the ciphertexts under pk_2. As long as the new noise added is less than the old noise removed, we have made "progress." A similar situation holds in Alice's jewelry store. When the worker extracts the piece from the used-up glovebox $\#i$, this process simultaneously uses up the gloves of box $\#(i+1)$. We have made "progress" as long as the process does not leave box $\#(i+1)$'s gloves completely used-up.

Of course, our goal is to perform actual operations on underlying messages, not merely to obtain a new encryption of the same message. So, suppose that ε can handle D_ε augmented by some gate—e.g., Add; call this augmented circuit D_{Add}. If c_1 and c_2 are two ciphertexts that encrypt m_1 and m_2, respectively, under pk_1, and we compute $\overline{c_1}$ and $\overline{c_2}$ as before, as ciphertexts encrypting the bits of the ciphertexts under pk_2, then we have that

$$c \leftarrow \mathsf{Evaluate}_\varepsilon(\mathrm{pk}_2, D_{\mathsf{Add}}, \overline{\mathrm{sk}_1}, \overline{c_1}, \overline{c_2})$$

is an encryption under pk_2 of $m_1 \oplus m_2$.

By recursing this process, we get a fully homomorphic encryption scheme. The public key in $\varepsilon^{\dagger}$ consists of a sequence of public keys $(\mathrm{pk}_1, \ldots, \mathrm{pk}_{\ell+1})$ and a chain of encrypted secret keys $\overline{\mathrm{sk}_1}, \ldots, \overline{\mathrm{sk}_\ell}$, where sk_i is encrypted under pk_{i+1}. To evaluate a function f in $\varepsilon^{\dagger}$, we express f as a circuit, topologically arrange its gates into levels, and step through the levels sequentially. For a gate at level $i+1$ (e.g., an Add gate), we take as input the encrypted secret key $\overline{\mathrm{sk}_i}$ and a couple of ciphertexts associated to output wires at level i that are under pk_i, and we homomorphically evaluate D_{Add} to get a ciphertext under pk_{i+1} associated to a wire at level $i+1$. Finally, we output the ciphertext associated to the output wire of f.

Putting the encrypted secret key bits $\overline{\mathrm{sk}_1}, \ldots, \overline{\mathrm{sk}_\ell}$ in $\varepsilon^{\dagger}$'s public key is not a problem for security. These encrypted secret-key bits are indistinguishable from encryptions of 0 as long as ε is semantically secure.

4.4. Circular security

Strictly speaking, $\varepsilon^{\dagger}$ does not *quite* meet our definition of fully homomorphic encryption, since the complexity of $\mathsf{KeyGen}_{\varepsilon^{\dagger}}$ grows linearly with the maximum circuit depth we want to evaluate. (Fortunately, $\mathsf{Encrypt}_{\varepsilon^{\dagger}}$ and $\mathsf{Decrypt}_{\varepsilon^{\dagger}}$ do not depend at all on the function f being evaluated.)

However, suppose that ε is not only bootstrappable, but also *circular-secure*—that is, it is "safe" to reveal the encryption of a secret key sk_i under its own associated public key pk_i. Then, we can simplify $\mathsf{KeyGen}_\varepsilon{}^{\dagger}$. We do not need distinct public keys pk_i for each circuit level and an acyclic chain of encrypted secret keys. Instead, the public key in $\varepsilon^{\dagger}$ can consist merely of a single public key pk and a single encrypted secret key $\overline{\mathrm{sk}}$ (sk under pk), where pk is associated to all levels of the circuit. This approach has the additional advantage that we do not need to decide beforehand the maximal circuit depth complexity of the functions that we want to be able to evaluate.

For most encryption schemes, including our somewhat homomorphic scheme (as far as we know), revealing an encryption of sk under pk does not lead to any attack. However, it is typically difficult to *prove* that an encryption scheme is circular-secure.

The issue of circular security also fits within our physical analogy. Suppose that a key is locked inside the very same box that the key could open from the outside. Is it possible to use the gloves and key to open the box *from the inside*? If so, it would be a strange lock. Similarly, encryption schemes that are insecure in this setting tend to be contrived.

5. SOMEWHAT HOMOMORPHIC TO BOOTSTRAPPABLE

Is our somewhat homomorphic encryption scheme from Section 3 already bootstrappable? Can it handle its own

decryption circuit? Unfortunately, as far as we can tell, ε can *almost* handle D_ε, but not quite. So, we modify ε slightly, constructing a new (but closely related) somewhat homomorphic scheme ε^* that can handle essentially the same functions that ε can, but whose decryption circuit is simple enough to make ε^* bootstrappable.

5.1. Alice gets her hands dirty

After her dream, Alice rushes to her store to see if her idea works. She locks box #1 and puts it inside box #2. Working with the gloves of box #2, she tries to unlock box #1 in less than 1 min. The thickness of the gloves and the stickiness of the lock combine to make it impossible.

She is despondent until she remembers that she has a special grease that makes her locks less sticky. This time, she locks box #3 and puts it inside box #4. She also puts her bottle of grease inside box #4. Working with the gloves of box #4, she squirts some grease on the lock and then tries to unlock it. But the gloves stiffen before she can finish.

Then, she thinks: why didn't I grease the box's lock *before* putting it inside the other box? That way, I wouldn't waste my valuable time with the gloves greasing the lock.

She locks box #5, greases its lock, and then puts it inside box #6. Working with gloves, she tries the lock again. This time it works, despite the clumsiness of the gloves!

At last, she has a system that lets her securely delegate the processing of her precious materials into arbitrarily complicated pieces! Her workers just need to apply the grease to each box before they put it inside the next box. She can hardly wait to put the system in place the following morning.

5.2. Greasing the decryption circuit

In our somewhat homomorphic encryption scheme ε from Section 3, the decryption function is:

$$m \leftarrow (c \bmod p) \bmod 2$$

Equivalently, but more simply, the equation is:

$$m \leftarrow \text{LSB(c)} \ \ \text{XOR} \ \ \text{LSB}(\lfloor c/p \rceil),$$

where LSB takes the least significant bit and $\lfloor \cdot \rceil$ rounds to the nearest integer. This is equivalent, since $(c \bmod p) = c - p \cdot \lfloor c/p \rceil$. Since p is odd, we have that $(c \bmod p) \bmod 2 = c - \lfloor c/p \rceil \bmod 2$. This is just the XOR of the least significant bits of c and $\lfloor c/p \rceil$.

In the decryption circuit D_ε, computing the LSB is immediate: the circuit simply does not have output wires for the more significant bits. Computing an XOR also takes only one gate. If the decryption function is complicated, it must be because computing $\lfloor c/p \rceil$ is complicated. Is the function $f(p, c) = \lfloor c/p \rceil$ (with the few steps afterward) something that ε can handle? If so, ε is bootstrappable, and can be used to construct a fully homomorphic encryption scheme.

Unfortunately, even a single multiplication of long numbers—namely, c with $1/p$—seems to be too complex for ε to handle. The reason is that c and $1/p$ each need to be expressed with at least $P \approx \log p$ bits of precision to ensure that $f(p, c)$ is computed correctly. When you multiply two P-bit numbers, a bit of the result may be a high-degree polynomial of the input bits; this degree is also roughly P. We saw that ε can handle an elementary symmetric polynomial in t variables of degree (roughly) $d < P/(N \cdot \log t)$. However, ε cannot handle even a single monomial of degree P, where the noise of output ciphertext is upper-bounded by $(2^N)^P \approx p^N \gg p/2$. Consequently, ε does not seem to be bootstrappable.

However, if we are willing to get our hands dirty by tinkering with ε to make the decryption function simpler, we eventually get a scheme ε^* that is bootstrappable. The main idea of the transformation is to replace ε's decryption function, which multiplies two long numbers, with a decryption function that adds a fairly small set of numbers. In terms of the bits of the addends, this summation corresponds to a polynomial of fairly low degree that ε^* can handle.

Let us go through the transformation step by step, beginning with $\text{KeyGen}_{\varepsilon^*}$. The transformation uses a couple of integer parameters: $0 < \alpha < \beta$.

- $\text{KeyGen}_{\varepsilon^*}(\lambda)$: Run $\text{KeyGen}_\varepsilon(\lambda)$ to obtain keys (pk, sk), where sk is an odd integer p. Generate a set $\vec{y} = \langle y_1, \ldots, y_\beta \rangle$ of rational numbers in $[0, 2)$ such that there is a sparse subset $S \subset \{1, \ldots, \beta\}$ of size α with $\Sigma_{i \in S} y_i \approx 1/p \bmod 2$. Set sk* to be the sparse subset S, encoded as a vector $s \in \{0, 1\}^\beta$ with Hamming weight α. Set $\text{pk}^* \leftarrow (\text{pk}, \vec{y})$.

The important difference between $\text{KeyGen}_{\varepsilon^*}$ and $\text{KeyGen}_\varepsilon$ is that $\text{KeyGen}_{\varepsilon^*}$ includes a *hint* about the secret integer p—namely, a set of numbers $\vec{y}$ that contains a (hidden) sparse subset that sums to $1/p$ (to within a very small error, and up to addition by an even number). This hint is the "grease," which will be used in $\text{Encrypt}_{\varepsilon^*}$ and $\text{Decrypt}_{\varepsilon^*}$. Although it is technically not the decryption key sk*, the integer p still can be used to decrypt a ciphertext output by $\text{Encrypt}_{\varepsilon^*}$, so revealing this hint obviously impacts security, a point we elaborate on in Section 5.4.

- $\text{Encrypt}_{\varepsilon^*}(\text{pk}^*, m)$: Run $\text{Encrypt}_\varepsilon(\text{pk}, m)$ to obtain ciphertext c. For $i \in \{1, \ldots, \beta\}$, set $z_i \leftarrow c \cdot y_i \bmod 2$ keeping only about $\log \alpha$ bits of precision after the binary point for each z_i. The ciphertext c^* consists of c and $\vec{z} = \langle z_1, \ldots, z_\beta \rangle$.

The important point here is that the hint $\vec{y}$ is used to *postprocess* a ciphertext c output by $\text{Encrypt}_\varepsilon$, with the objective of leaving less work remaining for $\text{Decrypt}_{\varepsilon^*}$ to do.

This sort of two-phase approach to decryption has been used before in *server-aided cryptography*. (See cites in Gentry[2].) In that setting, a user wants to minimize its cryptographic computation—e.g., because it is using a constrained device, such as a smartcard or handheld. So, it outsources expensive computations to a server. To set up this arrangement, the user (in some schemes) must give the server a hint $\vec{y}$ that is statistically dependent on its secret key sk, but which is not sufficient to permit the server to decrypt efficiently on its own. The server uses the hint to

process a ciphertext directed to the user, leaving less work for the user to do. In our setting, the encrypter or evaluator plays the role of the server, postprocessing the ciphertext so as to leave less work for the decryption algorithm to do.

- $\mathsf{Decrypt}_{\varepsilon^*}(\mathrm{sk}^*, c^*)$: Output $\mathrm{LSB}(c)$ XOR $\mathrm{LSB}(\lfloor \Sigma_i s_i z_i \rceil)$. Decryption works, since (up to small precision errors) $\Sigma_i s_i z_i = \Sigma_i c \cdot s_i y_i = c/p \bmod 2$.

To ensure that the rounding is correct despite the low precision, we need c to be closer (than the trivial $p/2$) to a multiple of p (say, within $p/16$). This makes $\mathcal{F}_{\varepsilon^*}$ smaller than $\mathcal{F}_{\varepsilon}$, since $\mathcal{F}_{\varepsilon^*}$ is limited to functions where $|f(a_1, \ldots, a_t)| < p/16$ when the a_i are N bits. This makes only a small difference.

The important point regarding $\mathsf{Decrypt}_{\varepsilon^*}$ is that we replace the multiplication of c and $1/p$ with a summation that contains only α nonzero terms. The bits of this summation can be computed by a polynomial of degree $\alpha \cdot \mathrm{polylog}(\alpha)$, which ε^* can handle if we set α to be small enough.

- $\mathsf{Add}_{\varepsilon^*}(\mathrm{pk}^*, c_1^*, c_2^*)$: Extract c_1 and c_2 from c_1^* and c_2^*. Run $c \leftarrow \mathsf{Add}_{\varepsilon}(\mathrm{pk}, c_1, c_2)$. The output ciphertext c^* consists of c, together with the result of postprocessing c with $\vec{y}$. $\mathsf{Mult}_{\varepsilon^*}(\mathrm{pk}^*, c_1^*, c_2^*)$ is analogous.

5.3. How to add numbers

To see that ε^* can handle the decryption function plus an additional gate when α is set small enough, let us consider the computation of the sum $\Sigma_i s_i z_i$. In this sum, we have β numbers $a_1, \ldots, a_\beta$, each a_i expressed in binary $(a_{i,0}, \ldots, a_{i,-\ell})$ with $\ell = O(\log \alpha)$, where at most α of the a_i's are nonzero (since the Hamming weight of s is α). We want to express each bit of the output as a polynomial of the input bits, while minimizing the degree of the polynomial and the number of monomials.

Our approach to the problem is to add up the column of LSBs of the numbers—computing the Hamming weight of this column—to obtain a number in binary representation. Then, we add up the column of penultimate bits, etc. Afterward, we combine the partial results. More precisely, for $j \in [0, -\ell]$, we compute the Hamming weight b_j, represented in binary, of $(a_{1,j}, \ldots, a_{\beta,j})$. Then, we add up the $\ell + 1$ numbers $b_0, \ldots, 2^{-\ell} b_{-\ell}$ to obtain the final correct sum.

Conveniently, the binary representation of the Hamming weight of any vector $\vec{x} \in \{0,1\}^t$ is given by

$$(e_{2\lfloor \log t \rfloor}(x_1, \ldots, x_t) \bmod 2, \ldots, e_2 0(x_1, \ldots, x_t) \bmod 2)$$

where $e_i(x_1, \ldots, x_t)$ is the ith elementary symmetric polynomial over $x_1, \ldots, x_t$. These polynomials have degree at most t. Also, we know how to efficiently evaluate the elementary symmetric polynomials. They are simply coefficients of the polynomial $p(z) = \prod_{i=1}^{t} (z - x_i)$. An important point is that, in our case, we only need to evaluate the polynomials up to degree α, since we know a priori that each of the Hamming weights is at most α. We saw in Section 3.3 that we can handle elementary symmetric polynomials in t variables of degree up to about $\lambda/\log t = \Omega(\lambda/\log \lambda)$ for our suggested parameters. We can set α to be smaller than this.

The final step of computing the sum of the b_j's does not require much computation, since there are only $\ell + 1 = O(\log \alpha)$ of them. We get that a ciphertext encrypting a bit of the overall sum has noise of at most $N \cdot \alpha \cdot g(\log \alpha)$ bits for some polynomial g of low degree. If the final sum modulo 2 is $(b'_0, b'_{-1}, \ldots)$ in binary, then the rounding operation modulo 2 is simply b'_0 XOR b'_{-1}. With the additional XOR operation in decryption, and possibly one more gate, the noise after evaluating the decryption function plus a gate has at most $N \cdot \alpha \cdot h(\log \alpha)$ bits for some polynomial h.

The scheme ε^* becomes bootstrappable when this noise has at most $\log(p/16) = P - 4$ bits. For example, this works when $\alpha = \lambda/\mathrm{polylog}(\lambda)$, $N = \lambda$, and $P = \lambda^2$.

5.4. Security of the transformed scheme

The encryption key of ε^* contains a hint about the secret p. But we can prove that ε^* is semantically secure, unless either it is easy to break the semantic security of ε (which implies that the approximate gcd problem is easy), or the following sparse (or low-weight) subset sum problem (SSSP) is easy: given a set of β numbers $\vec{y}$ and another number s, find the sparse (α-element) subset of $\vec{y}$ whose sum is s.

The SSSP has been studied before in connection with server-aided cryptosystems. If α and β are set appropriately, the SSSP is a hard problem, as far as we know. In particular, if we set α to be about λ, it is hard to find the sparse subset by "brute force," since there are $\binom{\beta}{\alpha} \approx \beta^\alpha$ possibilities. If the sparse subset sum is much closer to $1/p$ than any other subset sum, the problem yields to a lattice attack. But these attacks fail when we set β large enough (but still polynomial in λ) so that an *exponential* (in λ) number of subset sums are as close to $1/p$ as the sparse subset. Concretely, we can set $\beta = \lambda^5 \cdot \mathrm{polylog}(\lambda)$.

6. CONCLUSIONS

We now know that FHE is possible. We already have the scheme presented here, the lattice-based scheme by Gentry,[2,3] and a recent scheme by Smart and Vercauteren.[7]

There is still work to be done toward making FHE truly practical. Currently, all known FHE schemes follow the blueprint above: construct a bootstrappable somewhat homomorphic encryption scheme ε, and obtain FHE by running $\mathsf{Evaluate}_{\varepsilon}$ on ε's decryption function. But this approach is computationally expensive. Not only is the decryption function expressed (somewhat inefficiently) as a circuit, but then $\mathsf{Evaluate}_{\varepsilon}$ replaces each bit in this circuit with a large ciphertext that encrypts that bit. Perhaps someone will find a more efficient blueprint.

The scheme presented here, while conceptually simpler, seems to be less efficient than the lattice-based scheme. To get 2^λ security against known attacks—e.g., on the the approximate gcd problem—ciphertexts are $\lambda^5 \cdot \mathrm{polylog}(\lambda)$ bits, which leads to $\lambda^{10} \cdot \mathrm{polylog}(\lambda)$ computation to evaluate the decryption function. The lattice-based scheme with comparable security has $\lambda^6 \cdot \mathrm{polylog}(\lambda)$ computation. This is high, but not totally unreasonable. Consider: to make RSA 2^λ-secure against known attacks—in particular, against the number field sieve factoring algorithm—you need to use an RSA modulus with approximately λ^3

bits. Then, RSA decryption involves exponentiation by a λ^3-bit exponent—i.e., about λ^3 multiplications. Even if one uses fast Fourier multiplication, this exponentiation requires $\lambda^6 \cdot \text{polylog}(\lambda)$ computation. Also, unlike RSA, the decryption function in our scheme is highly parallelizable, which may make an enormous difference in some implementations.

7. EPILOGUE

The morning after her dream, Alice explains her glovebox solution to her workers. They are not happy, but they wish to remain employed. As the day progresses, it becomes clear that the gloveboxes are slowing down the pace of jewelry construction considerably. The main problem seems to be the thick gloves, which multiply the time needed for each assembly step. After a few days of low output, Alice curtails her use of the gloveboxes to pieces that contain the most valuable diamonds.

Alice loses her suit against Acme Glovebox Company, because, as far as anyone knows in Alice's parallel world, gloves in gloveboxes are always very stiff and stiffen completely after moderate use. The old judge explains this to her in a patronizing tone.

But Alice refuses to give up. She hires a handsome young glovebox researcher, and tasks him with developing a glove flexible enough to permit the nimble assembly of jewels and unlocking of boxes, but sturdy enough to prevent the boxes from being easily compromised. The researcher, amazed at his good fortune, plunges into the problem. C

References

1. Boneh, D., Lipton, R.J. Algorithms for black-box fields and their application to cryptography (extended abstract). In *CRYPTO* (1996), 283–297.
2. Gentry, C. *A fully Homomorphic Encryption Scheme*. Ph.D. thesis, Stanford University, 2009. crypto.stanford.edu/craig.
3. Gentry, C. Fully homomorphic encryption using ideal lattices. *STOC*. M. Mitzenmacher ed. ACM, 2009, 169–178.
4. Goldwasser, S., Micali, S. Probabilistic encryption. *J. Comp. Syst. Sci. 28*, 2 (1984), 270–299.
5. Rivest, R.L., Adleman, L.M., Dertouzos, M.L. On data banks and privacy homomorphisms. In *Foundations of Secure Computations* (1978), 169–180.
6. Rivest, R.L., Shamir, A., Adleman, L.M. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM 21*, 2 (1978), 120–126.
7. Smart, N.P., Vercauteren, F. Fully homomorphic encryption with relatively small key and ciphertext sizes, 2009. http://eprint.iacr.org/2009/571.
8. van Dam, W., Hallgren, S., Ip, L. Quantum algorithms for some hidden shift problems. *SIAM J. Comp. 36*, 3 (2006), 763–778.
9. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V. Fully homomorphic encryption over the integers, 2009. http://eprint.iacr.org/2009/616.

Craig Gentry (cbgentry@us.ibm.com), IBM T.J. Watson Research Center, Hawthorne, NY.

DOI:10.1145/1666420.1666443

Technical Perspective
Seeing the Trees, the Forest, and Much More

By Pietro Perona

YOUR PORTABLE PHONE can beat you at chess, but can it recognize a horse? Bristling with cameras, microphones, and other sensors, today's machines are nevertheless essentially deaf and blind; they do not have senses to interact with their environment. In the meantime, vast amounts of valuable sensory data is captured, transmitted, and inexpensively stored every day. TV programs and movies, fMRI scans, planetary surveys, footage from security cameras, and digital photographs pile up and lie fallow on hard drives around the globe. It is all too much for humans to organize and access by hand. Someone has appropriately called this the "data deluge." Automating the process of analyzing sensory data and transforming it into actionable information is one of the most useful and difficult challenges of modern engineering.

How shall we go about building machines that can see, hear, smell, touch? Sensory tasks come in all shapes and forms: reading books, recognizing people, or hitting tennis balls. It is expeditious to approach each one as a separate problem. However, one remarkable fact about our own senses is they adapt easily to new environments and tasks. Our senses evolved to help us navigate and forage among trees, rocks, and grass, as well as enable us to socialize with people. Despite this history, we can train ourselves to read text, to recognize galaxies in telescope images, and to drive fast-moving vehicles. Discovering general laws and principles that underlie sensory processing might one day allow us to design and build flexible and adaptable sensory systems for our machines.

In the following paper, Torralba, Murphy, and Freeman are concerned with visual recognition. They explore one principle that has general validity: the use of context. The authors propose an elegant and compelling demonstration showing that context is crucial for recognizing an object when the image has poor resolution and, as a result, the object's picture is ambiguous. That context may be useful in visual recognition is rather intuitive. However, to design a machine that makes use of context we must first define what context is, exactly how should one measure it, and how these measurements may be used to recognize objects.

The context of an object is a rich and complex phenomenon, and it is not easily defined. The identity of the scene (suburban street, kitchen) where the object is found could be thought of as its context. The identity of the surfaces and objects present in the scene (two automobiles, a pedestrian, a fire hydrant, a building's facade), as well as the mutual position of such surfaces and objects, are also considered context. So, too, is the weather, lighting conditions, time of day, historical period, and other circumstances. Where should one begin? What should one measure? One could worry that the entire problem of vision must be solved before one is able to define and compute context. It is not surprising that most researchers to date have sidestepped this baffling chicken-and-egg issue.

Discovering general laws and principles that underlie sensory processing might one day allow us to design and build flexible and adaptable sensory systems for our machines.

The authors avoid computing explicit scene semantic information. They start instead by considering easy-to-compute, image-like quantities that correlate with context. Inspired by what we know about the human visual system, they compute statistics of the output of wavelet-like linear filters applied to the image. These statistics capture some aspects of the visual statistics of the scene that, in turn, are indicative of its overall nature: for example, long and vertical structure in a forest, sparse horizontal structure in open grassland. Filter statistics are thus correlated to scene type. Torralba, Murphy, and Freeman call the ensemble of their measurements "gist," a term used in psychology to denote the overall visual meaning of a scene, which has been shown to be perceived quickly by human observers.[1,2]

The authors find that, surprisingly, their filter-based gist is rather good at predicting the number of instances of a given object category that might be present in the scene, as well as their likely position along the y-axis. Combining this with information coming from object detectors operating independently at each location produces an overall score for the presence of an object of a given class at location $(x; y)$. This is more reliable than using the detectors alone. It looks like it is *finally* open season on visual context. C

References

1. Biederman, I. Perceiving real-world scenes. *Science 177* (1972), 77-80.
2. Fei-Fei, L., Iyer, A., Koch, C., and Perona, P. What do we perceive in a glance of a real-world scene? *Journal of Vision 7*, 1534-7362 (2007), 1-29.

Pietro Perona is the Allen E. Puckett Professor of Electrical Engineering at the California Institute of Technology, Pasadena, where he directs Computation and Neural System—a Ph.D. program centered on the study of biological brains and intelligent machines.

DOI:10.1145/1666420.1666446

Using the Forest to See the Trees: Exploiting Context for Visual Object Detection and Localization

By A. Torralba, K.P. Murphy, and W.T. Freeman

Abstract

Recognizing objects in images is an active area of research in computer vision. In the last two decades, there has been much progress and there are already object recognition systems operating in commercial products. However, most of the algorithms for detecting objects perform an exhaustive search across all locations and scales in the image comparing local image regions with an object model. That approach ignores the semantic structure of scenes and tries to solve the recognition problem by brute force. In the real world, objects tend to covary with other objects, providing a rich collection of contextual associations. These contextual associations can be used to reduce the search space by looking only in places in which the object is expected to be; this also increases performance, by rejecting patterns that look like the target but appear in unlikely places.

Most modeling attempts so far have defined the context of an object in terms of other previously recognized objects. The drawback of this approach is that inferring the context becomes as difficult as detecting each object. An alternative view of context relies on using the entire scene information holistically. This approach is algorithmically attractive since it dispenses with the need for a prior step of individual object recognition. In this paper, we use a probabilistic framework for encoding the relationships between context and object properties and we show how an integrated system provides improved performance. We view this as a significant step toward general purpose machine vision systems.

1. INTRODUCTION

Visual object detection, such as finding cars and people in images, is an important but challenging task. It is important because of its inherent scientific interest (understanding how to make machines see may shed light on biological vision), and because it is useful for many applications, such as content-based image retrieval, robotics, etc. It is challenging because the appearance of objects can vary a lot from instance to instance, and from image to image, due to factors such as variation in pose, lighting, style, articulation, occlusion, low quality imaging, etc.

Over the last two decades, much progress has been made in visual object detection using machine learning techniques. Most of these approaches rely on using supervised learning to train a classifier to distinguish between instances of the object class and the background. The trained classifier is then applied to thousands of small overlapping patches or windows of each test image, and the locations of the high-confidence detections are returned. The features computed inside each patch are usually the outputs of standard image processing operations, such as a histogram of responses to Gabor filters at different scales and orientations. The classifiers themselves are standard supervised learning models such as SVMs, neural networks, or boosted decision stumps.[20]

This "sliding window classifier" technique has been quite successful in certain domains such as detecting cars, pedestrians, and faces. Indeed most contemporary digital cameras imply such a technique to detect faces, which they use to set the auto-focus. Also, some cars now come equipped with pedestrian detection systems based on similar principles.

One major problem with the standard approach is that even a relatively low false-positive rate per class can be unacceptable when there are many classes or categories. For example, if each detector generates about 1 false alarm every 10 images, and there are 1000 classes, we will have 100 false alarms per image. An additional problem is that running every detector on every image can be slow. These are both fundamental obstacles to building a general purpose vision system.

One reason for the relatively high false alarm rate of standard approaches is that most object detection systems are "myopic," in the sense that they only look at local features of the image. One possible remedy is to leverage global features of the image, and to use these to compute the "prior" probability that each object category is present, and if so, its likely location and scale. Previous work (e.g., Torralba[17])

An early version of this paper, entitled "Using the forest to see the trees: a graphical model relating features, objects and scenes," was published in *Neural Information Processing Systems*, 2003, MIT Press. Ref. [9].

has shown that simple global image features, known as the "gist" of the image, are sufficient to provide robust predictions about the presence and location of different object categories. Such features are fast to compute, and provide information that is useful for many classes and locations simultaneously.

In this paper, which is an extension of our previous work,[8,9,17] we present a simple approach for combining standard sliding-window object detection systems, which use local, "bottom up" image features, with systems that predict the presence and location of object categories based on global, or "top-down," image features. These global features serve to define the context in which the object detection is happening. The importance of context is illustrated in Figure 1, which shows that the same black "blob," when placed in different surroundings, can be interpreted as a plate or bottle on the table, a cell phone, a pedestrian or car, or even a shoe. Another example is shown in Figure 2: it is easy to infer that there is very probably a computer monitor behind the blacked out region of the image.

We are not the first to point out the importance of context in computer vision. For example, Strat and Fischler emphasized its importance in their 1991 paper.[16] However, there are two key differences between our approach and previous work. First, in early work, such as[16] the systems consist of hand-engineered if–then rules, whereas more recent systems rely on statistical models that are fit to data. Second, most other approaches define the context in terms of other objects[6, 13, 14, 18]; but this introduces a chicken-and-egg problem: to detect an object of type 1 you first have to detect an object of type 2. By contrast, we propose a hierarchical approach, in which we define the context in terms of an overall scene category. This can be reliably inferred using global images features. Conditioned on the scene category, we assume that objects are independent. While not strictly true, this results in a simple yet effective approach, as we will show below.

Figure 1. In presence of image degradation (e.g., blur), object recognition is strongly influenced by contextual information. The visual system makes assumptions regarding object identities based on its size and location in the scene. In these images, the same black blob can be interpreted as a plate, bottle, cell phone, car, pedestrian, or shoe, depending on the context. (Each circled blob has identical pixels, but in some cases has been rotated.)

Figure 2. What is hidden behind the mask? In this example, context is so strong that one can reliably infer that the hidden object is a computer monitor.

In the following sections, we describe the different components of our model. We will start by showing how we can represent contextual information without using objects as an intermediate representation. Then we will show how that representation can be integrated with an object detector.

2. GLOBAL IMAGE FEATURES: THE GIST OF AN IMAGE

In the same way that an object can be recognized without decomposing it into a set of nameable parts (e.g., the most successful face detectors do not try to detect the eyes and mouth first, instead they search for less semantically meaningful features), scenes can also be recognized without necessarily decomposing them into objects. The advantage of this is that it provides an additional source of information that can be used to provide contextual information for object recognition. As suggested in Oliva and Schyns and Oliva and Torralba,[10,11] it is possible to build a global representation of the scene that bypasses object identities, in which the scene is represented as a single entity. Recent work in computer vision has highlighted the importance of global scene representations for scene recognition[1,7,11] and as a source of contextual information.[3,9,17] These representations are based on computing statistics of low level features (similar to representations available in early visual areas such as oriented edges, vector quantized image patches, etc.) over fixed image regions. One example of a global image representation is the

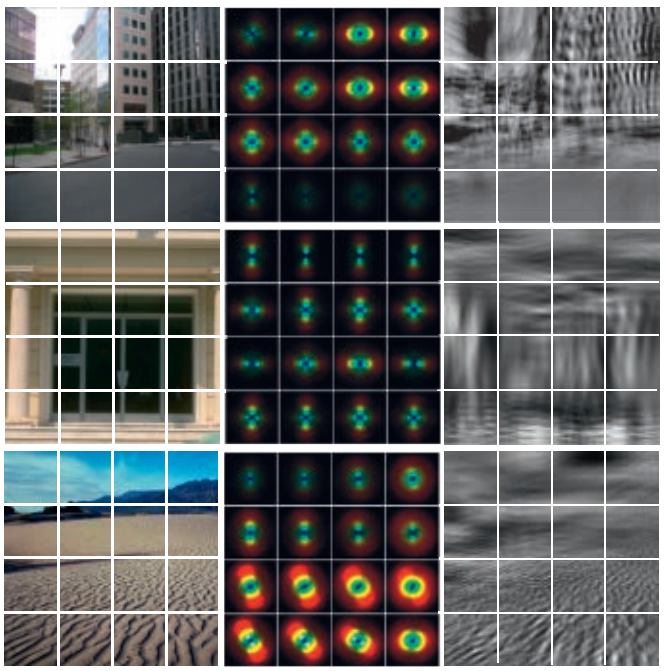
gist descriptor.[11] The gist descriptor is a vector of features g, where each individual feature g_k is computed as

$$g_k = \sum_{x,y} w_k(x,y) \times |I(x,y) \otimes h_k(x,y)|^2 \quad (1)$$

where $\otimes$ denotes image convolution and $\times$ is a pixel-wise multiplication. $I(x, y)$ is the luminance channel of the input image, $h_k(x, y)$ is a filter from a bank of multiscale-oriented Gabor filters (six orientations and four scales), and $w_k(x, y)$ is a spatial window that will compute the average output energy of each filter at different image locations. The windows $w_k(x, y)$ divide the image in a grid of 4×4 nonoverlapping windows. This results in a descriptor with a dimensionality of $4 \times 4 \times 6 \times 4 = 384$.

Figure 3 illustrates the amount of information preserved by the gist descriptor. The middle column shows the average of the output magnitude of the multiscale-oriented filters on a polar plot (note that the orientation of each plot is orthogonal to the direction of the edges in the image). The average response of each filter is computed locally by splitting the image into 4×4 windows. Each different scale is color coded (red for high spatial frequencies, and blue for the low spatial frequencies), and the intensity is proportional to the energy for each filter output. In order to illustrate the amount of information preserved by this representation, the right column of Figure 3 shows noise images that are coerced to have the same gist features as the target image, using the texture synthesis method of Heeger and Bergen.[2] As shown in Figure 3, the gist descriptor provides a coarse description of the textures present in the image and their spatial organization. The gist descriptor preserves relevant information needed for categorizing scenes into categories (e.g., classifying an image as being a beach scene, a street or a living-room). As reported in Quattoni and Torralba,[12] when trying to discriminate across 15 different scene categories, the gist descriptor classifies correctly 75% of the images. Recognizing the scene depicted by a picture is an important task on its own, but in addition it can be used to provide strong contextual priors as we will discuss in the next section.

Figure 3. This figure illustrates the information encoded by the gist features for three different images. See text for details.



3. JOINT SCENE CLASSIFICATION AND OBJECT DETECTION

In this section, we describe our approach in more detail. In Section 3.1, we briefly describe the standard approach to object detection and localization using local features. In Sections 3.3 and 3.2 we describe how to use global features for object localization and detection respectively. In Section 3.4 we discuss how to integrate these local and global features. A comparison of the performance of local and global features is deferred until Section 4.

3.1. Object presence detection and localization using local features

In our previous paper,[9] we considered detecting four different types or classes of objects: cars, people, keyboards, and screens (computer monitors). In this paper, we will mostly focus on cars, for brevity. We use a subset of the LabelMe dataset[11, 15] for training and testing (details are in Section 4).

There are two tasks that we want to address: object presence detection (where the goal is to predict if the object is present or absent in the image, i.e., to answer the question: is there any car in this image?) and object localization (where the goal is to precisely locate all the instances of an object class within each image). Solving the object presence detection task can be done even if the object localization is not accurate.

We can formalize the object presence detection and localization problem as follows. Let $P^t = 1$ if one or more objects of type t are present anywhere in the image, and $P^t = 0$ otherwise. The goal of object *presence detection* is to estimate the probability $p(P^t = 1|I)$, where I is the image. Later we will generalize this slightly by trying to estimate the number of instances of the object class that might be present, $p(N^t|I)$, where $N^t \in \{0, 1, 2, 3–5, 5–10, >10\}$. We call this object *counting*.

The goal of object *localization* is to specify the location and size of each of the object instances. More precisely, let O_i^t be a binary random variable representing whether image patch i contains an object of type t or not, for $i \in \{1, ..., N\}$, where $N \sim 1000$ is the number of image patches. (The size and shape of the image patches varies according to the object type; for side views of cars, we use patches of size 30×80; to handle cars of different sizes, we apply the technique to multiple versions of the image at different scales.) One way to perform localization is to compute the log-likelihood ratio

$$c_i^t = \log p(f_i^t|O_i^t = 1)/p(f_i^t|O_i^t = 0), \quad (2)$$

for each i and t, and then to return all the locations where this log likelihood ratio is above some threshold. Here f_i^t is

a set of local features extracted from image I at patch i for class t. The details of the features and classifier that we used can be found in Torralba et al.[19]

For simplicity, in this paper, we select the D most confident detections (after performing local nonmaximum suppression); let their locations be denoted by ℓ_i^t, for $i \in \{1,...,D\}$. Figure 6a gives an illustration of the output of our system on a typical image. For the results in this paper, we set $D = 10$ so that no correct detections are discarded and still small enough to be efficient. In the figure we show the top $D = 4$ detections to avoid clutter. The locations of each detection ℓ_i^t are indicated by the position and scale of the box, and their confidences c_i^t are indicated by the thickness of the border. In Figure 6b (top), we see that although the system has detected the car, it has also detected three false positives. This is fairly typical of this kind of approach. Below we will see how to eliminate many of these false positives by using global context.

3.2. Object presence detection using global image features

To determine if an object class is present in an image given the gist, we could directly learn a binary classifier of the form $p(P^t = 1|g)$. Similarly, to predict the number of objects, we could learn an ordinal regression function of the form $p(N^t|g)$. Instead, we choose a two-step approach in which we first estimate the category or type of scene, $p(S = s|g)$, and then use this to predict the number of objects present, $p(N^t|S = s)$. This approach has the benefit of having an explicit representation of the scene category (e.g., a street, a highway, a forest) which is also an important desired output of an integrated model.

We can classify the scene using a simple Parzen-window based density estimator

$$p(S = s|g) \propto p(g|S = s) = \frac{1}{J}\sum_{j=1}^{J} \mathcal{N}(g|\mu_j, \sigma_j^2 I),$$

where J is the number of mixture components for each class conditional density. Some examples of scene classification are shown in Figure 4. As shown in Quattoni and Torralba,[12] this technique classifies 75% of the images correctly across 15 different scene categories. Other classifiers give similar performance.

Once we have estimated the scene category, we can predict the number of objects that are present using

$$p(N^t = n|g) = \sum_s p(N^t = n|S = s)\, p(S = s|g) \qquad (3)$$

where $p(N^t = n|S = s)$ is estimated by simple counting.

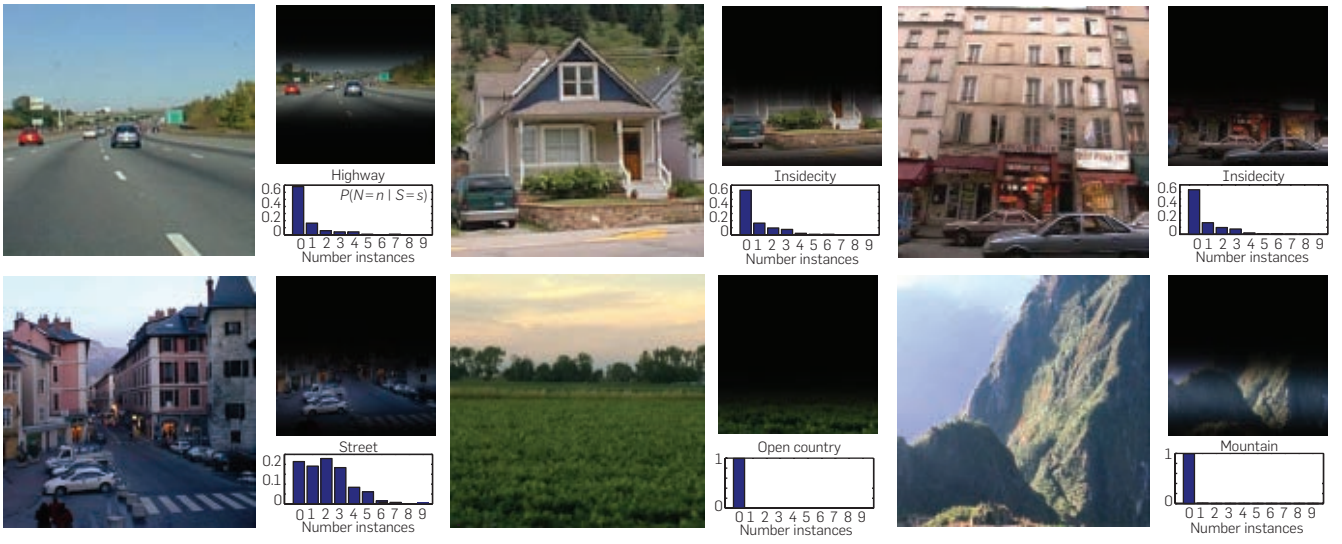
3.3. Object localization using global image features

The gist captures the overall spatial layout of the image, and hence can be used to predict the expected vertical location of each object class before running any detectors; we call this *location priming*. However, the gist is not useful for predicting the *horizontal* locations of objects, which are usually not very constrained by the overall structure of the scene (except possibly by the horizontal location of other objects, a possibility we ignore in this paper).

We can use any nonlinear regression function to learn the mapping from gist to expected vertical location. We used a mixture of experts model,[4] which is a simple weighted average of locally linear regression models. More precisely, we define

$$p(Y^t|g) = \sum_{k=1}^{K} w_k(g)\, \mathcal{N}(Y^t|\beta_k^T g, \sigma_k^2)$$

Figure 4. Predicting the presence/absence of cars in images and their locations using gist. The outputs shown here do not incorporate any information coming from a car detector and are only based on context. Note that in the dataset used to fit the distributions of object counts for each scene category, it is more common to find cars in street scenes (with many cars circulating and parked) than in highway scenes, where there are many shots of empty roads. Hence the histogram for highway shows $p(N^{car} = 0) = 0.6$.



where Y^t is the vertical location of class t, K is the number of experts or mixture components, $\mathcal{N}$ represents a Gaussian or normal distribution, β_k are the regression weights for mixture component k, σ_k^2 is the residual variance, and $w_k(g)$ is the weight or "responsibility" of expert k, given by the softmax or multinomial logistic function:

$$w_k(g) = \frac{\exp(v_k^T g)}{\sum_{k'=1}^{K} \exp(v_{k'}^T g)}$$

We illustrate the predictions made by this model in Figure 6b, where we scale the intensity of each image pixel by the probability density function $p(Y^t|g)$. We see that the effect is to "mask out" regions of the image which are unlikely to contain the object of interest. Some more examples can be seen in Figure 4.

3.4. Integrated model

We now discuss how to combine the various pieces described above. The basic idea is to use the global features to make "top-down" predictions about how many object instances should be present, and where, and then to use the local patch classifiers to provide "bottom-up" signals.

The key issue is how to combine these two information sources. The approach we take is as follows (this differs slightly from the method originally described in Murphy et al.[9]). Let us initially ignore location information. We treat the confidence score of the detector (c_i^t, defined in Equation 2) as a local likelihood term, and fit a model of the form $p(c_i^t|O_i^t = o) = \mathcal{N}(c_i^t|\mu_o^t, \sigma_o^t)$ for $o \in \{0, 1\}$. We can learn the parameters of this Gaussian by computing the empirical mean and variance of the scores when the detector is applied to a set of patches which do contain the object (so $o = 1$) and which do not contain the object (so $o = 0$). If we have a uniform prior over whether each detection is a true or false positive, $p(O_i^t = 1) = 0.5$, we can compute the posterior using Bayes rule as follows:

$$p(O_i^t = 1|c_i^t) = \frac{p(c_i^t|O_i^t = 1)}{p(c_i^t|O_i^t = 1) + p(c_i^t|O_i^t = 0)}$$

However, the detections are not all independent, since we have the constraint that $N^t = \sum_{i=1}^{D} I(O_i^t = 1)$, where N^t is the number of objects of type t. If we have top-down information about N^t from the gist, based on Equation 3, then we can compute the posterior distribution over detections in $O(2^D)$ time, given the gist, as follows:

$$p(O_{1:D}^t|g) \propto \sum_{n=0}^{D} p(O_{1:D}^t|n)p(N^t = n|g)$$

Here the term $p(O_{1:D}^t|n)$ is 1 only if the bit vector $O_{1:D}^t$ of length D has precisely n elements turned on. For compactness, we use the notation 1: D to denote the indices 1, ..., D. We can combine this with the local detectors as follows:

$$p(O_{1:D}^t|c_{1:D}^t, g) \propto p(O_{1:D}^t|g) \prod_{i=1}^{D} p(c_i^t|O_i^t)$$

If the gist strongly suggests that the object class is absent, then $p(N^t = 0|g) \approx 1$, so we turn all the object bits off in the posterior regardless of the detector scores, $p(O_{1:D}^t = \mathbf{0}|c_{1:D}^t, g) \approx 1$. If the gist strongly indicates that one object is present, then $p(N^t = 1|g) \approx 1$, and only one O_i^t bit will be turned on in the posterior; this will be the one with the highest detector score. And so on.

Now we discuss how to integrate location information. Let ℓ_i^t be the location of the i'th detection for class t. Since Y^t represents the expected location of an object of class t, we define another local likelihood term $p(\ell_i^t|O_i^t = 1, Y^t) = \mathcal{N}(\ell_i^t|Y^t, \tau^t)$, where τ^t is the variance around the predicted location. If the object is absent, we use a uniform distribution $p(\ell_i^t|O_i^t = 0, Y^t) \propto 1$. Of course, Y^t is not observed directly, but we can predict it based on the gist; this yields

$$p(\ell_i^t|O_i^t, g) = \int p(\ell_i^t|O_i^t, Y_t)p(Y_t|g)dY_t$$

which can be solved in closed form, since it is the convolution of two Gaussians. We can now combine expected location and detections as follows:

$$p(O_{1:D}^t|c_{1:D}^t, \ell_{1:D}^t, g) \propto p(O_{1:D}^t|g) \prod_{i=1}^{D} p(c_i^t|O_i^t)p(\ell_i^t|O_i^t, g)$$

To see the effect of this, suppose that the gist strongly suggests that only one object of type t is present, $p(N^t = 1|g) \approx 1$; in this case, the object bit which is turned on will be the one that has the highest score and which is in the most likely location. Thus confident detections in improbable locations are suppressed; similarly, unconfident detections in likely locations are boosted.

Finally, we discuss how to combine multiple types of objects. Intuitively, the presence of a car makes the presence of a pedestrian more likely, but the presence of a computer monitor less likely. However, it is impractical to encode a joint distribution of the form $p(P^1, ..., P^T)$ directly, since this would require $O(2^T)$ parameters. (Encoding $p(N^1, ..., N^T)$ directly would be even worse.) Instead, we introduce the scene category latent variable S, and assume that the presence (and number) of object types is conditionally independent given the scene category:

$$p(N^1, \ldots, N^T) = \sum_s p(S = s) \prod_{t=1}^{T} p(N^t|S = s)$$

Given this assumption, we can perform inference for multiple object types in parallel as follows: for each possible scene category, compute the posterior $p(O_{1:D}^t|c_{1:D}^t, \ell_{1:D}^t, g, S = s)$ as described above, and then combine them using a weighted average with $p(S = s|g)$ as the weights.

In summary, our whole model is the following joint probability distribution:

$$p(O_{1:D}^{1:T}, N^{1:T}, Y^{1:T}, S|c_{1:D}^{1:T}, \ell_{1:D}^{1:T}, g) \propto p(S|g) \times \prod_{t=1}^{T} p(Y^t|g)p(N^t|S)p(O_{1:D}^t|N_t) \prod_{i=1}^{D} p(\ell_i^t|O_i, Y^t)p(c_i^t|O_i)$$

This is illustrated as a probabilistic graphical model (see e.g., Koller and Friedman[5]) in Figure 5. There is one node for each random variable: the shaded nodes are observed (these are deterministic functions of the image), and the unshaded nodes are hidden or unknown, and need to be inferred. There is a directed edge into each node from all the variables it directly depends on. For example, the $g \rightarrow S$ are reflects the scene classifier; the $g \rightarrow Y^t$ arc reflects the location priming based on the gist; the $S \rightarrow N^t$ arc reflects the object counts given the scene category; the $O_i^t \rightarrow c_i^t$ arc reflects the fact that the presence or absence of an object of type t in patch i affects the detector score or confidence c_i^t; the $O_i^t \rightarrow \ell_i^t$ arc is a deterministic link encoding of the location of patch i; the $Y^t \rightarrow \ell_i^t$ arc reflects the $p(\ell_i^t | Y^t, O_i^t)$ term; finally, there are the $O_i^t \rightarrow \Sigma^t$ and $N^t \rightarrow \Sigma^t$ arcs, which is simply a trick for enforcing the $N^t = \Sigma_{i=1}^D I(O_i^t = 1)$ constraint. The Σ^t node is a dummy node used to enforce the constraint between the N^t nodes and the O_i^t nodes. Specifically, it is "clamped" to a fixed state, and we then define $p(\Sigma^t | O_{1:D}^t, N^t = n) = \mathcal{I}(\Sigma_i O_i^t = n)$ (conditional on the observed child Σ^t, all the parent nodes, N^t and O_i^t, become correlated due to the "explaining away" phenomenon[5]).

From Figure 5, it is clear that by conditioning on S, we can perform inference on each type of object independently in parallel. The time complexity for exact inference in this model is $O(ST2^D)$, ignoring the cost of running the detectors. (Techniques for quickly evaluating detectors on large images, using cascades of features, are discussed in Viola and Jones[20].) We can speed up inference in several ways. For example, we can prune out improbable object categories (and not run their detectors) if $p(N^t > 0|g)$ is too low, which is very effective since g is fast to compute. Of the categories that survive, we can just run their detectors in the primed region, near $E(Y^t|g)$. This will reduce the number of detections D per category. Finally, if necessary, we can use Monte Carlo inference (such as Gibbs sampling) in the resulting pruned graphical model to reduce time complexity.

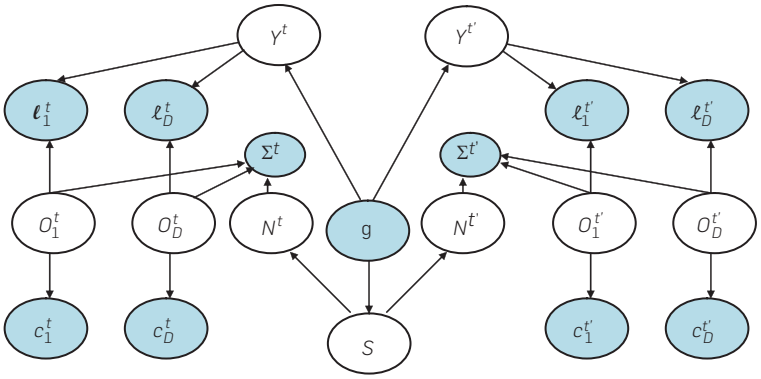
4. RESULTS

Examples of the integrated system in action are shown in Figure 6c: We see that location priming, based on the gist, has down-weighted the scores of the detections in improbable locations, thus eliminating false positives. In the second row, the local detector is able to produce a confident detection, but the second car produces a low confidence detection. As the low confident detection falls inside the predicted region, the confidence of the detection increases. Note that in this example there are two false alarms that happen to also fall within the prediction region. In this case, the overall system will increase the magnitude of the error. If the detector produces errors that are contextually correct, the integrated model will not be able to discard those. The third row shows a different example of failure of the integrated model. In this case, the structure of the scene makes the system think that this is a street scene, and then mixes the boats with cars. Despite these sources of errors, the performances of the integrated system are substantially better than the performances of the car detectors in isolation.

For a more quantitative study of the performance of our method, we used the scenes dataset from Oliva and Torralba[11] consisting of 2688 images covering 8 scene categories (streets, building facades, skyscrapers, highways, mountainous landscapes, coast, beach, and fields). We use half of the dataset to train the models and the other half for testing.

Figure 7 shows performances at two tasks: object localization and object presence detection. The plots correspond to precision–recall plots: the horizontal axis denotes the percentage of cars in the database that have been detected for a particular detection threshold and the vertical axis is the percentage of correct detections for the same threshold. Different points in the graph are achieved by varying the decision threshold. For both tasks, the plot shows the performances using an object detector alone, the performances of the integrated model, and the performance of an integrated model with an oracle that tells for each image the true context.

Figure 5. Integrated system represented as a directed graphical model. We show two object types, *t* and *t'*, for simplicity. The observed variables are shaded circles, the unknown variables are clear circles. Variables are defined in the text. The Σ^t node is a dummy node used to enforce the constraint between the N^t nodes and the O_i^t nodes. O_i^t = indicator of presence of object class t in box i; Y^t = vertical location of object class *t*; N^t = number of instances of object class *t*; l_i^t = location of box i for object class *t*; c_i^t = score of box i for object class i; D = number of high-confidence detections; g = gist descriptor; S = scene category.



O_i^t—Indicator of presence of object class t in box i

Y^t—Vertical location of object class t

N^t—Number of instances of object class t

ℓ_i^t—Location of box i for object class t

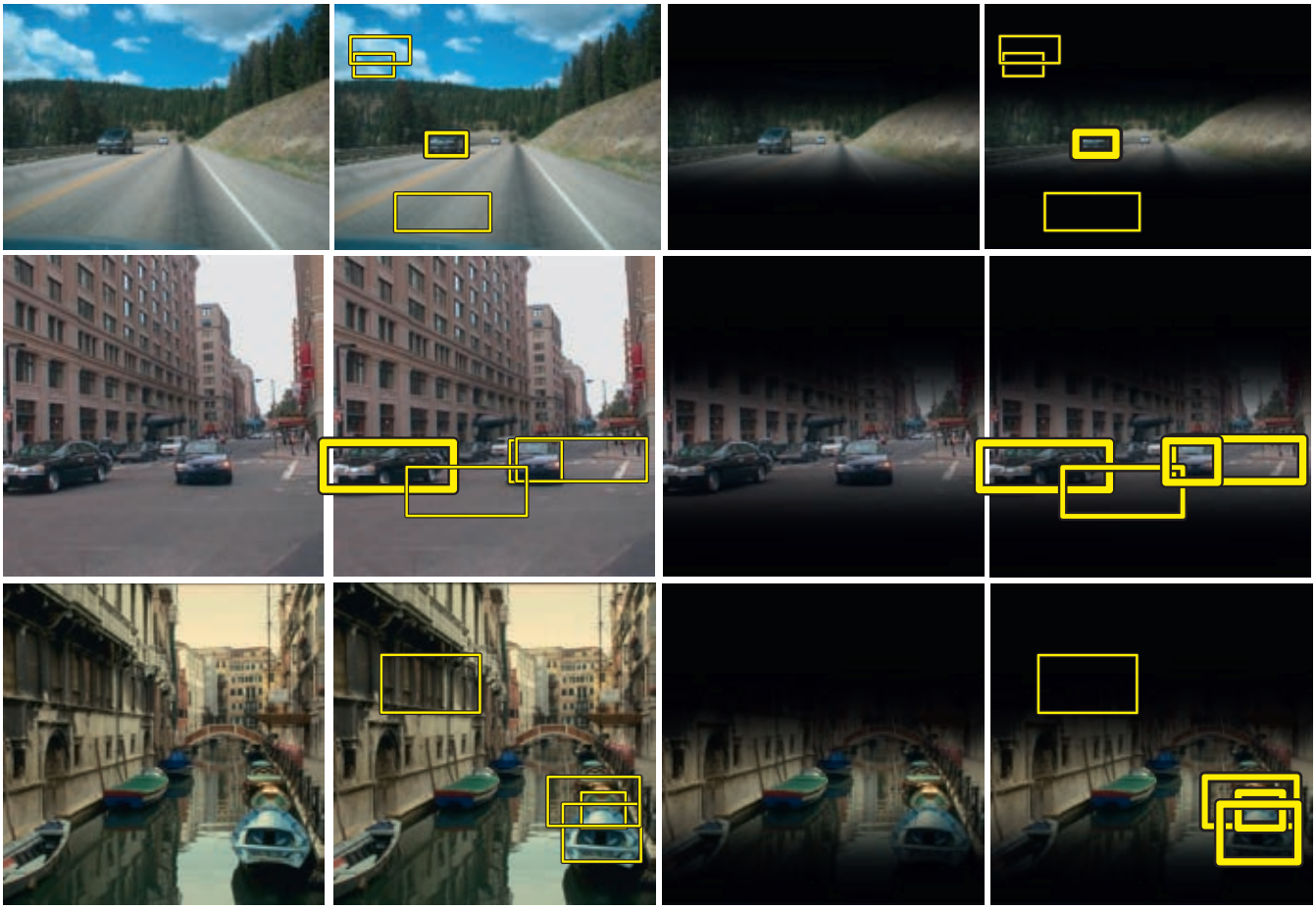
c_i^t—Score of box i for object class t

D—Number of high-confidence detections

g—Gist descriptor

S—Scene category

Figure 6. (a) Three input images. (b) Top four detections from an object detector based on local features. The thickness of the boxes is related to the confidence of the detection. (c) Predicted location of the car based on global features. (d) Combining local and global features.



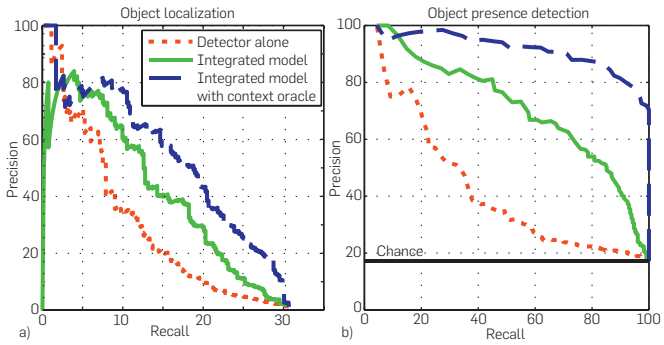
(a) Input image (b) Car detector output (c) Location priming (d) Integrated model output

The performance of the integrated model has to be within the performance of the detector alone and the context oracle.

Figure 7 (right) shows a precision–recall curve which quantifies the performance of three different systems for detecting object presence. The worst one is based on an object detector using local features alone; the middle one is our integrated system which uses local and global features; and the best one is an oracle system based on using the true scene category label. We see that our integrated model does much better than just using a detector, but it is clear that better scene classification would improve the results further. It is important to note that detecting if an object is present in an image can be done with good accuracy even without object localization. The knowledge of the scene depicted by the image can be enough. For instance, in a picture of a street it is quite certain that a car will appear in the picture, while it is unlikely that a car will appear on a beach scene. Therefore, the relation between the scene category and the object can provide a lot of information even when the detector fails to locate the object in the image.

Figure 7 (left) shows a precision–recall curve which quantifies the performance of three different systems for localizing objects. Again the worst one is based on an object detector using local features alone; the middle one is our integrated system which uses local and global features; and the best one is a oracle system based on using the true scene category label. In this case, knowing the true scene category does not help as much: it can eliminate false positives such as cars in indoor scenes, but it cannot eliminate false

Figure 7. Performance on car localization (left) and car presence detection (right).



positives such as cars detected in a street scene but up in the sky. (Of course, the gist-based location priming system tries to eliminate such spatial outliers, but knowing the scene category label does not help with localization.)

Object localization is a much harder task than merely detecting the presence of an object. This is evident from the horizontal scale in Figure 7 (left): the recall never goes beyond about 30%, meaning that about 70% of cars are missed by the detector, mostly due to occlusion. Even if context can be used to narrow down the search space and to remove false alarms that occur outside the relevant image region, still, if the detector is not able to localize the object, context information will not be able to precisely localize the object. The use of global context (even with the oracle) does not increase the recall (as this requires the detector to work), however context is able to increase the precision as it is able to remove false alarms in scenes in which cars are not expected. It is possible that a finer grained notion of context, perhaps based on other objects, could help in such cases. Note, however, that for image retrieval applications (e.g., on the web), object presence detection is sufficient. For speed reasons, we could adopt the following two stage approach: first select images that are predicted to contain the object based on the gist alone, since this is much faster than applying a sliding window classifier; then apply the integrated model to further reduce false positives.

5. CONCLUSION

We have discussed one approach for combining local and global features in visual object detection and localization. Of course, the system is not perfect. For example, sometimes objects appear out of context and may be accidently eliminated if the local evidence is ambiguous (see Figure 8). The only way to prevent this is if the local detector gives a sufficiently strong bottom-up signal. Conversely, if the detector makes a false-positive error in a contextually plausible location, it will not be ruled out by our system. But even people can also suffer from such "hallucinations."

In more general terms, we see our system as a good example of probabilistic information fusion, an approach which is widely used in other areas such as speech recognition, which combines local acoustic models which longer-range language models. Since computer vision is inherently a difficult inverse problem, we believe it will be necessary to combine as many sources of evidence as possible when trying to infer the true underlying scene structure.

Figure 8. An object which is out of context may be falsely eliminated by our system.

Acknowledgments

Funding for this work was provided by NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, NSF Career award IIS 0747120, NSF contract IIS-0413232, a National Defense Science and Engineering Graduate Fellowship, and gifts from Microsoft and Google. KPM would like to thank NSERC and CIFAR for support.

References

1. Fei-Fei, L., Perona, P. A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2005), 524–531.
2. Heeger, D., Bergen, J.R. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, USA, 1995). ACM, NY, 229–238.
3. Hoiem, D., Efros, A., Hebert, M. Geometric context from a single image. In *IEEE International Conference on Computer Vision* (2005).
4. Jordan, M.I., Jacobs, R.A. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6* (1994), 181–214.
5. Koller, D., Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
6. Kumar, S., Hebert, M. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *IEEE International Conference on Computer Vision* (2003).
7. Lazebnik, S., Schmid, C., Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2006), 2169–2178.
8. Murphy, K., Torralba, A., Eaton, D., Freeman, W.T. Object detection and localization using local and global features. *Toward Category-Level Object Recognition*. J. Ponce, M. Hebert, C. Schmidt, and A. Zisserman, eds. 2006.
9. Murphy, K., Torralba, A., Freeman, W. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Advances in Neural Information Proceedings Systems* (2003).
10. Oliva, A., Schyns, P.G. Diagnostic color blobs mediate scene recognition. *Cogn. Psychol. 41* (2000), 176–210.
11. Oliva, A., Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comp. Vision 42* (2001), 145–175.
12. Quattoni, A., Torralba, A. Recognizing indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009), 413–420.
13. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S. Objects in context. In *IEEE International Conference on Computer Vision* (Rio de Janeiro, 2007).
14. Richard, X.H., Zemel, R.S., Carreira-perpinan, M.A. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2004), 695–702.
15. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T. LabelMe: a database and web-based tool for image annotation. *Int. J. Comp. Vision 77*, 1–3 (2008), 157–173.
16. Strat, T.M., Fischler, M.A. Context-based vision: recognizing objects using information from both 2-D and 3-D imagery. *IEEE Transaction on Pattern Analysis and Machine Intelligence 13*, 10 (1991) 1050–1065.
17. Torralba, A. Contextual priming for object detection. *Int. J. Comp. Vision 53*, 2 (2003), 153–167.
18. Torralba, A., Murphy, K., Freeman, W. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Proceedings Systems* (2004).
19. Torralba, A., Murphy, K.P., Freeman, W.T. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 5 (2007), 854–869.
20. Viola, P., Jones, M. Robust real-time object detection. *Int. J. Comp. Vision 57*, 2 (2004), 137–154.

A. Torralba (torralba@csail.mit.edu), Computer Science and Artificial Intelligence Laboratory, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

K. P. Murphy (murphyk@cs.ubc.ca), Department of Computer Science, University of British Columbia, Vancouver, Canada.

W. T. Freeman (billf@csail.mit.edu), Computer Science and Artificial Intelligence Laboratory, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

CAREERS

Kuwait University
College of Engineering and Petroleum
Kuwait

The Department of **Computer Engineering** at Kuwait University is seeking qualified applicants for a faculty position in the Networks field starting from September 2010.

Required Qualifications:
Ph.D. degree in Computer Engineering with specialization in computer networks from a reputable university. Applicants should have a minimum GPA of 3.0/4.0 or equivalent at the undergraduate level. Applicants should have well-established research experience and publications in refereed international journals. Applicants should have demonstrated outstanding teaching experience in the specified field as well. The successful candidate is expected to teach at both the undergraduate and graduate level and to establish an active collaborative research program. For further information refer to the website: http://www.eng.kuniv.edu/

To apply send by express mail/courier service or e-mail, within **six weeks** of the date of announcement, a copy of this ad, a completed application form, with required documents as stated in the application form as well as a detailed CV, a copy of the passport and three recommendation letters, to the following address:

Administration for Academic Staff Affairs
Academic Staff Appointment Department
Kuwait University, Khaldiya Campus
Block 3, Al Firdous Street Building No: 3
Khaldiya, State of Kuwait
Tel: 00965-24844189 Fax: 00965-24849562
Email: Vpaa.faculty@ku.edu.kw

For application forms refer to website http://www.kuniv.edu/ku/Downloads/index.htm

Kuwait University
Faculty of Science
Kuwait

The Department of Mathematics and Computer Science at the Faculty of Science, Kuwait University invites applications for the academic year 2010/2011, in the following specialized areas: **Algorithms; Computer Architecture, Database, Graphics, Operating Systems, Software Engineering; and Theory of Computation**.

Required Qualifications:
Applicants should have bachelors, masters, and doctorate degrees in **Computer Science** from a reputable university. The applicant's GPA in the first university degree should be 3 out of 4 or its equivalent. Research experience and publication in refereed international journals are also required. A full command of English and a minimum of 2 years experience in university teaching in the specified field are expected. The successful candidates are expected to have a strong commitment and dedication to quality teaching and research. For further information refer to the website: http://www.science.kuniv.edu.kw/

To apply send by express mail/courier service or e-mail, within six weeks of the date of announcement, a copy of this ad, a completed application form, with required documents as stated in the application form as well as a detailed CV, a copy of the passport and three recommendation letters, to the following address:

Administration for Academic Staff Affairs
Academic Staff Appointment Department
Kuwait University, Khaldiya Campus
Block 3, Al Firdous Street, Building No: 3
Khaldiya, State of Kuwait
Tel: 00965-24844189 Fax: 00965-24849562
Email: Vpaa.faculty@ku.edu.kw

For application forms refer to website http://www.kuniv.edu/ku/Downloads/index.htm

NEC Laboratories America, Inc.
Research Staff Member - Grid Storage

NEC Laboratories America, Inc. is seeking researchers who are passionate about solving real world problems to join our Grid Storage Department in Princeton, NJ. The department engages in storage research with a focus on networked storage. To qualify for the position, candidates must have:

- PhD in Computer Science (or equivalent), with a strong publication record
- Experience with storage systems (file systems, object or content based storage systems, databases)
- Experience in designing, building, and evaluating distributed systems and protocols
- Knowledge of fault tolerance and availability techniques for local and wide area networked systems
- Excellent verbal and written communication skills

Candidates must be proactive and assume leadership in proposing and executing innovative research projects, as well as in developing advanced prototypes leading to demonstration in an industry environment.

Experience in Cloud Computing or SaaS is a plus.

For more information, visit http://www.nec-labs.com/careers/. For consideration, please forward résumé to recruit@nec-labs.com and reference "Grid Storage" in the subject line.

EOE/AA/MFDV

North Carolina Central University
Computer Science Faculty Position

The Department of Mathematics and Computer Science invites applications for tenure-track faculty positions in all areas beginning Fall 2010. A Ph.D. in computer science or related area is required. The successful candidate must have a commitment to the academic process, excellence in research, education, and service, and to diversity in the community. The candidate must have a desire to participate in student academic and thesis advising, and curriculum development. We are particularly interested in candidates with research interests in artificial intelligence, computer vision, computer graphics, grid computing, robotics, computational biology, software engineering, multimedia applications, networks, mobile computing, wireless sensor networks and security/cryptography.

The campus is located in the Research Triangle Area, an ideal location in NC with several universities and high-tech companies. Applications and inquiries should be sent to ruma@nccu.edu. Further information can be found at: http://boole.cs.nccu.edu/emp2010/employment.html

Departmental resources include extensive computing facilities of workstations, servers and personal computers with multimedia capabilities and specialized networks and devices. Faculty members have access to high performance computing platforms provided by the university and its partners.

Sandia National Laboratories
Math & CS Research
Discrete Math and Computer Science R&D for Social and Computer Network Analysis

Sandia National Laboratories seeks new PhD researchers for long-term positions in mathematics and computer science for understanding large-scale, complex, social and engineered networks. Of particular interest are experts in computational topology, graph-feature identification, community detection, statistics, machine learning, computational linguistics, and uncertainty.

PhD in CS, math, statistics, or equivalent is required. Publications, software, and application experience is desired. The position involves national security applications; the ability to obtain and maintain a U.S. security clearance is required.

Apply at http://www.sandia.gov/careers to Job ID 64380 before 3 April 2010.

See http://www.cs.sandia.gov/hpc-informatics/careers/index.htm or contact Brett Bader (bwbader@sandia.gov) for details. Sandia National Laboratories is an Equal Opportunity Employer M/F/D/V.

Strategic Analysis Enterprises
Software Engineer

Seek full-time software engineer to support and extend a text information extraction system and perform other programming tasks. Salary is commensurate with qualifications. SAE provides health & dental insurance, 401k plan, profit sharing, and bonus earning possibilities to its employees.

Master's or doctorate in computer science or computational linguistics, ability to program in C# and build a good Windows user interface (Java programmers who can switch to C# will be considered). Desirable: knowledge of pragmatics and English lexical semantics; Python and/or Perl; statistical research methods. US citizenship is mandatory.

Apply for this job: steve@strategicanalysisenterprises.com

The University of Tennessee at Chattanooga
Assistant/Associate Professor

UTC invites applications for a full-time, tenure track appointment in Computer Science and Engineering, beginning July 1, 2010. The department seeks applicants with a Ph.D. in Computer Science or Computer Engineering, and a commitment to excellence in teaching and research. The successful candidate will have experience in teaching a broad spectrum of computer related courses, and will have experience/interest in interdisciplinary teaching and research, with a particular emphasis on theoretical aspects of Computer Science. The CSE department (www.cs.utc.edu), part of the College of Engineering and Computer Science, offers an ABET accredited B.S. degree, a M.S. degree, and has received certification by the CNSS, NSA, and DHA as a National Center of Academic Excellence in Information Assurance Education. The College is also home to the SimCenter and its graduate programs (MS/Ph.D.) in Computational Engineering.

To apply, please e-mail in Word or pdf format an application letter, resume and descriptions of teaching and research philosophies to Dr. Claire McCullough, Claire-McCullough@utc.edu. Also, please arrange for 3 letters of recommendation and a copy of your transcript listing the completion of your doctoral degree to:

Faculty Search Committee
Computer Science, Dept. 2302
The University of Tennessee at Chattanooga
735 Vine Street
Chattanooga, TN 37403-2598

Screening of applicants who have provided complete information will begin immediately and continue until the position is filled. The University of Tennessee at Chattanooga is an equal employment opportunity/affirmative action/Title VI & IX/Section 504 ADA/ADEA institution, and, as such, encourages the application of qualified women and minorities.

DOI:10.1145/1666420.1666447

Peter Winkler

Puzzled
Solutions and Sources

Last month (February 2010, p. 120) we posted a trio of brainteasers, including one as yet unsolved, concerning the breaking of a bar of chocolate.

1. Making S'Mores. *Solution.* Charlie was able to break a five-by-nine-square-segment rectangular bar into its constituent squares using 44 breaks along its seams. But could he have done better? If you tried it yourself, you might conclude that he could not have done better, but also that he couldn't have done worse. Indeed, breaking only one piece at a time, Charlie is foreordained to make exactly 44 breaks.

Very smart people have been stumped by this puzzle, only to slap themselves on the forehead when they realized that every break increases the number of pieces of chocolate by one. Since there are 45 squares, there must be 44 breaks, no matter how they did it. In general, of course, given an m by n chocolate bar, the conclusion is that the required number of breaks is always $mn - 1$.

2. Playing Chomp. *Solution.* Charlie's children, Alice and Bobby, play a game called Chomp in which they alternate eating a square together with every square northeast of the first square, trying to avoid eating the last square.

The game was invented (in a different form) in 1952 by Dutch mathematician Frederik "Fred" Schuh and independently in 1974 by the late mathematician and economist David Gale. The name "Chomp" was coined by an amateur mathematician, the great puzzle maven Martin Gardner. The proof that Alice can force a win is a classic strategy-stealing argument that goes like this: First, since the game is deterministic, full-information, and bounded in length, *someone* must have a winning strategy. Assume it's Bobby, and let square X be his winning reply to Alice's first move of biting off only the northeast corner square. But Alice could instead have begun by taking X (and everything northeast of X) on her first move, later adopting Bobby's winning strategy.

This contradiction shows it must have been Alice, not Bobby, who had the winning strategy.

3. Alice's Winning Strategy. This proof works for any m by n chocolate bar (as long as it has more than one square) but fails to reveal what Alice's winning strategy actually is. Subsequent work (such as at http://www.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/chomp.html) has solved the three-row game, but still no one knows how Alice is able to win the game in general. Indeed, there may not be a general strategy that can be described in a simple way.

But, hey, you never know. The game Bridg-It, also invented by Gale and publicized by Gardner, had the same curious property: It could be proved that the first player had a winning strategy, though no such strategy is known. It was later produced and sold commercially as a board game by game publisher Hasbro. Mathematician Oliver Gross of the Rand Corporation then came up with an elegant winning strategy. Explore the game and that remarkable strategy at http://home.flash.net/~markthom/html/bridg-it.html.

So maybe Chomp has an elegant winning strategy after all. Meanwhile, if you find one, please tell the rest of us.

All readers are encouraged to submit prospective puzzles for future columns to puzzled@cacm.acm.org.

Peter Winkler (puzzled@cacm.acm.org) is Professor of Mathematics and of Computer Science and Albert Bradley Third Century Professor in the Sciences at Dartmouth College, Hanover, NH.

[CONTINUED FROM P. 120] be grafted onto our instincts and drilled into our minds. It's just a set of guerilla tactics for the lawless byways and ramshackle security of the Internet. Consider the warning about giving away our passwords. Are you "giving it away" if the site that requests it promises not to store it—as social networking sites often do? Are we even aware that we're giving it away if a Trojan (infected software) on our computer pops up an apparently perfect but fake Web page for our online banks? Other planks of cybersecurity education are equally flimsy.

Online privacy is another arena in which human instinct is foundering. Drawing a curtain over a window at night offers a concrete, intuitive form of privacy (and doesn't require agreement to a thousand-word privacy policy). Online privacy is a different matter. Suppose the average user—or savvy one, for that matter—could digest online privacy policies. Suppose the policy was simply "you own your data," a widely favored nostrum. It is still well beyond any person's mental capacity today to understand what data this person owns and how to go about controlling it. When, for instance, photos of our face seep into search engines, friends' online content, archived Webcam images, and digital photo albums of sightseers in cities we've visited, what does ownership or control mean?

Cybersecurity education often fails because it doesn't teach fundamental principles that can be grafted onto our instincts.

The poster children for the future of computer security are often intellectually flashy inventions, such as, say, quantum cryptography. These technological showpieces create trustworthy connections between machines (sometimes) but not trustworthy connections between people—the source of the real challenge.

The Romans adjusted to a new material world. Today, we're mentally capable of translating numbers on computer screens into a measure of wealth, then into bread and circuses, houses, clothes, and cars. Human instinct lags in most of the places where cyberspace is swelling and ramifying. A future of informed and secure choice demands tools—technological, educational, policy-oriented—that project cyberspace down to the scale of human instinct and intelligence. If not, we might wind up as stupefied as an early Roman staring at a chunk of bronze. C

Ari Juels (ari.juels@rsa.com) is chief scientist and director of RSA Laboratories, Cambridge, MA, and author of the novel *Tetraktys*, Emerald Bay Books, Newport Coast, CA, 2009.

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 will and could be.

DOI:10.1145/1666420.1666448

Ari Juels

Future Tense
The Primal Cue

Cybersecurity depends on the human dimension.

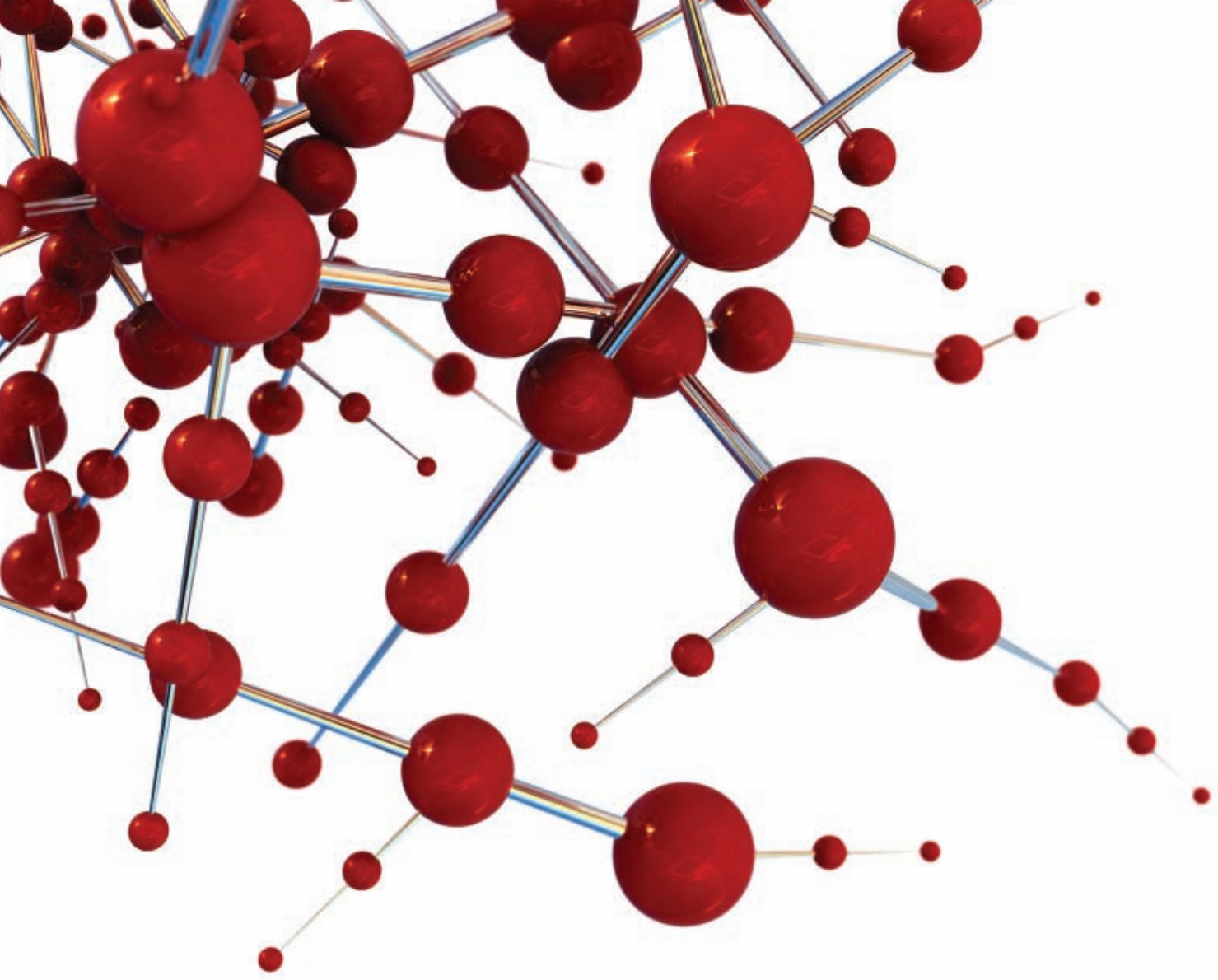
Many centuries ago, a mystified Roman farmer held a bronze ingot crudely imprinted with a cow. He was handling an early form of currency that supplanted a true cow—a life-sustaining, milk-and-flesh-producing piece of wealth—with a chunk of metal that was strangely, with its embossed animal figure, supposedly of equivalent value. (Roman cattle spawned our English word "pecuniary"; the Latin for cattle is "pecus.")

The early Romans faced an abstraction that often distorted the material world beyond their intuition. Their befuddlement gives an historical glimpse of the vast mental challenges that people of all stripes face today as cyberspace undercuts our own deeply embedded intuition and instincts—with ripple effects throughout security and privacy.

For pecuniary surrealism today, look no farther than virtual worlds like *World of Warcraft* and *Second Life*. In them, developers of virtual "real estate" earn real-world money for their oxymoronic efforts. Laborers in third-world sweatshops work in gold mines represented only in cyberspace. There have been real-world prosecutions for larceny of virtual-world goods and at least one real-world murder over the theft of a virtual sword. Virtual-world currency is spilling over into the real world in the billions of dollars, adding a new dimension to security concerns like money laundering. The law can't keep pace with these phenomena; the Internal Revenue Service doesn't yet know whether or how to tax them. The interpenetration of the real and virtual worlds is happening in other ways, too. It's possible to order a pizza in a virtual world and have it delivered to our real doorsteps. It's just a matter of time before other real-virtual linkages become routine, say, surgery conducted in a virtual world operating on real patients and electric grids mapped into virtual space. Security failures will inevitably propagate from virtual worlds into the real one.

It's difficult to wrap our minds around these virtual/real entanglements. But the online world also thwarts our security instincts in much simpler ways. Humans are biologically wired to make trust judgments through attunement to faces, gestures, and verbal intonations. Social networking sites strip away these primal cues. For instance, when a social networking site used by friends asks us to log into an external email account, the request seems instinctively safe thanks to the friends' implicit endorsement. Some social networking sites have exploited this herd instinct toward safety to entrap subscribers through viral attacks. They invite new users to "Log into your email account so we can see if you have other friends on this network." They then hijack our address books and send email to our contacts in their name—inviting new victims in turn to join the social network and render themselves vulnerable to the same trick.

Consumer education about online security is often trumpeted as a countermeasure to such blunders. "Never give away your email password to another site" is a ubiquitous warning. But cybersecurity education often fails because it's not true education. It doesn't teach fundamental principles that can [CONTINUED ON P. 119]



Think Parallel.....

It's not just what we make.
It's what we make possible.

Advancing Technology Curriculum
Driving Software Evolution
Fostering Tomorrow's Innovators

Learn more at: www.intel.com/thinkparallel